Information-consistent systems modeling and analysis

With applications in offshore engineering

Glavind, Sebastian Tølbøll

DOI (link to publication from Publisher):
10.54337/aau443238866

Publication date:
2021

Document Version
Publisher's PDF, also known as Version of record

INFORMATION-CONSISTENT SYSTEMS MODELING AND ANALYSIS

WITH APPLICATIONS IN OFFSHORE ENGINEERING

BY
SEBASTIAN TØLBØLL GLAVIND

DISSERTATION SUBMITTED 2021

AALBORG UNIVERSITY
DENMARK

INFORMATION-CONSISTENT SYSTEMS MODELING AND ANALYSIS

WITH APPLICATIONS IN OFFSHORE ENGINEERING

Ph.D. Dissertation

Sebastian Tølbøll Glavind

Aalborg University
Department of the Built Environment
Thomas Manns Vej 23
DK-9220 Aalborg Ø
Denmark

Dissertation submitted: April 5, 2021

PhD supervisor: Prof. Michael Havbro Faber,
Aalborg University, Denmark

Assistant PhD supervisors: Prof. John Dalsgaard Sørensen,
Aalborg University, Denmark

Prof. Bo Friis Nielsen,
Technical University of Denmark

PhD committee: Professor Lars Bo Ibsen (chairman)
Aalborg University

Prof. Engineer, Prof emeritus, Marc Maes
University of Calgary

Professor Eleni Chatzi
ETH Zürich

PhD Series: Faculty of Engineering and Science, Aalborg University

Department: Department of the Build Environment

ISSN (online): 2446-1636
ISBN (online): 978-87-7210-928-2

Published by:
Aalborg University Press
Kroghstræde 3
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Rosendahls, 2021

ABSTRACT

In the engineering community, it is increasingly appreciated that traditional systems modeling has a significant potential for improvement. The main purpose of the present thesis is thus to introduce and further develop recent research on probabilistic modeling and analysis of complex systems, which has relevance within offshore engineering. This involves showcasing how current state-of-the-art machine learning frameworks, such as Bayesian networks, Gaussian processes, and neural networks, can be applied to formulate knowledge- and information-consistent models within the domain. Such models need not only to reflect average tendencies within the modeling domain but should also be able to consistently utilize the available data sources, e.g., experiments, measurements, hindcasts and forecasts, and represent as well as propagate the associated uncertainties. It is emphasized that although this research is demonstrated in the context of offshore engineering, it is fully generic and applies in principle to any field of application where complex systems occur.

Appreciating that model building is inherently a subjective task that relies on the modeler's bias towards and knowledge of existing modeling frameworks, as well as the available or chosen data for modeling, we often end up with a set of relevant model hypotheses that explain the data almost equally well. At the same time, our modeling efforts should always be seen in relation to the decisions they serve to support, i.e., it is imperative that system representations support the decision context at hand in the best way possible when performing decision optimization. This research accommodates these considerations by introducing practical statistical and decision analytical frameworks for dealing with competing system representations in inferential modeling and decision-making.

To show how the modeling frameworks covered in the thesis can be applied in practice, a GitHub repository containing toolboxes, code examples, and tutorials developed during the PhD project is provided. This includes toolboxes developed for Bayesian network learning, which are used in several research papers, and tutorials applying the toolboxes. For systems representation, additional tutorials and code examples are available on the application of among others Gaussian process, neural network, and gradient boosting machines. For systems analysis, several examples are available on cluster and sensitivity analysis, and tutorials on concepts such as hyper-parameter tuning, and model selection and averaging are available as well.

ABSTRACT

The research papers introduce a set of principle examples on the application of the systems modeling approaches. These examples consider different aspects relevant when building probabilistic models for the offshore engineering community. First, probabilistic models are formulated for offshore, environmental storm events, accounting for domain knowledge, as well as measurement and hindcast data. In this regard, the framework of Bayesian networks is used to formulate computational efficient, categorical models for storm events, and Gaussian processes are used to represent the discrepancies between measurements and hindcasts. This research also shows how to consistently account for competing system representations when considering a specific decision context. Second, probabilistic modeling of fatigue crack growth in welded, steel joints is considered. This work employs Bayesian hierarchical modeling to jointly model data collected from different laboratory experiments, thus representing different measurement uncertainties, and Bayesian model averaging for statistical inference in case of new details. This approach provides a means for systematically accounting for the uncertainties, which naturally scales to larger data sets.

The research papers further introduce a set of principle examples on the analysis of complex system performances. First, the potential of cluster and sensitivity analysis are explored for a simple moment resisting portal frame structure. These assessments show how model-based clustering may be used to establish a probabilistic representation of realizations exhibiting a certain target behavior, which provides significant insight on the characteristics of the targeted system behavior in terms of e.g., latent groupings in the realizations and driving variables. Moreover, it is shown how variance-based sensitivity analysis can be used to decompose the variability in system responses into contributions from the individual, random system inputs, which thus provides a means for assessing e.g., the importance of the inputs in driving the random responses. Second, a framework for systems identification is presented in the context of damage detection for the simple moment resisting portal frame structure, and a slightly more complex 3-story, 3-bay, moment resisting frame structure. In this regard, the fidelity and robustness of gradient boosting in classifying the damage patterns are verified by considering different levels of uncertainties associated with observations of structural responses and different numbers of observed structural responses. Furthermore, as numerical simulations as well as observations of failure events usually result in an imbalanced database in the failure events, a novel structured Markov chain Monte Carlo upsampling scheme is introduced to balance such databases.

RESUMÉ

Indenfor ingeniørvidenskaben anerkendes det i stigende grad at traditionel systemmodellering har et betydeligt forbedringspotentiale. Hovedformålet med denne afhandling er således at introducere og videreudvikle på den seneste forskning inden for probabilistisk modellering og analyse af komplekse systemer, som har relevans for offshore ingeniørsamfundet. Dette indebærer en demonstrering af, hvordan moderne maskinlæringstilgange, såsom Bayesianske netværk, Gaussiske processer og neurale netværk kan anvendes til at definere videns- og informationskonsistente modeller inden for domænet. Sådanne modeller skal ikke alene kunne afspejle middeltendenser i modeldomænet, men også konsistent kunne udnytte de tilgængelige datakilder, f.eks. eksperimenter, målinger, simuleringer og prognoser, samt kunne repræsentere og propergere de relaterede usikkerheder. Det understreges, at selvom denne forskning demonstreres i relation til offshore ingeniørvidenskaben, er den fuldstændig generisk og gælder i princippet for ethvert anvendelsesområde, hvor komplekse systemer forekommer.

Ved at værdsætte at modelopbygning i sagens natur er en subjektiv opgave, der afhænger af modelbyggerens bias mod og viden om eksisterende modelleringstilgange, såvel som de tilgængelige eller valgte data til modelopbygningen, ender man ofte med et sæt af relevant modelhypoteser, som på lige fod kan forklare data. Samtidig skal en modelleringsindsats altid ses i forhold til de beslutninger, hvilke modellerne tjener til at støtte, dvs. det er bydende nødvendigt, at en systemrepræsentation understøtter den aktuelle beslutningskontekst på den bedst mulige måde, når der udføres beslutningsoptimering. Denne forskning imødekommer disse overvejelser ved at introducere praktiske, statistiske og beslutningsteoretiske tilgange til håndtering af konkurrerende systemrepræsentationer i inferensmodellering og beslutningsoptimering.

For at skitsere den praktiske anvendelse af de introducerede systemmodelleringstilgange, er der oprettet et GitHub-arkiv med programmer og kodeeksempler, der er udviklet i løbet af ph.d.-projektet. Dette inkluderer programmer udviklet til automatisk læring af Bayesianske netværk, som er anvendt i flere af forskningsartiklerne, samt kodeeksempler vedrørende programmernes anvendelse. Til systemrepræsentation er der yderligere kodeeksempler tilgængelige omhandlende blandt andet Gaussiske processer, neuralt netværk og gradientforstærkede maskiner. Til systemanalyse er der adskillige kodeeksempler tilgængelige vedrørende klynge- og føl-

somhedsanalyse, og der findes også kodeeksempler vedrørende koncepter såsom hyperparametervalg, samt modelvalg og -gennemsnit.

Forskningsartiklerne introducerer et antal principeksempler, hvor systemmodelleringstilgangene anvendes i praksis. Disse eksempler illustrerer forskellige aspekter, der er relevante, når man bygger sandsynlighedsmodeller til offshore ingeniørsamfundet. For det første formuleres sandsynlighedsmodeller for offshore stormhændelser baseret på domæneviden samt målinger og simuleringer. I denne forbindelse anvendes Bayesianske netværk til at formulere beregningseffektive, kategoriske modeller for stormhændelser, og Gaussiske processer benyttes til at repræsentere forskellene mellem målinger og simuleringer. Denne forskning viser også, hvordan man for en given beslutningskontekst konsekvent kan tage højde for konkurrerende systemrepræsentationer. For det andet defineres probabilistisk modeller for metaltræthedsrevnedannelse i svejste stålsamlinger. Dette arbejde anvender Bayesiansk, hierarkisk modellering til at repræsentere data fra forskellige laboratorieeksperimenter med hver deres måleusikkerhed, og Bayesiansk modelgennemsnit til statistisk inferens i tilfælde af nye samlingsdetaljer. Denne fremgangsmåde muliggør en systematisk håndtering af usikkerhederne, som kan skaleres til større datasæt.

Forskningsartiklerne introducerer ligeledes et antal principeksempler vedrørende analyse af komplekse systemrepræsentationer. For det første undersøges potentialet af klynge- og følsomhedsanalyse på en simpel, momentbestandig portalrammekonstruktion. Denne forskning viser, hvordan modelbaseret klyngedannelse kan anvendes til at etablere en probabilistisk repræsentation af realiseringer, som giver anledning til bestemte systemadfærdsmønstre. Dette bidrager med betydelig indsigt i de enkelte systemadfærdsmønstre med hensyn til f.eks. latente grupperinger i realisationerne og drivende variable. Endvidere viser forskningen, hvordan variansbaseret følsomhedsanalyse kan benyttes til at nedbryde variabiliteten i systemresponser i bidrag fra de enkelte, stokastiske systeminput, hvilket f.eks. kan anvendes til at vurdere effekten af de enkelte systeminput på de stokastiske systemresponser. For det andet præsenteres en modeltilgang til systemidentifikation i en skadesdetekteringskontekst for den momentbestandig portalrammekonstruktion og en lidt mere kompleks 3-etager, 3-enheder momentbestandig rammekonstruktion. Disse analyser bekræfter nøjagtigheden og robustheden af gradientforstærkede maskiner in relation til klassificering af skadesmønstrene ved forskellige niveauer af usikkerhed forbundet med strukturelle observationer, samt forskellige antal observerede strukturelle observationer. Idet numeriske simuleringer såvel som observationer af skadeshændelser normalt resulterer i en ubalanceret database af skadeshændelser, introduceres der en ny struktureret Monte Carlo simuleringsprocedure baseret på Markov-kæder til at balancere sådanne databaser.

PREFACE

A famous quote in statistics by George E. P. Box goes

> *"all models are wrong but some are useful"*
>
> — *Robustness in the strategy of scientific model building*, 1979 (p. 202).

Taking this perspective, it becomes increasingly important to reflect the quality of scientific models in terms of model fit and especially the uncertainty associated with models. Moreover, since all models are wrong Box advocates that Occam's razor applies for model selection, and thus we should seek an economical description of natural phenomena, often termed a parsimonious model representation.

In the same vein, Vladimir N. Vapnik has pointed out that

> *"if you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate step. It is possible that the available information is sufficient for a direct solution but is insufficient for solving a more general intermediate problem"*
>
> — *Statistical learning theory*, 1998 (p. 12),

and Judea Pearl emphasizes that

> *"you are smarter than your data. Data do not understand causes and effects; humans do"*
>
> — *The book of why: the new science of cause and effect*, 2018 (p. 21).

On the other hand, so-called black-box models, like deep neural networks, are dominating the scene for large and highly complex machine learning applications, such as image classification, recommender systems, computer vision and natural language processing. Deep neural networks emulate the

way the human brain processes information by activation impulses, and thus the fitting of such system representations can to some extend be compared to the development of the human brain from cradle to grave – like neural networks we too learn by a huge amount examples.

As apparent from the considerations above, there are a lot of opinions on and approaches to scientific modeling, and new algorithms and modified variants of existing algorithms continue to appear in scientific publications. This makes it increasingly difficult for newcomers to navigate in the field of probabilistic modeling, as formulated by Christopher M. Bishop

> "*the popularity and importance of machine learning means that it has moved beyond the domain of the machine learning community to the point where many researchers whose expertise lies in other fields, such as the physical and biological sciences, statistics, medicine, finance and many others, are interested in solving practical problems using machine learning techniques. The variety of algorithms, as well as the complex nomenclature, can make the field challenging for newcomers.*"
>
> — *Model-based machine learning*, 2013 (p. 3),

and Leo Breiman further states that

> "*The best available solution to a data problem might be a data model; then again it might be an algorithmic model. The data and the problem guide the solution. To solve a wider range of data problems, a larger set of tools is needed.*"
>
> — *Statistical modeling: The two cultures*, 2001 (p. 204).

Moreover, it is appreciated by Peter McCullagh and John A. Nelder that

> "*Data will often point with almost equal emphasis at several possible models and it is important that the statistician recognize and accept this.*"
>
> — *Generalized linear models*, 1989 (p. 8).

Based on these reflections, this thesis strives to bring recent developments in probabilistic modeling and analysis to the challenging field of offshore engineering by showcasing how current state-of-the-art machine learning frameworks can be applied within the domain to formulate knowledge- and information-consistent scientific models relevant for offshore design and assessments. This generally involves multi-scale, temporal, and spatial considerations for multiple, interrelated variables. In this regard, the thesis considers the frameworks used in detail and elaborates on how to analyze response

characteristics of systems in order to understand these and improve their representation in terms of modeling. As models are generally built to support decision-making, it is crucially important that system representations support the decision context in the best way possible when performing decision optimization. Thus, the thesis considers feasible approaches for embedding probabilistic modeling within the decision problem in order to choose the best possible system representation for the decision context at hand, and for quantifying the value of existing and additional information as well as and model improvements.

The thesis is composed of two part: Part I frames the thesis and elaborate on the theory used in the accompanying research papers. This part, named *Treatise*, is meant as a general introduction to systems modeling and analysis, as many technical details are omitted in the research papers. In this way, I have attempted to make the thesis as self-contained as possible. Part II contains a set of published papers demonstrating applications of the theory (Part I) on practical engineering problem of relevance within the context of offshore engineering. Only papers where I act as the lead author are included in this part, which is named *Papers*. The papers in this part are individual works with their own reference list and section, figure, table, and equation numbering. Therefore, similar bits and pieces appear in more than one paper. Note that I have tried to keep Part I inclusive by use of "we" to referrer to you (the reader) and me, whereas in Part II, "we" refers to my co-authors and me. Moreover, in support of the study, a GitHub repository hosted at `https://github.com/SebastianGlavind/PhD-study` has been created, which includes code examples on most of the theory presented and introduces the use of two toolboxes for Bayesian network learning that I have developed during the course of this study.

ACKNOWLEDGEMENTS

First of all, I would like to thank my main supervise Michael Havbro Faber for giving me the opportunity to conduct this research. It has been a great pleasure working with you on all the many interesting aspects of systems modeling and analysis that we have touched upon, and I am deeply grateful for the way you have introduced me to your professional network by always bringing me along to meetings, workshops etc., whenever you thought I had something to contribute to the discussion. Michael has been the perfect supervisor by giving me the freedom to explore and discover the areas of probabilistic artificial intelligence, always pushing me to do my best and giving me the inputs needed to make this work something I am proud of. A special thanks goes to Michael, his wife Linda Nielsen, and their son Jasper for always opening their home to me, and making me feel like home, when-

ever I visit Aalborg.

I am thankful to my co-supervisors John Dalsgaard Sørensen and Bo Friis Nielsen for their valuable insides and interest in contributing to the project. I would of cause have loved to collaborate more with both of you, but this is surely on my agenda for future research! Also, I would like to show my gratitude to Erik Damgaard Christensen for welcoming me to his research group at DTU Mechanical Engineering under the current COVID-19 situation; thank you for the inspiring collaboration and for sharing your expertise within wave hydrodynamics with me. This collaboration is just getting started, and I am looking forward to the upcoming joint research initiatives.

My research colleagues and friends at Aalborg University (AAU) and Centre for Oil and Gas – DTU (DHRTC) also need my acknowledgements; thanks to Juan G. Sepulveda and Henning Brüske for the joint research and our academic as well as private discussions — it has been a pleasure working with you both. Thanks to Jesper Sören Dramsch for our countless discussions on probabilistic modeling and for introducing me to neural network modeling with TensorFlow. I would also like to thank the members of the Risk, Resilience and Sustainability in the Build Environment (R2SBE) research group at AAU, which counts, besides the people already mentioned, José Guadalupe Rangel Ramirez, Jianjun Qin, Kashif Ali, Yue Guan, Akinyemi Olugbenga Akinsanya and Min Liu.

No research could have been conducted without funding. Therefore, I am gratefully for the funding received from Centre for Oil and Gas – DTU / Danish Hydrocarbon Research and Technology Centre (DHRTC), and I am thankful to the Danish Underground Consortium (Total E&P Denmark, Noreco and Nordsøfonden) for providing data and granting the permission to publish the research. In this regard, Total E&P Denmark, Danish Hydraulic Institute, Haw Metocean (Hans Fabricius Hansen), and Shell Research Ltd. needs a special thanks for their support in this project; without their help, the research would not have been as targeted and relevant.

Finally, there is one very important group of people that needs my acknowledgements, namely my family and friends. I would like to thank all my friends for taking my mind off research once a while. I am gratefully to both my parents Marianne and Claus and my in-laws Anne and Lars for all their support throughout the years; it has really meant a lot to me. Last, and most importantly, I am at a loss for words on how to to express my gratitude to my lovely wife Maria for her patience, understanding and support and to our two sons Hugo and Isak for reminding me of the things that are truly important in life.

Sebastian T. Glavind
Copenhagen, April, 2021

CONTENTS

PART I

TREATISE

ON SYSTEMS MODELING AND ANALYSIS

1 INTRODUCTION

1.1 BACKGROUND, MOTIVATION AND PURPOSE[1]

The Danish oil and gas reserves are by now rapidly decreasing, whereby the efficiency in recovery of the remaining oil and gas resources plays a significant role in the extent to which the Danish state can benefit from these reserves – and for how long. In appreciation of this, the partners of the Danish Underground Consortium (DUC; Total E&P Denmark, Noreco & Nordsøfonden) established the Danish Hydrocarbon Research and Technology Centre (DHRTC)[2] at the Technical University of Denmark (DTU) in 2014, in collaboration with other universities and industrial partners in Denmark and abroad. The main objectives of this DKK 1 billion investment over 10 years are to identify possible technical means to increase the recovery efficiency of the remaining oil and gas resources, to facilitate sustainable, cost-effective exploitation activities, and, at the same time, to ensure that the safety level for personnel, environment, and assets comply with given acceptance criteria.

Integrity management of the offshore structures presently in operation constitutes in itself a major challenge, as most structures were built during the period 1970-1980 and typically with design service lives of around 20-30 years. Thus, many of these structures are still in operation despite exceeding their originally intended service lives. The situation is further complicated by new findings, which indicate that the original design assumptions regarding the offshore wave load environment in the Danish North Sea result in an underestimation of the extreme loads [1, 2]. This calls among others for the development of new methods and technology in support of asset integrity management to enable a more detailed modeling and analysis of structural

[1] https://vbn.aau.dk/da/projects/load-environment-modeling-and-forecasting
[2] https://www.oilgas.dtu.dk

performances, which in turn facilitates that the various uncertainties affecting these performances can be quantified and that the reliability and risks associated with the structures can be updated based on observations, inspections, and repair actions and be documented transparently to the responsible authorities.

To accommodate the effort on safety assurance of offshore structures, the main purpose for the present thesis is to bring resent advances on systems modeling and analysis to the offshore engineering community in pursuit of knowledge- and information-consistent probabilistic models relevant for offshore design and assessments. Such models should among others comply with the new requirements in relation to e.g., uncertainty, reliability and risk quantification.

1.2 THE MODELING PHILOSOPHIES

Model building often leads to a multitude of competing model hypotheses, due to limited amounts of data and the vast amount of modeling frameworks and associated parameter representations in the literature. Thus, a fundamental question in model building is how to choose a system representation, or model, for a given problem context.

Simple, or parsimonious, models are commonly advocated for and preferred to complex models when representing systems, as it is generally easier to express distributional assumptions for and interpret such models [3]. This preference is often expressed through Occam's razor, or Ockham's razor, due to the English monk and philosopher William of Occam (1288–1348) [4], who originally used the philosophy that simpler explanations should be preferred over complex ones to ground his reasoning in his faith. A variant of this principle, i.e.,

> "*We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances*",

also appear as the first rule of reasoning in philosophy in Isaac Newton's Principia Mathematica [5, p. 202]. Today, Occam's razor is cited in most textbooks on probabilistic modeling to guide model comparison and selection through Occam's and Bayes factor, see e.g., [4, 6].

The general challenge, though, with this selection bias towards simpler models is that model accuracy or data faithfulness should not be unduly compromised for the sake of simplicity [3]. This has led to a set of so-called anti-razors, expressed e.g., by Karl Menger [7, p. 415] as

> "*it is vain to try to do with fewer what requires more*".

This philosophy complies very well with today's deep learning applications, where the general solution to a poor model fit is to enlarge the model by

including additional hidden layers, which often result in models that have more parameters than data points available to fit the model, see e.g., [8].

In either case, the no free lunch theorem states that without any specific knowledge of the problem or the data at hand, i.e., the problem context, there are no reasons to support one modeling framework over another. At the same time, even if such knowledge is available, models customized for a specific situation, e.g., handling missing data, may be constraint in other ways, e.g., linear models, which may not allow the model to represent the underlying data distribution anyway. Thus, the model choice should be seen in consistency with the knowledge and information (data) at hand. To this end, the common approach is to test the generalization performance of the models against a data set that has not been used to fit the models using a measure of "average" or "overall" performance, like mean-squared error or classification accuracy [3, 9].

This procedure has proven to work well when the objective of modeling is to understand the overall data generating mechanism or to perform average predictions, but what is often overseen when specifying the problem context are the decisions about the system, the model aims to support. As an example, an overall score metric for model performance is of no value if the decision context relies only on the upper or lower tail of the model distribution. Also, a choice of system representation that does not account for the decision context will not accommodate a quantification of the true value of additional information in a decision analysis, as the model hypothesis space is already reduced to a specific model class. In the general setting, where the decision context is accommodated for in the model choice, utility theory provides a principled means for context-specific model selection and decision optimization. Early developments along this line are found in [10].

1.3 THESIS OBJECTIVES AND RESEARCH QUESTIONS

In response to the foregoing outline of the research purpose and context, the principal objectives of the present thesis are categorized into (i) representation of systems, (ii) analysis of systems, and (iii) model selection and decision optimization, and listed below along with their associated research questions.

Representation of systems: The objective is to formulate and develop probabilistic models of engineering systems, like the joint representation of the offshore load environment (e.g., wind, waves and current), which are relevant for design and assessments of offshore structures and facilitate for (i) a consistent utilization of available and potential future knowledge and information, and (ii) a consistent representation of the prevailing aleatory and epistemic uncertainties.

Q1: Which information sources are commonly used and available for

building offshore engineering models, and how can we utilize all available, and future, knowledge and information to this end?

Q2: What are current practices for representing offshore engineering systems and what can be improved?

Q3: What are the trends in state-of-the-art systems representation and how do they apply to (offshore) engineering systems?

Q4: How can we consistently account for the aleatory and epistemic uncertainties when formulating system representations?

Analysis of systems: The objective is to investigate how methods of big data analysis can be used as a means for understanding complex probabilistic system representations by revealing behavioral patterns and sensitivities in the model specification.

Q5: What are the trends in state-of-the-art big data analysis of complex systems and how do they apply to (offshore) engineering systems?

Q6: How can we consistently explore the space of possible, competing systems and rank their relevance in terms of occurrence?

Model selection and decision optimization: With basis in the early developments on context-specific model selection presented in [10], the objective is to assess whether these ideas can be operationalized and further developed to accommodate not only a selection among pre-specified models but also a full embedding of the model building operation in case of complex systems.

Q7: Can the framework of Faber & Maes [10] be operationalized to accommodate context-specific model selection for complex systems?

Q8: If so, how does it aid a situation of (i) pre-specified models, and (ii) embedding of the model specification?

1.4 THESIS OUTLINE

The thesis is composed of two parts: Part I is the treatise, which acts as a wrapper for the accompanying research papers, and Part II contains the research papers, which document the research conducted as part of this PhD study. Part I introduces the general frame for the papers and elaborates on the theory of some of the applied techniques, as the paper format only makes room for a glimpse of the mathematical details. Note that this part does not include a dedicated state-of-the-art literature survey; however, when found relevant, a survey appears in the papers of Part II.

In the coming sections of the treatise, we will start out with a general introduction to systems modeling and the terminology and notation used

throughout this study in Sec. 2. Next, Secs. 3–5 proceed to cover three of the frameworks for systems representation, which are used extensively throughout the study and part of the current state-of-the-art in machine learning. These are Bayesian networks (BNs), Gaussian processes (GPs), and (deep) neural networks (NNs), mentioned in order of decreasing explainability.

In order to understand and assess probabilistic system representations in the context of decision analysis, Sec. 6 considers how input-output relations leading to different performances of systems can be analyzed with this perspective. In this regard, we focus our attention on state-of-the-art methodologies from the global sensitivity analysis (SA) literature and show how these may be complemented by methods from the machine learning literature to enhance our understanding of complex systems. This section also elaborates on surrogate modeling using polynomial chaos expansions for efficient implementation of variance-based SA techniques, and model-based cluster analysis for regionalized SA.

As most real systems are complex to model in a way that reflect all aspects of the systems, model building may result in different system representations that explain the information (and knowledge) equally well. In these situations, tools for handling different (competing) systems are needed for subsequent decision optimization. Section 7 reviews some of the tools used in this study for handling model multiplicity in regard to the management of systems.

To finalize the treatise (Part I), Sec. 8 points to the papers in Part II, where among others the theory from Secs. 3–7 is used on practical engineering problems relevant for offshore engineering design and assessment. This is followed by Sec. 9 that concludes the thesis and points to future research. The treatise is visually outlined in Fig. 1 using the section titles.

The papers in Part II generally fall into two categories: (i) papers where the main objective is the representation of systems (Papers A, C, D, and F); and (ii) papers where the main objective is the analysis of system responses (Papers B and E). Both categories encompass the overarching envelope of decision support and particularly risk-informed integrity management. The papers in Part II are listed below.

Paper A: Sebastian T. Glavind and Michael H. Faber, "A framework for offshore load environment modeling", in *Proceedings of the ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering (OMAE2018)*, OMAE2018-77674, 2018.

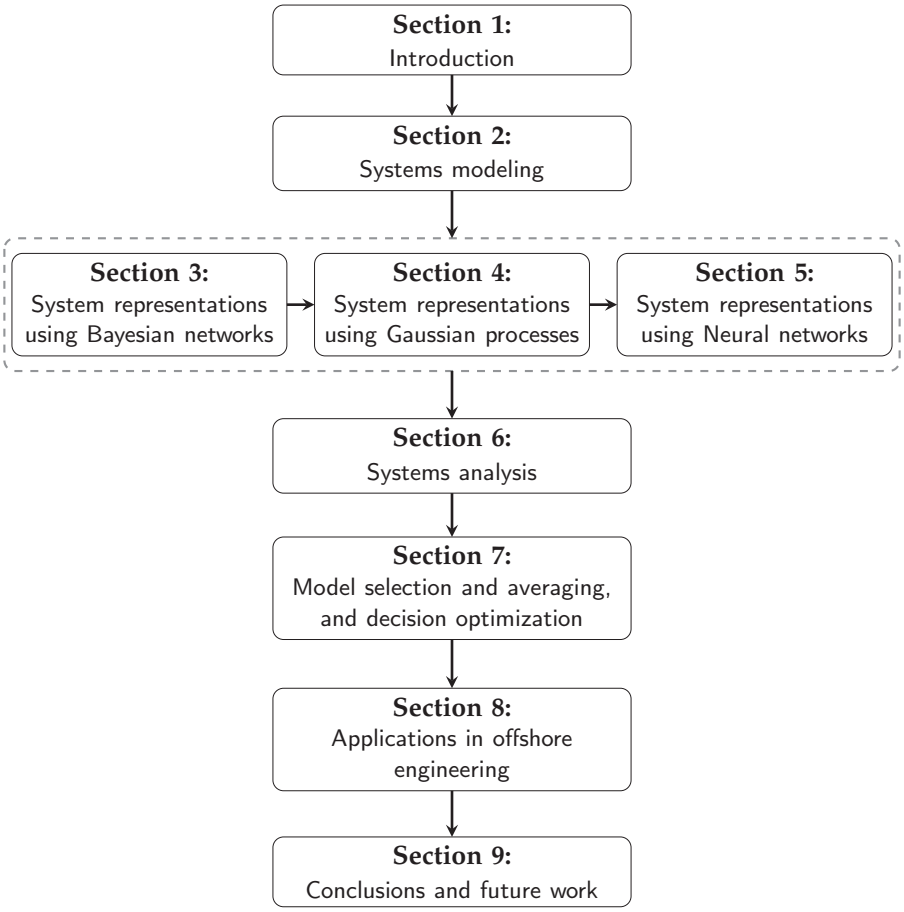


Figure 1: Outline of the treatise.

Paper B: Sebastian T. Glavind, Juan G. Sepulveda, Jianjun Qin and Michael H. Faber, "Systems modeling using big data analysis techniques and evidence", in *Proceedings of the IEEE 2019 4th International Conference on System Reliability and Safety (ICSRS2019)*, ICSRS2019-R0119, 2019.

Paper C: Sebastian T. Glavind and Michael H. Faber, "A framework for offshore load environment modeling", *Journal of Offshore Mechanics and Arctic Engineering*, vol. 142, no. 2, pp. 021702, OMAE-19-1059, 2020.

Paper D: Sebastian T. Glavind, Henning Brüske and Michael H. Faber, "On normalized fatigue crack growth modeling", in *Proceedings of the ASME 2020 39th International Conference on Ocean, Offshore and Arctic Engineering (OMAE2020)*, OMAE2020-18613, 2020.

Paper E: Sebastian T. Glavind, Juan G. Sepulveda and Michael H. Faber, "On a simple scheme for systems modeling and identification using big data techniques", submitted to *Reliability Engineering & System Safety*.

Paper F: Sebastian T. Glavind, Henning Brüske, Erik D. Christensen and Michael H. Faber, "On systems modeling and context-specific model selection in offshore engineering", submitted to *Computer-Aided Civil and Infrastructure Engineering*.

If not stated otherwise, I carried out the research documented in the aforementioned papers, and co-authors assisted in an advisory role. In Paper B, Juan G. Sepulveda supplied the finite-element program for the portal frame structure and conducted the simulations for the study. In Paper D, Henning Brüske used the developed normalized fatigue crack growth model for risk-based inspection planning. In Paper E, Juan G. Sepulveda supplied the finite-element program for the portal frame structure, and the 3-story, 3-bay frame structure, and conducted the simulations for the study.

In addition to the papers in Part II, I have also contributed to the following publication, which is not included in the thesis:

Paper: Linda Nielsen, Sebastian T. Glavind, Jianjun Qin and Michael H. Faber, "Faith and fakes – dealing with critical information in decision analysis", *Civil Engineering and Environmental Systems*, vol. 36, no. 1, pp. 32-54, 2019.

Note that Paper B received the session award of *Network and Data Security* at the *4th International Conference on System Reliability and Safety (ICSRS2019)*,[3] and the aforementioned additional paper by Nielsen et al. received the 2019 best paper award of the journal *Civil Engineering and Environmental Systems*.[4]

Finally, the thesis is supplemented by a GitHub repository hosted at `https://github.com/SebastianGlavind/PhD-study`, which includes code examples on the techniques discussed in this thesis and introduces the use of two toolboxes developed during the PhD study.

[3] `http://www.icsrs.org/icsrs19.html`

[4] `https://think.taylorandfrancis.com/journal-prize-civil-engineering-and-environmental-systems-best-paper-award/`

2 SYSTEMS MODELING

This section frames the concept of systems modeling as considered in this study and provides a common ground for discussion in terms of terminology and notation. Section 2.1 introduces the basis for systems modeling along with the different components of the modeling, which are elaborated on in subsequent sections. The study employs techniques from different technical disciplines, e.g., probability theory, statistics, machine learning, and decision theory; thus Secs. 2.2 and 2.3 provide some general terminology and notation, respectively, used throughout the remainder of the study.

2.1 MODELING BASIS

Parts of this section appear in Papers A, and C.

Knowledge and information form the basis for representing systems subject to decision optimization. Thus, before proceeding on the topic of model development, we will start out with a brief outline on how we account for this modeling basis. Following the guideline for system representations proposed by the Joint Committee on Structural Safety (JCSS) [11], Fig. 2 provides a system representation in terms of the flow of consequences generated as a result of exposure events (Paper C).

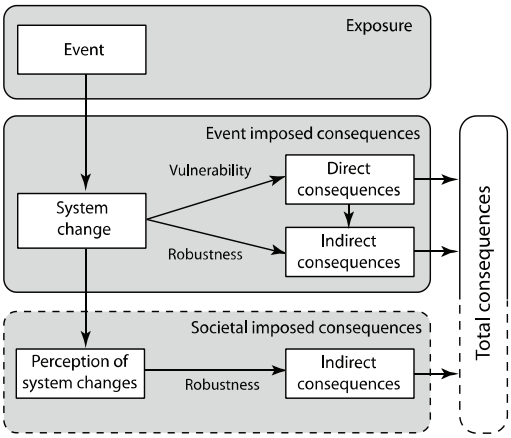


Figure 2: The JCSS systems representation [11].

In the top of Fig. 2, we consider an exposure event (e.g., natural hazard, act of terrorism, etc.) acting on the constituents of the system. This exposure event may lead to system changes, as depicted in the middle part of the figure, in terms of direct consequences (e.g., loss of lives, damages to in-

frastructure, etc.) and indirect consequences (e.g., loss of business continuity, repetition loss, etc.). In this regard, vulnerability is associated with the risk imposed by direct consequences, and robustness relate to the risk of event-imposed consequences exceeding the direct consequences. The lower part of the figure shows an additional source of indirect consequences, namely consequences attributed to the societal or public perception, as a result of system changes. This part indicates how risk management relies on risk communication, i.e., efficient risk communication before, during, and after e.g., events of natural hazards can mitigate consequences associated with perception (e.g., loss of trust) [11, 12].

We note that depending on the granularity of the risk assessment, the exposure, constituents, and consequences are different. As an example, a system could be an oil and gas field in the North Sea with constituents being the different offshore facilities in the field (e.g., production and accommodation platforms, etc.), or a system could be one of the offshore facilities in the field with constituents being e.g., the structural members. For one such offshore facility, a decision problem concerning the structural design would consider exposure events in terms of the operational and environmental loads acting on the structure, and the constituents would include e.g., the structural components and joints. The direct consequences would be e.g., material damages and loss of lives, and the indirect consequence would be e.g., monetary consequences imposed by the loss of functionality of the structure and additional material losses imposed on the surrounding assets [11].

The flow of consequences and their magnitude are generally subject to both aleatory (type I) and epistemic (type II) uncertainty, where aleatory uncertainty represents inherent variation associated with the system, i.e., it can be described as intrinsic, irreducible randomness. Epistemic uncertainty is uncertainty caused by lack of knowledge (or information) about the system, i.e., it can be described as subjective uncertainty, which may be reduced by e.g., better models and/or more data [13–15]. In accordance with JCSS [11], the uncertainties may be adequately represented by means of Bayesian probability theory, whereby Bayesian decision analysis [16] and the axioms of von Neumann and Morgenstern [17] provide the necessary means for decision optimization regarding the management of systems, as illustrated in Fig. 2, in support of the available knowledge and information about how decisions change the generation of the expected value of consequences.

Nielsen et al. [12] point out that the knowledge and information relevant to consider when establishing a probabilistic system representation (model) are the knowledge and information which affect the identification of optimal decisions, i.e., the ranking of decision alternatives. With this insight, the process of model building and systems management, i.e., the context of the model building, should be considered jointly and not as separate tasks (Paper C). To this end, given a decision context, systems modeling

includes (i) knowledge- and information-consistent system representation(s), (ii) analysis of system performances, and (iii) context-driven optimization of system performance management decisions. Thus, these elements should constitute a joint (integrated) system formulation. The individual elements are as mentioned elaborated on in subsequent sections, see Sec. 1.4.

2.2 MACHINE LEARNING TERMINOLOGY AND CONCEPTS

Statisticians and machine learners generally use different terminology for the same concepts. For instance, statisticians often use covariates, predictors, or independent variables for the inputs to a model, whereas machine learners often use features, or attributes, and the model output is commonly referred to as outcome, response, dependent variable, or target. Fitting a model to data is often termed estimation in statistics and learning in machine learning (ML), where the data are typically given a label like training, validation, and testing, depending on where in the learning process it is used. In this regard, the data points in a data set are commonly referred to as samples, observations, or instances [18, 19]. Throughout this study, we will use these terms interchangeably.

Probabilistic models are generally built to support decision-making within a given problem domain (area of expertise or application). In this regard, ML models can be grouped into two categories, namely generative and discriminative models. Generative models provide a joint distribution model for the random variables in the problem domain, and discriminative models, also referred to as conditional models, fit the posterior distribution of some variable(s) in the problem domain, i.e., the outputs, based on other variable(s) in the problem domain, i.e., the inputs, which are assumed to be given, and thus their distribution is not modeled [6]. The discriminative approach is appealing when data are limited, and we are targeting a specific output variable or set of output variables. As argued by Vapnik [20, p. 12], we should

> "*try to solve the problem directly and never solve a more general problem as an intermediate step*".

However, the generative approach offers a principled means for dealing with real data, where e.g., some input values may be missing and/or outliers may be present. Thus, in accordance with the no free lunch theorem, the most appropriate modeling approach is problem dependent [3, 21].

Applications in which the training data consists of observations of input vectors and corresponding output vectors are known as supervised learning problems. If the outputs are continuous-valued, i.e., variables with a continuous sample space, then the task is called regression, and if the outputs are

discrete-valued, i.e., variables with a discrete sample space, then the task is called classification [6]. An example of a regression problem considered in this study is the prediction of fatigue crack growth for fatigue sensitive details, see Paper D, and an example of a classification problem is damage class identification in a structural health monitoring context, see Paper E.

Applications in which the training data consists of observations of input vectors without corresponding output vectors are known as unsupervised learning problems. The objective of such problems can be to reflect latent groups of similar samples within the data, typically referred to as cluster analysis or clustering, to define a distribution based on the data samples from input space, known as density estimation, or to reduce the dimensionality by means of projection for e.g., visualization purposes [6]. An example of a clustering problem considered in this study is the discovery of most likely failure points in structural reliability, and the corresponding systems sensitivity towards the individual inputs, see Paper B.

Whether we are building a probabilistic model for supervised or unsupervised learning, we need to decide on a modeling scheme. Do we want a model with a specific functional form defined by a small number of parameters, which are estimated from the data set? Or do we want a more flexible model for which the functional form depends on the size of the data set? The former is named a parametric model, which e.g., has the advantage of being fast to train and use but the disadvantage of making strong assumptions about the underlying data distribution. The latter is named a non-parametric model, which as noted is more flexible but often computationally intractable for large data sets in which case approximation schemes are needed. Non-parametric models also have parameters, but these define the model complexity rather than the form of the distribution [4, 6]. An example of a parametric regression model considered in this study is again the prediction of fatigue crack growth for fatigue sensitive details, see Paper D, and an example of a non-parametric regression is the discrepancy modeling performed in Paper F, where the discrepancy between a hindcast data set and a corresponding data set of observations is modeled.

As noted earlier, an important consideration when building models is the choice of model complexity. Thus, we want a model that reflects all information on the underlying data distribution contained in the data set, but, at the same time, we do not want to include the observation noise contained in the data in our model. This is referred to as the bias-variance trade-off, see e.g., [3, 18]. Failing to reflect all information on the underlying data distribution is generally referred to as underfitting, and adaption of the data distribution model to observation noise is referred to as overfitting. Underfitting occurs when the considered probabilistic model cannot adequately capture the underlying structure of the data due to too limited flexibility, and overfitting occurs when the probabilistic model contains more parameters than

can be justified by the data. Measures to prevent overfitting are referred to as regularization measures, which have an equivalent prior interpretation in a Bayesian setting, see e.g., [6] for further details.

2.3 GENERAL NOTATION

In this study, we consider both discrete and continuous random variables. In either case, we denote a random variable by an uppercase letter, e.g., X, Y, Z, and a generic (realized) state or value of a variable by that same letter in lowercase, e.g., x, y, z. Moreover, we denote a set of random variables by a boldface uppercase letter, e.g., $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$, and a boldface lowercase letter, e.g., $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, denote a generic assignment of state or value to each variable in a given set. Thus, we may also refer to a set of random variables $\boldsymbol{X}$ as being in configuration $\boldsymbol{x}$ [22, 23].

For discrete random variables, $P(X = x)$, or simply $P(x)$, refers to the probability that $X = x$, i.e., the probability mass at $X = x$; and for continuous random variables, $p(X = x)$, or simply $p(x)$, refers to the probability density at $X = x$. When we discuss discrete, categorical random variables, we use the notation $x \in \text{Val}(X) = \{x^1, x^2, ..., x^{|X|}\}$, when we need to enumerate the possible values of X. Other shorthand notations used in this study are $\sum_x$ to refer to a sum over all possible values that X can take and $P(X = x, Y = y)$, or simply $P(x, y)$, to refer to the conjunction $P((X = x) \cap (Y = y))$. Moreover, we will use the notation $P(X|Y)$ to represent the set of conditional probability distributions defined by the two random variables X and Y. Thus, for each value of Y, this object assigns a conditional probability distribution over X. Note that these definitions equally apply to sets of random variables [6, 24].

When we consider a data set of observations of a set random variables, we will denote it $\mathcal{D}$, and the corresponding number of observations will be denoted by N, e.g., $\mathcal{D} = \{\boldsymbol{x}[n]\}_{n=1}^{N}$ or $\mathcal{D} = \{\boldsymbol{x}[n], y[n]\}_{n=1}^{N}$. Moreover, the number of observations in $\mathcal{D}$ for which the random variable X takes the value x is denoted $N[x]$. Sometimes it is easier to consider our data set as matrix quantities. In these situations, the data set may equivalently be defined as e.g., $\mathcal{D} = \{\hat{\boldsymbol{X}}, \hat{\boldsymbol{y}}\}$, where $\hat{\boldsymbol{X}} \in \mathbb{R}^{N \times M}$ is referred to as the design matrix of N observations in M input variables, and $\hat{\boldsymbol{y}}$ is a corresponding vector of N observations of output variable Y.

3 SYSTEM REPRESENTATIONS USING BAYESIAN NETWORKS

A Bayesian network (BN) is a probabilistic graphical model that allows for reasoning and learning in complex, uncertain domains. In this regard, reasoning refers to the task of performing probabilistic inference one or several variable(s) in the problem domain, e.g., querying the (conditional) distribution of a variable, potentially given observations on some other variables in the model. Learning refers to the task of specifying the BN model, i.e., model structure and parameters given a training data set (Paper F).

This section sets off by introducing BNs and their semantics in Secs. 3.1 and 3.2. Next, Sec. 3.3, accompanied by Appendix A, covers inferences in BNs, and Sec. 3.4 considers learning of discrete BN representations, including optimal discretization policies, based on fully observed and partially observed data sets, respectively. Finally, Sec. 3.5 discusses the concept of template modeling for structured data, e.g., populations of similar groups or temporal systems, which naturally leads to the introduction of an emerging methodology for applying machine learning, called model-based machine learning, in Sec. 3.6. Note that two toolboxes have been developed during the PhD study to support the research on probabilistic system representations using BNs. The toolboxes, along with supporting tutorials on their use, are available at the GitHub repository.

3.1 INTRODUCTION TO BAYESIAN NETWORKS

Parts of this section appear in Papers A, C, D, and F.

BNs define a joint probability distribution over a set of random variables $\boldsymbol{X}$ by decomposing it into a product of local, conditional probability distributions according to a directed acyclic graph (DAG) $\mathcal{G}$, i.e., the model structure. In the DAG $\mathcal{G}$, each vertex or node corresponds to a random variable $X_i \in \boldsymbol{X}$, and the edges between the nodes represent the set of direct dependence relations implied by $\mathcal{G}$. In the following, we use X_i to denote both variable i and its corresponding node in $\mathcal{G}$. Now, by studying the edges and missing edges in $\mathcal{G}$, we can directly read off a set of (conditional) independence assertions about the domain variables in $\boldsymbol{X}$. For each random variable X_i represented in $\mathcal{G}$, we specify a conditional probability distribution $P(X_i|\mathbf{Pa}_i)$. This distribution defines the dependence of X_i on the random variables that X_i is directly dependent on in $\mathcal{G}$, termed the parent set $\mathbf{Pa}_i$ of variable X_i, see e.g., [25, 26].

The joint distribution defined by a BN is written as

$$P(\boldsymbol{X}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}) = \prod_i P(X_i|\mathbf{Pa}_i), \tag{1}$$

where $\boldsymbol{\Theta}_{\mathcal{G}}$ denotes the set of model parameters. For discrete random variables, the set of parameters corresponds to the probability masses of each combination of states, and for continuous random variables, the parameter set corresponds to the parameters needed to specify the probability density functions of the random variables. In a supervised learning setting, the local distribution function $P(x_i|\mathbf{pa}_i)$ may be regarded as a probabilistic classification or regression function. Thus, a BN can be viewed as a collection of probabilistic classification/regression models, organized by conditional independence relations, and in principle, any combination of models from the supervised learning literature can be used to define the conditional probability distributions of a BN [23] (Papers A, C, and D).

The language of BNs combines ideas from probability theory (calculus) and computer science (data structures, i.e., graphs and algorithms for exploiting them) in order to model and reason about complex domains in an efficient manner. Thus, while BNs can represent arbitrary, unique probability distributions, they provide computational advantages for distributions that allow for a simple structural representation. BNs are particularly useful when we wish to reason about multiple interrelated variables simultaneously, and not just a single target variable, in which case other machine learning frameworks may be better suited, see e.g., Secs. 4 and 5. Furthermore, BNs provide a natural way of incorporating prior knowledge on both the structure and parameters into the modeling. This is in contrast to most traditional machine learning frameworks, where it can be difficult to embed prior knowledge in a natural way, see e.g., [23, 24].

3.2 REPRESENTATION

Parts of this section appear in Paper A.

The semantics of BNs are easiest explained by example. Figure 3 shows a classic example from [25], which illustrates how Mr. Holmes is reasoning about his burglary alarm A going off. If the alarm goes off, his neighbor Dr. Watson W may call him. A triggering of the alarm will have one of two causes: (i) there is a burglar B in his house, or (ii) there is an earthquake E in the area. Moreover, Holmes may gain additional information on the earthquake scenario by listening to the radio news R. The joint distribution, which factorizes according to Fig. 3, is written as

$$P(B, E, A, R, W) = P(B)P(E)P(A|B, E)P(R|E)P(W|A).$$

Now, imagine that Holmes is out and gets a call from Watson, who has heard Holmes' alarm going off. Holmes rushes to his car believing that a burglar has triggered the alarm. On his way home, the radio news reports an earthquake in the area. This additional piece of information makes him change

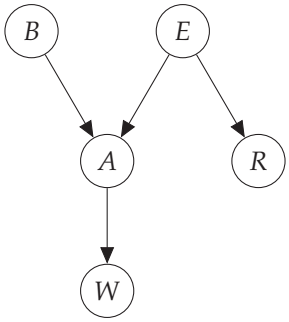


Figure 3: Burglary or earthquake Bayesian network (Paper A).

his belief in the burglary scenario, as the reported earthquake "explain away" the triggered alarm (Paper A).

The BN in Fig. 3, as every BN, is constructed from three distinct connection types: serial, e.g., $B \rightarrow A \rightarrow W$ and $E \rightarrow A \rightarrow W$; diverging, e.g., $A \leftarrow E \rightarrow R$; and converging, e.g., $B \rightarrow A \leftarrow W$. Based on these connection types, the rules for flow of information in a BN can be formulated under one criterion known as d-separation, which defines when information cannot flow between two variables. Two distinct variables X and Z in a BN are said to be d-separated if for all trails between X and Z, there is an intermediate variable Y (distinct from X and Z) such that either

- the connection is serial (Fig. 4a) or diverging (Fig. 4b) and the state of Y is observed, or
- the connection is converging and neither Y nor any of Y's descendants (variables descending from Y in $\mathcal{G}$) have received evidence (Fig. 4c).

If X and Z are not d-separated, they are d-connected, and information can flow between them. The blocking of information flow between variables, which appears in the first item above, reflects the concept of conditional independence, i.e., $X \perp Z|Y$; and the blocking of information flow, which appears in the second item above, reflects the concept of (unconditional) independence, i.e., $X \perp Z$. It should further be noted that the d-separation criterion also applies to disjoint sets of variables [27, 28].

3.3 INFERENCE IN BAYESIAN NETWORKS

Parts of this section appear in Papers C, and D.

In this section, we consider how to use the framework of BNs to answer queries on a subset of the variables in a BN, potentially given evidence (observations) on some of the other variables in the network. There are many

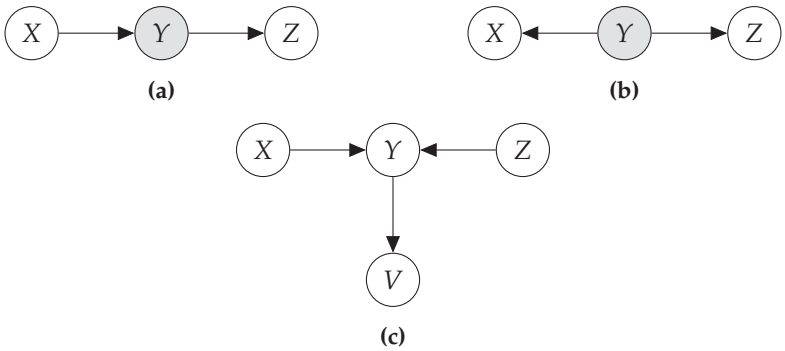


Figure 4: D-separation in Bayesian networks: (a) serial connection with observed intermediate variable Y, (b) diverging connection with observed intermediate variable Y, and (c) converging connection with unobserved intermediate variable Y and descendant V of Y. Colored vertices mark observed variables.

queries we can use a BN to answer, but the two most common types are the conditional probability query and the maximum a posteriori (MAP) query, both of which we explain below.

CONDITIONAL PROBABILITY QUERIES

A conditional probability query evaluates the posterior distribution $P(\boldsymbol{Y}|\boldsymbol{E}_v = \boldsymbol{e}_v)$ of a subset $\boldsymbol{Y}$ of the variables in a BN, given a (possibly empty) evidence set $\boldsymbol{E}_v = \boldsymbol{e}_v$ on some of the other variables in the network. By the definition of conditional probability, we may write this probability distribution as

$$P(\boldsymbol{Y}|\boldsymbol{E}_v = \boldsymbol{e}_v) = \frac{P(\boldsymbol{Y}, \boldsymbol{e}_v)}{P(\boldsymbol{e}_v)}. \tag{2}$$

In Eq. 2, the numerator is computed from the factorization of the joint distribution $P(\boldsymbol{X})$, defined by the BN, by marginalizing out the latent variables $\boldsymbol{W} = \boldsymbol{X} - \boldsymbol{Y} - \boldsymbol{E}_v$, which are neither query nor evidence variables, i.e.,

$$P(\boldsymbol{Y}, \boldsymbol{e}_v) = \sum_{\boldsymbol{W}} P(\boldsymbol{Y}, \boldsymbol{W}, \boldsymbol{e}_v). \tag{3}$$

Here, we assume that the variables are discrete, but the considerations in this section apply equally well to continuous variables, or to a combination of discrete and continuous variables, in which case, the summations are replaced, where appropriate, by integrals.

Now, because $\boldsymbol{Y}$, $\boldsymbol{W}$, and $\boldsymbol{E}_v$ are all the variables in the BN, each term in the summation $P(\boldsymbol{y}, \boldsymbol{w}, \boldsymbol{e}_v)$ is simply one entry in the joint distribution. The

denominator in Eq. 2 may then be computed as

$$P(\boldsymbol{e}_v) = \sum_{\boldsymbol{Y}} P(\boldsymbol{Y}, \boldsymbol{e}_v), \tag{4}$$

which allows us to reuse the result of Eq. 3, instead of having to marginalize out both $\boldsymbol{Y}$ and $\boldsymbol{W}$ from the joint distribution $P(\boldsymbol{X})$ of all variables in the BN. Note that $P(\boldsymbol{e}_v)$ may also simply be regarded as an elementary renormalization constant, which ensures that the posterior distribution sums to 1, i.e., $P(\boldsymbol{Y}|\boldsymbol{e}_v) \propto P(\boldsymbol{Y}, \boldsymbol{e}_v)$ [24, 27] (Papers C, and D).

To illustrate the procedure, we consider the example of Sec. 3.2. Again, the joint distribution that factorizes according Fig. 3 is written as

$$P(B, E, A, R, W) = P(B)P(E)P(A|B, E)P(R|E)P(W|A).$$

Now, say that we receive some evidence on the variables B and R, i.e., $\boldsymbol{e}_v = \{B = b, R = r\}$, and we want to query A, given this evidence. First, we reduce the relevant factors by the evidence, and marginalize out all non-query variables, i.e.,

$$P(A, \boldsymbol{E}_v = \boldsymbol{e}_v) = \sum_{E,W} P(B = b)P(E)P(A|B = b, E)P(R = r|E)P(W|A).$$

Second, we normalize by the evidence to produce the desired posterior distribution, i.e.,

$$P(A|\boldsymbol{E}_v = \boldsymbol{e}_v) = \frac{P(A, \boldsymbol{e}_v)}{P(\boldsymbol{e}_v)}, \quad \text{where} \quad P(\boldsymbol{e}_v) = \sum_{A} P(A, \boldsymbol{e}_v).$$

Equation 3 represents a brute force procedure for computing $P(\boldsymbol{Y}, \boldsymbol{E}_v = \boldsymbol{e}_v)$ called sum-product, where we first compute the product of factors in the summation and then marginalize out the variables $\boldsymbol{W}$ that are not of immediate interest, but there exists a variety of more efficient inference algorithms, both exact, like the sum-product algorithm, and approximate [24]. Some of the more common algorithms are summarized in Appendix A. Moreover, a thorough review on algorithms used for inference in BNs may be found in e.g., [29], and a recent review on general Bayesian inference is found in [30] (Papers C, and D).

MAXIMUM A-POSTERIORI INFERENCE

MAP inference finds the MAP assignment $\text{MAP}(\boldsymbol{Y}|\boldsymbol{E}_v = \boldsymbol{e}_v)$ of a subset $\boldsymbol{Y} = \boldsymbol{X} - \boldsymbol{E}_v$ of the variables in a BN, given a (possibly empty) evidence set $\boldsymbol{E}_v = \boldsymbol{e}_v$ on the other variables in the network, i.e.,

$$\text{MAP}(\boldsymbol{Y}|\boldsymbol{E}_v = \boldsymbol{e}_v) = \arg\max_{\boldsymbol{y}} P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{E}_v = \boldsymbol{e}_v). \tag{5}$$

That is, the MAP assignment is a single coherent assignment of highest posterior probability, which in general do not coincide with maximizing individual marginal probabilities. This assignment may not be unique, thus there may be several assignments that produce similar posterior probabilities.

If we again take basis in Eq. 2, we see that the denominator is constant with respect to $\boldsymbol{Y}$. Therefore, for the purpose of finding the MAP assignment, it is sufficient to consider the numerator in Eq. 5, i.e., $P(\boldsymbol{Y}|\boldsymbol{e}_v) \propto P(\boldsymbol{Y}, \boldsymbol{e}_v)$. The procedure represented by Eq. 5 is often referred to as max-product, because in this case we are maximizing a product of factors [24]. Again, there is a variety of different inference algorithms for finding the MAP assignment. For instance, the algorithms in Appendix A may easily be modified for MAP inference by replacing marginalizations with maximizations [24].

3.4 LEARNING DISCRETE BAYESIAN NETWORKS

Parts of this section appear in Papers A, C, and F.

As apparent from Eq. 1, a BN is fully specified by its DAG $\mathcal{G}$ and its parameters $\boldsymbol{\Theta}_{\mathcal{G}}$. The process of specifying the pair $\{\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}\}$ is termed learning, and it is usually performed in two steps: structure learning and parameter learning. Structure learning refers to the construction of the graph structure $\mathcal{G}$, and parameter learning refers to the specification of the model parameters $\boldsymbol{\Theta}_{\mathcal{G}}$.

Both learning tasks may be undertaken by use of a bottom-up or top-down approach, or by a combining hereof. In a top-down approach, the DAG and parameters are defined using information provided in a database, and in a bottom-up approach, domain experts are interviewed to identify the DAG and parameters [31]. As mentioned, BNs are defined in terms of conditional dependence relations and probabilistic properties, without any implication that edges should point from causes to effects in the DAG. However, it is argued by Pearl [26] that causal BNs pose a more reliable and natural way of expressing our knowledge about a given problem domain. That is, we should strive to use a combined learning approach whenever possible, as it makes the best use of the available and relevant knowledge and information about a given system (Papers A, and C).

We adapt the Bayesian approach to learning, which involves a consistent quantification of uncertainty using probabilities. After observing some data $\mathcal{D}$, the (current) prior distribution $P(\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}})$ may be updated using Bayes' theorem to obtain the posterior distribution, i.e.,

$$\underbrace{P(\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{D})}_{\text{posterior}} \propto \underbrace{P(\mathcal{D}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}})}_{\text{likelihood}} \underbrace{P(\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}})}_{\text{prior}}. \tag{6}$$

In this setting, the data set $\mathcal{D} = \{\boldsymbol{x}[n]\}_{n=1}^{N}$ is composed of N i.i.d. observations in M random vector. The change in distribution represented by Eq. 6

reflects the information gain we get by observing some data, and it further illustrates what it means for a machine to "learn from data". Moreover, the posterior distribution in turn becomes the prior distribution to be used with new observations, which makes the updating process inherently sequential, and therefore well suited for online learning, where the data points are considered one at a time and the probability distributions updated after observing each data point [6, 32] (Paper F).

The Bayesian approach is especially relevant when data are limited and the resulting uncertainty in the model itself and its model parameters is significant. In such cases, traditional optimization-based approaches are often prone to overfitting, which means that the model is tuned to noise in the data, leading to poor generalization for new data. In this regard, a relevant distinction is made by Bishop [32] between the computational size of a data set, which refers to its size in terms of bits, and the statistical size of a data set in relation to the model being considered. Thus, though we are in the era of "big data", statistically small data sets arise in many situations due to data fragmentation, also known as the curse of dimensionality.

In the remainder of this section, we consider learning of system representations for discrete-valued random variables, or accordingly dynamically discretized continuous-valued random variable. By casting the discretization process as part of the learning problem, we hereby strive to make as few assumptions as possible regarding the distribution family of the domain variables, when learning a BN representation of a given system (Papers C, and F).

PARAMETER LEARNING

In this section, we consider how to learn the parameters of a Bayesian network from data, when the corresponding DAG $\mathcal{G}$ is given, and our data set $\mathcal{D}$ consists of complete assignments to all variables. In this setting, with reference to Eq. 6, parameter learning is usually performed by searching for a set of parameters in $\boldsymbol{\Theta}_{\mathcal{G}}$ that maximizes

$$P(\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{G},\mathcal{D}) = \frac{P(\mathcal{G},\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{D})}{P(\mathcal{G})} \propto P(\mathcal{D}|\mathcal{G},\boldsymbol{\Theta}_{\mathcal{G}})P(\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{G}), \tag{7}$$

where $P(\mathcal{G},\boldsymbol{\Theta}_{\mathcal{G}}) = P(\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{G})P(\mathcal{G})$ (Paper F). First, we consider how the likelihood function decomposes

$$\begin{aligned} P(\mathcal{D}|\mathcal{G},\boldsymbol{\Theta}_{\mathcal{G}}) &= \prod_n P(\boldsymbol{x}[n]|\boldsymbol{\Theta}_{\mathcal{G}},\mathcal{G}) \\ &= \prod_n \prod_i P(x_i[n]|\mathbf{pa}_i[n],\boldsymbol{\Theta}_{\mathcal{G}},\mathcal{G}) \\ &= \prod_i \left[\prod_n P(x_i[n]|\mathbf{pa}_i[n],\boldsymbol{\Theta}_{\mathcal{G}},\mathcal{G}) \right] \end{aligned}$$

$$P(\mathcal{D}|\mathcal{G},\boldsymbol{\Theta}_{\mathcal{G}}) = \prod_i \left[\prod_n P(x_i[n]|\mathbf{pa}_i[n],\boldsymbol{\Theta}_{X_i|\mathbf{Pa}_i},\mathcal{G}) \right], \tag{8}$$

where $\boldsymbol{\Theta}_{X_i|\mathbf{Pa}_i}$ denotes the subset of parameters that defines $P(X_i|\mathbf{Pa}_i)$ in $\mathcal{G}$. Thus in Eq. 8, the terms in the square brackets define the conditional likelihood of a variable given its parents [24]. Second, if we assume that the parameter vectors $\boldsymbol{\Theta}_{X_i|\mathbf{Pa}_i}$ are independent a priori, which corresponds to an assumption of global parameter independence, the prior decomposes as

$$P(\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{G}) = \prod_i P(\boldsymbol{\Theta}_{X_i|\mathbf{Pa}_i}|\mathcal{G}). \tag{9}$$

Moreover, if we further assume that the parameter vectors $\boldsymbol{\Theta}_{X_i|\boldsymbol{u}_i}$ for each parent configuration $\boldsymbol{u}_i$ in $\boldsymbol{\Theta}_{X_i|\mathbf{Pa}_i}$ are independent a priori, which corresponds to an assumption of local parameter independence, the individual prior terms $P(\boldsymbol{\Theta}_{X_i|\mathbf{Pa}_i}|\mathcal{G})$ decompose as

$$P(\boldsymbol{\Theta}_{X_i|\mathbf{Pa}_i}|\mathcal{G}) = \prod_{\boldsymbol{u}_i \in \mathrm{Val}(\mathbf{Pa}_i)} P(\boldsymbol{\Theta}_{X_i|\boldsymbol{u}_i}|\mathcal{G}), \tag{10}$$

where $\mathrm{Val}(\mathbf{Pa}_i)$ are the parent configurations of $\mathbf{Pa}_i$. In Eq. 10, we assume $P(\boldsymbol{\Theta}_{X_i|\boldsymbol{u}_i}|\mathcal{G})$ to be the Bayesian Dirichlet equivalent uniform (BDeu) prior, i.e.,

$$P(\boldsymbol{\Theta}_{X_i|\boldsymbol{u}_i}|\mathcal{G}) = \mathrm{Dir}(\boldsymbol{\alpha}_{X_i|\boldsymbol{u}_i}) \quad \text{with} \quad \boldsymbol{\alpha}_{X_i|\boldsymbol{u}_i} = \left\{ \alpha_{x_i^j|\boldsymbol{u}_i} = \frac{\alpha}{|\boldsymbol{\Theta}_{X_i|\mathbf{Pa}_i}|} \right\}_{j=1}^{|X_i|}, \tag{11}$$

where $\alpha_{x_i|\boldsymbol{u}_i}^j$ is the prior weight in bin j of variable X_i, with parent configuration $\boldsymbol{u}_i$; and α is the so-called imaginary sample size associated with the BDeu prior, which specifies the weight assigned to the prior, compared to the weight assigned to the likelihood through the sample size of $\mathcal{D}$ [24, 31, 33]. Recommendations on the imaginary sample size α are given in [34].

Based on these assumptions, the parameter posterior also decomposes, with one posterior term $P(\boldsymbol{\Theta}_{X_i|\mathbf{Pa}_i}|\mathcal{G},\mathcal{D})$ per variable, i.e.,

$$\begin{aligned} P(\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{G},\mathcal{D}) &= \prod_i P(\boldsymbol{\Theta}_{X_i|\mathbf{Pa}_i}|\mathcal{G},\mathcal{D}) \\ &= \prod_i \prod_{\boldsymbol{u}_i \in \mathrm{Val}(\mathbf{Pa}_i)} \mathrm{Dir}\left(\left\{ \alpha_{x_i^j|\boldsymbol{u}_i} + N[x_i^j,\boldsymbol{u}_i] \right\}_{j=1}^{|X_i|} \right), \end{aligned} \tag{12}$$

where $N[x_i^j,\boldsymbol{u}_i]$ is the number of samples in bin j of variable X_i, with parent configuration $\boldsymbol{u}_i$ [24, 35] (Paper F).

As mentioned, parameter learning is usually performed by searching for a parameter vector that maximizes Eq. 12 to produce the maximum a posteriori (MAP) estimate of the parameter vector $\hat{\boldsymbol{\theta}}_{\mathcal{G}}$, which is then later used

for inference. The problem with this approach is that the statistical uncertainties, due to a limited amount of data, are disregarded. Instead, we may adapt a full Bayesian approach and average over all possible realizations of the parameter vector. In practice, this is typically pursued by sampling a number of realizations of the parameter vector $\{\boldsymbol{\theta}_{\mathcal{G}}^{(t)}\}_{t=1}^{T}$ using Eq. 12 and then performing inferences for each of these realizations.

PARAMETER LEARNING FROM INCOMPLETE DATA

In this section, we consider how to learn the parameters of a BN from data, when the corresponding DAG $\mathcal{G}$ is given, and our data set $\mathcal{D}$ is partially observed. In this case, we write $\mathcal{D} = \{\mathcal{D}_{obs}, \mathcal{D}_{hid}\}$, where $\mathcal{D}_{obs}$ denotes the observed data and $\mathcal{D}_{hid}$ denotes the hidden data. Moreover, we also define an inclusion indicator variable $\mathcal{I}$, which determines, whether the n'th instance of variable i is observed $\mathcal{I}[i,n] = 1$, or not $\mathcal{I}[i,n] = 0$. That is, we assume that the incomplete data set has been generated by a mechanism that hides some of the data values in a corresponding complete data set [28, 36]. The joint distribution of $\{\mathcal{D}, \mathcal{I}\}$, given parameters $\{\boldsymbol{\Theta}_{\mathcal{G}}, \boldsymbol{\Phi}\}$ and DAG $\mathcal{G}$, can be written as

$$P(\mathcal{D}, \mathcal{I}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}, \boldsymbol{\Phi}) = P(\mathcal{D}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}})P(\mathcal{I}|\mathcal{D}, \boldsymbol{\Phi}), \tag{13}$$

where the conditional distribution of $\mathcal{I}$ describes the missing-data mechanism. We can now obtain the probability distribution for the observed information $\{\mathcal{D}_{obs}, \mathcal{I}\}$ by summing $\mathcal{D}_{hid}$ out of the equation, i.e.,

$$P(\mathcal{D}_{obs}, \mathcal{I}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}, \boldsymbol{\Phi}) = \sum_{\mathcal{D}_{hid}} P(\mathcal{D}_{obs}, \mathcal{D}_{hid}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}})P(\mathcal{I}|\mathcal{D}_{obs}, \mathcal{D}_{hid}, \boldsymbol{\Phi}). \tag{14}$$

One of three assumptions for the missing-data mechanism is typically considered: (i) data missing completely at random (MCAR), where the mechanism is assumed to be independent of the data, i.e., $P(\mathcal{I}|\mathcal{D}_{obs}, \mathcal{D}_{hid}, \boldsymbol{\Phi}) = P(\mathcal{I}|\boldsymbol{\Phi})$; (ii) data missing at random (MAR), where we assume that the mechanism does not depend on the hidden data, i.e., $P(\mathcal{I}|\mathcal{D}_{obs}, \mathcal{D}_{hid}, \boldsymbol{\Phi}) = P(\mathcal{I}|\mathcal{D}_{obs}, \boldsymbol{\Phi})$; and (iii) data missing not at random (MNAR), where the mechanism depends on both the observed and the hidden data [36] (Paper F).

As an example, we consider a questionnaire survey. A MCAR situation may arise when some of the questionnaires are lost in the mail, thus the missingness does not depend on the characteristics of the individuals participating in the survey. A MAR situation may arise when men refuse to answer some questions in the questionnaire at rates significantly higher than women. Under the MAR assumption, we can correct for this kind of missingness, when we have observed the gender of the individuals. A MNAR situation may arise when people from certain social groups, or people from a specific large city, do not answer. In this case, the missing-data mechanism is non-ignorable, because it is needed to identify the non-responders [37].

In this study, we assume data to be MAR, thus we assume the observed data to be informative of the missing data. Under this assumption, Eq. 14 may now be written as

$$\begin{aligned} P(\mathcal{D}_{obs}, \mathcal{I}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}, \boldsymbol{\Phi}) =& P(\mathcal{I}|\mathcal{D}_{obs}, \boldsymbol{\Phi}) \sum_{\mathcal{D}_{hid}} P(\mathcal{D}_{obs}, \mathcal{D}_{hid}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}) \\ =& P(\mathcal{I}|\mathcal{D}_{obs}, \boldsymbol{\Phi}) P(\mathcal{D}_{obs}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}). \end{aligned} \tag{15}$$

We see that the first term depends only on the parameters $\boldsymbol{\Phi}$, and the second term depends only on the parameters $\boldsymbol{\Theta}_{\mathcal{G}}$. If we further assume the two parameter vectors to be independent in the prior distribution, we can optimize the posterior distribution of the parameters $\boldsymbol{\Theta}_{\mathcal{G}}$ in $P(\boldsymbol{X})$ independently of the parameters of the missing-data mechanism. That is, while learning the parameters $\boldsymbol{\Theta}_{\mathcal{G}}$, we can ignore the missing-data mechanism [24, 36].

We are thus back in the setting of Eq. 7, where the corresponding likelihood function may now be written as

$$P(\mathcal{D}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}) = \prod_n P(\boldsymbol{o}[n]|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}) = \prod_n \sum_{\boldsymbol{h}[n]} P(\boldsymbol{o}[n], \boldsymbol{h}[n]|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}), \tag{16}$$

with $\boldsymbol{o}[n]$ being the observed values in instance n, and $\boldsymbol{h}[n]$ being the unknown values of hidden variables in instance n. In order to evaluate the likelihood, we need to perform inference for the hidden variables of each instance. Note that this formulation implies that we lose the property of parameter independence and thereby the decomposability of the likelihood function. This is easiest seen from the simple meta-networks in Fig. 5. If both $x[n]$ and $y[n]$ are observed (Fig. 5a), the path $\boldsymbol{\Theta}_X \rightarrow x[n] \rightarrow y[n] \leftarrow \boldsymbol{\Theta}_Y$ is blocked, but if $x[n]$ is missing and $y[n]$ is observed (Fig. 5b), the path is active and information can flow. The former case corresponds to learning from complete data, and the latter case corresponds to learning from incomplete data [4, 24].

Again, we assume the BDeu prior (Eq. 11), which as mentioned satisfies both global and local parameter independence, but because the poste-

Figure 5: Meta-networks for parameter estimation: (a) fully observed training data, and (b) partially observed training data.

rior is a product of the likelihood and the prior, it follows that the posterior does not decompose and no closed form solution exists. One way of addressing this problem is to learn the parameter setting that maximizes the posterior probability using e.g., a generic gradient-based optimization algorithm or the expectation maximization (EM) algorithm. Another way of addressing the problem is to use a sampling-based method, like Gibbs sampling, to approximate the posterior distribution [24, 38]. A recent review on common algorithms used for parameter learning in BNs may be found in e.g., [29, 39, 40] (Paper F).

Expectation maximization Instead of integrating over the entire posterior $P(\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{G},\mathcal{D})$, the EM algorithm searches for a parameter setting that maximizes the following expression

$$\hat{\boldsymbol{\theta}}_{\mathcal{G}} = \arg\max_{\boldsymbol{\theta}_{\mathcal{G}}} P(\boldsymbol{\theta}_{\mathcal{G}}|\mathcal{G},\mathcal{D}) = \arg\max_{\boldsymbol{\theta}_{\mathcal{G}}} \frac{P(\mathcal{D}|\mathcal{G},\boldsymbol{\theta}_{\mathcal{G}})P(\boldsymbol{\theta}_{\mathcal{G}}|\mathcal{G})}{P(\mathcal{D})}, \tag{17}$$

thus it produces a MAP parameter estimate. This search is conducted by successively applying two steps until convergence: Expectation (E-step), the algorithm uses the current parameter setting $\hat{\boldsymbol{\theta}}_{\mathcal{G}}^{(t)}$ to compute the expected sufficient statistics $\bar{N}[\cdot,\cdot]$ related to Eq. 12, i.e., the expected sufficient statistics are the expected counts of the different events $\{x_i^j, \boldsymbol{u}_i\}$ in the data. Maximization (M-step), the expected sufficient statistics are used in Eq. 12, and the expression is optimized to produce a new MAP estimate $\hat{\boldsymbol{\theta}}_{\mathcal{G}}^{(t+1)}$. The expected sufficient statistics in the E-step are calculated as

$$\bar{N}[x_i^j, \boldsymbol{u}_i] = \sum_{n=1}^{N} P(x_i^j, \boldsymbol{u}_i|\boldsymbol{o}[n], \mathcal{G}, \hat{\boldsymbol{\theta}}_{\mathcal{G}}^{(t)}). \tag{18}$$

That is, in order to calculate the sufficient statistics, we need to perform inference for every instance in the data set over the current BN specified by $\{\mathcal{G}, \hat{\boldsymbol{\theta}}_{\mathcal{G}}^{(t)}\}$ [4, 24, 41]. Pseudo-code for the EM algorithm is provided in Alg. 1.

Note that if we consider the expected sufficient statistics from the last fixed point update of the EM algorithm, in combination with Eq. 12, the resulting EM distribution appears as a special case of a variational distribution of the posterior [6, 24, 36]. See e.g., Appendix A for further details on variational inference in a BN setting. Moreover, we discuss the EM algorithm in more details in Sec. 6.3 in the context of Gaussian mixture models.

Gibbs sampling A Gibbs sampler is a Markov chain Monte Carlo (MCMC) method, in which we construct a Markov chain whose states are one assignment to the unobserved variables in our model; such that the stationary distribution of the chain corresponds to the posterior distribution over the

Algorithm 1: Pseudo-code of the EM algorithm for BNs.

Input: $\mathcal{D}$, $\mathcal{G}$, α
Output: $\hat{\boldsymbol{\theta}}_{\mathcal{G}}^{(t)}$

Initialization: $\hat{\boldsymbol{\theta}}_{\mathcal{G}}^{(0)}$
for *t = 0, 1, ..., until convergence* **do**
  E-step:
    Initialization: $\{\bar{N}[x_i^j, \boldsymbol{u}_i]\} \leftarrow 0$
  for $n = 1, ..., N$ **do**
    for $i = 1, ..., M$ **do**
      for *each* $x_i^j, \boldsymbol{u}_i \in Val(X_i, \mathbf{Pa}_i)$ **do**
        $\bar{N}[x_i^j, \boldsymbol{u}_i] \leftarrow \bar{N}[x_i^j, \boldsymbol{u}_i] + P(x_i^j, \boldsymbol{u}_i | \boldsymbol{o}[n], \mathcal{G}, \hat{\boldsymbol{\theta}}_{\mathcal{G}}^{(t)})$
      end
    end
  end
  M-step:
  for $i = 1, ..., M$ **do**
    for *each* $x_i^j, \boldsymbol{u}_i \in Val(X_i, \mathbf{Pa}_i)$ **do**
      $\alpha_{x_i^j|\boldsymbol{u}_i} \leftarrow \frac{\alpha}{|\boldsymbol{\Theta}_{X_i|\mathbf{Pa}_i}|}$
      $\hat{\boldsymbol{\theta}}_{x_i^j|\boldsymbol{u}_i}^{(t+1)} \leftarrow \frac{\alpha_{x_i^j|\boldsymbol{u}_i} + \bar{N}[x_i^j, \boldsymbol{u}_i]}{\alpha_{\boldsymbol{u}_i} + \bar{N}[\boldsymbol{u}_i]}$
    end
  end
end

unobserved variables. In our case, a state of the Markov chain consists of $\boldsymbol{Z} = \{\boldsymbol{\Theta}_{\mathcal{G}}, \mathcal{D}_{hid}\}$, and two assumptions are made: (i) the sampler is irreducible, i.e., $P(Z = z) > 0$ for all unobserved variable states, and (ii) each $Z \in \boldsymbol{Z}$ is chosen infinitely often, which is typically pursued in practice by cycling deterministically through the unobserved variables in $\boldsymbol{Z}$ until a steady state condition is reached, called the mixing time [23, 24].

A generic Gibbs sampler proceeds as follows: We initiate the chain at an arbitrary point in $\boldsymbol{Z}$-space $\boldsymbol{z} = \{z_k\}_{k=1}^K$, and produce a new state by cycling through the elements of $\boldsymbol{Z}$. At each iteration t, an ordering of the K elements of $\boldsymbol{Z}$ is chosen and, in turn, each $Z_k^{(t)}$ is sampled from the conditional distribution given all the other components of $\boldsymbol{Z}$, i.e.,

$$P(Z_k^{(t)} | \boldsymbol{z}_{\sim k}^{(t)}), \tag{19}$$

where $\boldsymbol{z}_{\sim k}^{(t)}$ represents an instantiation of all the components of $\boldsymbol{Z}$ but Z_k, at

Algorithm 2: Pseudo-code of Gibbs sampling in BNs.

Input: $\mathcal{D}$, $\mathcal{G}$, $\boldsymbol{Z}$, T
Output: $\{z^{(t)}\}_{t=0}^{T}$
Initialization: $z^{(0)} \leftarrow \{\boldsymbol{\theta}_{\mathcal{G}}^{(0)}, \{\boldsymbol{h}[n]^{(0)}\}\}$
for $t = 1, ..., T$ **do**
  $z^{(t)} \leftarrow z^{(t-1)}$
  for *each* $Z_k \in \boldsymbol{Z}$ **do**
    Sample $z_k^{(t)}$ from $P(Z_k^{(t)} | \boldsymbol{z}_{\sim k}^{(t)})$
  end
end

their current values in iteration t, i.e.,

$$\boldsymbol{z}_{\sim k}^{(t)} = \{z_1^{(t)}, ..., z_{k-1}^{(t)}, z_{k+1}^{(t-1)}, ..., z_K^{(t-1)}\}. \tag{20}$$

Pseudo-code for a generic Gibbs sampler is provided in Alg. 2.

In our sample space $\boldsymbol{Z}$, we have two distinct classes of variables: those of $\boldsymbol{\Theta}_{\mathcal{G}}$ and those of $\mathcal{D}_{hid}$. If we consider a variable $Z_k \in \mathcal{D}_{hid}$, at each step, we know all the current parameters and all the other variables in $\mathcal{D}_{hid}$, so we can infer the distribution in Eq. 19 from the BN representation as $P(X_i = Z_k | \boldsymbol{E}_v = \boldsymbol{e}_v)$, where $\boldsymbol{e}_v = \{\boldsymbol{o}[n], \boldsymbol{z}_{\sim k}^{(t)}[n]\}$ corresponds to the evidence in instance n, to which Z_k belongs. Now, if we consider a variable $Z_k \in \boldsymbol{\Theta}_{\mathcal{G}}$, at each step, all variables in $\mathcal{D}_{hid}$ are instantiated, and thus we have a complete data set. This means that we can instantiate the parameters as we would do in a complete data setting, see Eq. 12. The natural order of inference in each iteration is therefore to impute all the missing values in $\mathcal{D}_{hid}$ first, and then draw the parameter values from the distribution of $\boldsymbol{\Theta}_{\mathcal{G}}$ [24, 36, 41].

STRUCTURE LEARNING

We now consider how to learn the DAG of a BN from complete data. Taking basis in Eq. 6, structure learning is usually performed by searching for a DAG $\mathcal{G}$ that maximizes

$$P(\mathcal{G}|\mathcal{D}) \propto P(\mathcal{G})P(\mathcal{D}|\mathcal{G}) = P(\mathcal{G}) \int P(\mathcal{D}|\mathcal{G}, \boldsymbol{\theta}_{\mathcal{G}})P(\boldsymbol{\theta}_{\mathcal{G}}|\mathcal{G})d\boldsymbol{\theta}_{\mathcal{G}}, \tag{21}$$

where $P(\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}) = P(\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{G})P(\mathcal{G})$, and the prior over structures $P(\mathcal{G})$ is usually assumed to be uniform. As is apparent from Eq. 21, it is not possible to perform this computation without also considering the parameters $\boldsymbol{\Theta}_{\mathcal{G}}$ of the BN model. Therefore, to make $P(\mathcal{G}|\mathcal{D})$ independent of any specific choice of $\boldsymbol{\Theta}_{\mathcal{G}}$, we need to integrate $\boldsymbol{\Theta}_{\mathcal{G}}$ out of the equation [31, 35] (Paper F).

If we assume that the parameter prior $P(\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{G})$ satisfies global parameter independence, see Eq. 9, and we define $P(\mathcal{D}|\mathcal{G},\boldsymbol{\Theta}_{\mathcal{G}})$ according to Eq. 8, then $P(\mathcal{D}|\mathcal{G})$ may be written as

$$P(\mathcal{D}|\mathcal{G}) = \prod_i \int \prod_n P(x_i[n]|\mathbf{pa}_i[n],\boldsymbol{\theta}_{X_i|\mathbf{Pa}_i},\mathcal{G}) P(\boldsymbol{\theta}_{X_i|\mathbf{Pa}_i}|\mathcal{G}) d\boldsymbol{\theta}_{X_i|\mathbf{Pa}_i}. \tag{22}$$

Moreover, if $P(\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{G})$ also satisfies local parameter independence, see Eq. 10, then $P(\mathcal{D}|\mathcal{G})$ decomposes as

$$P(\mathcal{D}|\mathcal{G}) = \prod_i \prod_{\boldsymbol{u}_i \in \mathrm{Val}(\mathbf{Pa}_i)} \int \prod_{n,\boldsymbol{u}_i[n]=\boldsymbol{u}_i} P(x_i[n]|\boldsymbol{u}_i,\boldsymbol{\theta}_{X_i|\mathbf{Pa}_i},\mathcal{G}) P(\boldsymbol{\theta}_{X_i|\boldsymbol{u}_i}|\mathcal{G}) d\boldsymbol{\theta}_{X_i|\boldsymbol{u}_i}. \tag{23}$$

For discrete BNs, we can estimate $P(\mathcal{D}|\mathcal{G})$ in closed form. In this estimation, we again assume a BDeu prior over the parameter space of each parent configuration $\boldsymbol{u}_i$, see Eq. 11. Under these assumptions, $P(\mathcal{D}|\mathcal{G})$ may be evaluated as

$$P(\mathcal{D}|\mathcal{G}) = \prod_i \prod_{\boldsymbol{u}_i \in \mathrm{Val}(\mathbf{Pa}_i)} \frac{\Gamma(\alpha_{X_i|\boldsymbol{u}_i})}{\Gamma(\alpha_{X_i|\boldsymbol{u}_i} + N[\boldsymbol{u}_i])} \prod_{x_i^j \in \mathrm{Val}(X_i)} \frac{\Gamma(\alpha_{x_i^j|\boldsymbol{u}_i} + N[x_i^j,\boldsymbol{u}_i])}{\Gamma(\alpha_{x_i^j|\boldsymbol{u}_i})}, \tag{24}$$

where $\Gamma(\cdot)$ is the Gamma function; $N[\boldsymbol{u}_i]$ is the number of samples with configuration $\boldsymbol{u}_i$; $N[x_i^j,\boldsymbol{u}_i])$ is the number of samples in bin j of variable X_i, with parent configuration $\boldsymbol{u}_i$; and $\alpha_{\cdot|\boldsymbol{u}_i}$ are the imaginary samples of the BDeu prior [24]. Recommendations on the imaginary sample size α of the BDeu prior are given in [34]. At the large sample limit, Eq. 24 is equivalent to what is known as the Bayesian information criterion (BIC), which negation is equivalent to an information theoretic score called minimum description length (MDL) [23, 33, 42].

Finding a DAG that maximizes Eq. 21 is generally an intractable problem [43]. One approach to deal with this problem is to resort to a heuristic search strategy to find a high-scoring DAG. Another approach is to avoid the need to define a measure of goodness-of-fit for $\mathcal{G}$, and instead use conditional independence tests on $\mathcal{D}$ to learn a DAG one edge at a time. The former approach is called score-based structure learning, and the latter approach is called constraint-based structure learning. Additionally, one may combine the two approaches by first reducing the search space of DAGs using a constraint-based structure learning algorithm, and then search for a high-scoring DAG in the reduced search space using a score-based structure learning algorithm. This approach is often referred to as hybrid structure learning [31] (Paper F).

Recent advances on structure learning with respect to decomposable scores, e.g., Eq. 21, explore techniques such as dynamic programming,

e.g., [44], and integer linear programming, e.g., [45, 46], to perform an exact search on the space of DAGs, but as the computational and memory requirements are exponential in the number of vertices in the problem domain, these approaches are only feasible for smaller domains. For a recent review on structure learning in graphical models see e.g., [29, 47, 48].

Based only on data, structure learning algorithms identify an equivalence class, either by finding a DAG within the equivalence class or by finding the essential graph.[5] Furthermore, under the assumption that there is a DAG $\mathcal{G}^*$ faithful to the true underlying probability distribution $P^*(\boldsymbol{X})$, which has generated the data $\mathcal{D}$, both score-based and constraint-based algorithms will return an optimally scoring graph structure in the large sample limit.[6] When the faithfulness assumption is not fulfilled, a score-based algorithm still returns a suitable graph structure, given that the weaker composition condition holds [47, 49].[7]

In this study, we apply a set of hill-climbing strategies to perform score-based structure learning. A greedy hill-climbing strategy proceeds as follows: In each iteration, we define the neighborhood of the current DAG $\mathcal{G}^{(t)}$ as all DAGs we can produce from $\mathcal{G}^{(t)}$ by adding an edge, removing an edge, or reversing an edge. In this neighborhood, we pick the DAG that has the highest score and update $\mathcal{G}^{(t+1)}$. This strategy only guaranties to find a local optimum, but we may improve our chances of finding a "good" optimum by including a tabu list of previously visited structures and/or performing random restarts, when a local optimum is reached [31, 50]. Score-based algorithms and tabu search in particular have been proven to perform well in practice, in terms of accuracy and speed of network reconstruction, for both small and large sample sizes [50] (Paper F).

In this section, we have so far only dealt with the problem of finding a high-scoring DAG corresponding to a local optimum, but there may be an ensemble of models, which correspond to different local optima that explain the data equally well or even better than this model. This leads us to a discussion of how we can address model uncertainty in the setting of structure learning. One approach is to run the structure learning algorithm several times with different bootstrap data sets [51] and/or different initial DAGs, generated at random or by use of a MCMC scheme [52–54]. Another approach is to use a MCMC scheme to simulate over the space of DAGs [55] or possible orderings of the vertices within the DAG [56].

[5] Two DAGs that have the same d-separation properties are said to be Markov equivalent. The essential graph of a Markov equivalence class is a graph that has both directed and undirected edges (known as a p-DAG), where the directed edges are those that retain the same direction for all DAGs within the Markov equivalence class, and the undirected are those that change direction [47].

[6] A probability distribution P is faithful to a DAG $\mathcal{G}$, if any independence in P is reflected in the d-separation properties of $\mathcal{G}$ [24].

[7] The composition condition stats that if $X \perp Y|Z$ and $X \perp W|Z$ then $X \perp Y \cup W|Z$ [47].

STRUCTURE LEARNING AND AUTOMATED DISCRETIZATION

In this section, we consider how to learn the DAG of a BN and, at the same time, the optimal discretization of continuous variables from complete data. When discretizing numerical data, it is important to keep the information loss at a minimum by establishing a trade-off between model accuracy and model complexity, in terms of the number of discretization intervals for each variable, as we are generally likely to find more dependencies in the data, when the data are represented by only a few intervals (coarse discretization), than when the data are represented by many intervals (fine discretization) [57, 58]. Under these considerations, we discretize the continuous variables by use of a multivariate discretization procedure, embedded in the structure learning procedure, which accounts for the interactions in the current graph structure.

The method we use is based on [59], thus we assume that a continuous data set is generated in two steps. First, an interval of a variable is selected from the distribution of the discrete variable. Second, the corresponding continuous value is drawn from a distribution over the discrete interval. We then seek an optimal discrete representation $\mathcal{D}$ of the original continuous data set $\mathcal{D}^c$ (Papers A, and C). The discretization policy $\Lambda_\mathcal{G}$ depends on $\mathcal{G}$ and the observed continuous data $\mathcal{D}^c$, thus we need to specify our beliefs about the triple $\{\mathcal{G}, \boldsymbol{\Theta}_\mathcal{G}, \Lambda_\mathcal{G}\}$. Analogue to Eq. 6, our joint posterior distribution for this problem takes the form

$$P(\mathcal{G}, \boldsymbol{\Theta}_\mathcal{G}, \Lambda_\mathcal{G}|\mathcal{D}^c) \propto P(\mathcal{D}^c|\mathcal{G}, \boldsymbol{\Theta}_\mathcal{G}, \Lambda_\mathcal{G})P(\mathcal{G}, \boldsymbol{\Theta}_\mathcal{G}, \Lambda_\mathcal{G}), \tag{25}$$

where $\Lambda_\mathcal{G}$ specifies a set of interval boundary points for each variable. Furthermore, as implicitly implied by the generative process for $\mathcal{D}^c$, we assume that, given $\mathcal{D}$ and $\Lambda_\mathcal{G}$, $\mathcal{D}^c$ is conditionally independent of $\{\mathcal{G}, \boldsymbol{\Theta}_\mathcal{G}\}$, whereby the likelihood of $\mathcal{D}^c$ may be written as

$$P(\mathcal{D}^c|\mathcal{G}, \boldsymbol{\Theta}_\mathcal{G}, \Lambda_\mathcal{G}) = P(\mathcal{D}^c|\mathcal{D}, \Lambda_\mathcal{G})P(\mathcal{D}|\mathcal{G}, \boldsymbol{\Theta}_\mathcal{G}, \Lambda_\mathcal{G}). \tag{26}$$

This assumption is illustrated in Fig. 6. Based on Eq. 26, we may now rewrite Eq. 25 as

$$P(\mathcal{G}, \boldsymbol{\Theta}_\mathcal{G}, \Lambda_\mathcal{G}|\mathcal{D}^c) \propto \underbrace{P(\mathcal{D}^c|\mathcal{D}, \Lambda_\mathcal{G})}_{\text{likelihood (continuous)}} \underbrace{P(\mathcal{D}|\mathcal{G}, \boldsymbol{\Theta}_\mathcal{G}, \Lambda_\mathcal{G})}_{\text{likelihood (discrete)}} \underbrace{P(\mathcal{G}, \boldsymbol{\Theta}_\mathcal{G}, \Lambda_\mathcal{G})}_{\text{prior}}. \tag{27}$$

The prior term in Eq. 27 factorizes as

$$P(\mathcal{G}, \boldsymbol{\Theta}_\mathcal{G}, \Lambda_\mathcal{G}) = P(\boldsymbol{\Theta}_\mathcal{G}|\mathcal{G}, \Lambda_\mathcal{G})P(\Lambda_\mathcal{G}|\mathcal{G})P(\mathcal{G}), \tag{28}$$

where we assume that $P(\Lambda_\mathcal{G}|\mathcal{G})$ and $P(\mathcal{G})$ are uniformly distributed over the space of discretization policies and DAGs, respectively; and $P(\boldsymbol{\Theta}_\mathcal{G}|\mathcal{G}, \Lambda_\mathcal{G})$ follow a product Dirichlet distribution defined by Eqs. 9–11. Under these

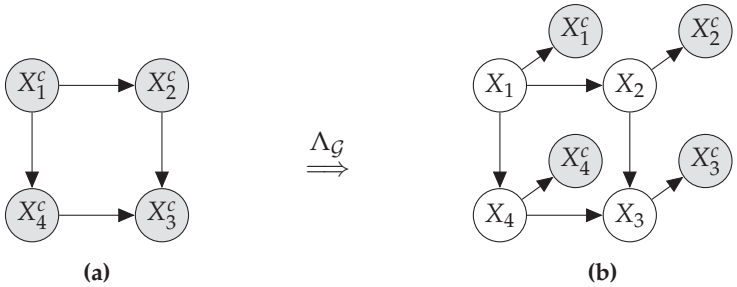


Figure 6: Structure learning and automated discretization: (a) the dependency structure is captured by interactions of the continuous variables; (b) the dependency structure is captured by interactions of the underlying discrete variables, and the continuous variables depend only on their discrete counterpart.

assumptions, the product of the two last terms in Eq. 27 (the discrete part) corresponds to the evaluation of Eq. 21 (Eq. 24) [60] (Paper F).

Two formulations of the continuous likelihood in Eq. 27 are considered in this study, i.e.,

$$P(\mathcal{D}^c|\mathcal{D},\Lambda_{\mathcal{G}}) = \prod_i \prod_{x_i^j \in \text{Val}(X_i)} \left(\frac{1}{\overline{\lambda}_i^j - \underline{\lambda}_i^j} \right)^{N[x_i^j]} \tag{29a}$$

$$P(\mathcal{D}^c|\mathcal{D},\Lambda_{\mathcal{G}}) = \prod_i \prod_{x_i^j \in \text{Val}(X_i)} \left(\frac{1}{N[x_i^j]} \right)^{N[x_i^j]}, \tag{29b}$$

where $N[x_i^j]$ is the number of samples in bin j of variable X_i, regardless of the parent configuration; and $\{\underline{\lambda}_i^j, \overline{\lambda}_i^j\}$ are the boundary points in bin j of variable X_i. Moreover, we only consider the $N-1$ midpoints of each data vector $\mathcal{D}_i^c$ as candidate boundary points for X_i in our implementations.

The formulation in Eq. 29a, which was proposed in the original work by Monti and Cooper [59], may not be normalizable for the boundary intervals corresponding to $j = 1$ and $j = |X_i|$. In these cases, we solve this issue by using the smallest value of the observation vector $\{x_i^c[n]\}_{n=1}^N$ as the lower bound, and the largest value of this vector as the upper bound. Another way to overcome the normalization problem of Eq. 29a is to use the formulation in Eq. 29b, which was proposed by Vogel [60]. This metric has the property that each observation of X_i^c corresponds to one unit of the metric.

Based on these assumptions, the scoring function of Eq. 27 establishes a trade-off between model accuracy and model complexity. On one hand,

the formulations of $P(\mathcal{D}^c|\mathcal{D}, \Lambda_\mathcal{G})$ in Eq. 29 rewards model complexity and prediction accuracy, with respect to the continuous variable, whereby it increases with an increasing number of intervals. On the other hand, the discrete part of Eq. 27, penalizes model complexity, whereby it balances the resolution of the discretization by decreasing as the number of intervals increases [59] (Paper F).

As the graph structure changes throughout the structure learning phase, the discretization is adjusted dynamically to maximize the score function Eq. 27 in a manner similar to that proposed in [61]. That is; first, the data are discretized based on the current graph structure and second, this discretization is used to learn a new graph structure. These two steps are repeated until the score function converges to a local optimum. A similar scheme for combined structure learning and discretization is used in [60] (Papers A, and C).

An important point regarding the discretization step above is that we only change the discretization policy $\Lambda_{\mathcal{G},i}$ for one variable at a time, while treating all other variables as being discrete and fixed. That is, at each iteration, we pick the variables one by one, and optimize the local scoring metric for the current variable, considering only the variable itself and the variables in its Markov blanket in Eq. 27.[8] We repeat this procedure until we cannot improve the local score for any of the continuous variables in the domain, and proceed to learn a new graph structure related to this discretization polity $\Lambda_\mathcal{G}^{(t+1)}$.

STRUCTURE LEARNING AND AUTOMATED DISCRETIZATION FROM INCOMPLETE DATA

Learning the DAG of a BN (in addition to the parameters) from an incomplete data set is challenging from both a statistical and a computational point of view. The score metrics we defined in the two previous sections, i.e., Eq. 21 and Eq. 27, are functions of the sufficient statistics $N[\cdot]$, through the definition of the (discrete) data likelihood in Eq. 24, and these are not defined for incomplete data sets. To circumvent this problem, we may adopt the definition of the (missing) data likelihood from Eq. 16, and thereby operate with a notion of expected sufficient statistics [40, 60].

As for the case of parameter learning given a graph structure, this may be accomplished by utilizing a deterministic optimization algorithm like EM, or a stochastic procedure like Gibbs sampling. The former approach is termed structural EM [62, 63], and it proceeds by embedding the structural search inside the EM procedure. The latter approach is usually termed data augmentation [64], and it utilizes sampling procedures to produce several completions of the training data set, which may then be used for structure learning in the

[8] In a Bayesian network, the Markov blanket of node X includes its parents, children, and the other parents of all of its children [25].

same manner as in a complete data setting. See e.g., [40] for a recent review on learning from incomplete data.

By use of one of the approaches mentioned above, the (discrete) data likelihood may now be evaluated, and learning of the triple $\{\mathcal{G}, \mathbf{\Theta}_{\mathcal{G}}, \Lambda_{\mathcal{G}}\}$ is again performed in accordance with Eq. 27. Note that in order to evaluate the (continuous) data likelihood in Eq. 26, the corresponding continuous imputations are needed. These imputations are sampled from the generative model implied by Eq. 29 (Paper F).

In this study, we employ a variant of the structural EM algorithm [37, 65] in order to learn a DAG $\mathcal{G}$ from incomplete data in correspondence with Eq. 27. In the E-step, we impute the missing data based on their maximum a-posteriori estimates using the current BN, which is termed hard-assignment EM [24]. In the M-step, we learn a new BN based on the imputed data set. This approach is computationally less demanding than ordinary EM, which is appreciated given that we already need to run several iterations of BN learning and discretization, where each iteration requires estimation of the missing values [60].

3.5 TEMPLATE MODELING

In this section, we consider an important extension to the language of BNs discussed so far, namely template models, which allow us to define a template representation that can be reused to solve multiple problem, either in sequence or simultaneously over multiple objects in the problem domain. The trick here is to increase the statistical power of the analysis by introducing parameters sharing between and/or within the groups. First, we discuss how to represent the distribution over multiple objects that are somehow related to each other, such as pixels in an image. Second, we discuss how we can represent distributions over systems that change over time, i.e., in sequence. Third and last, we discuss how to encode higher level dependencies in our models. In a template representation, random variables that are instantiated (duplicated) multiple times within the model are referred to as template variables, and the joint distribution over a set of instantiated template variables, called ground variables, is referred to a template factor [24].

PLATE MODELS

Plate models is a special case of a general class of object-relational models, which is commonly use in practice, notably for encoding the assumptions made in various learning tasks. The basic objects in these models are plates, which share a common set of variables and a common probabilistic model. Figure 7 shows the simplest possible plate model of multiple random variables $\{X_m\}$ generated from the same distribution; exemplified by multiple

tosses of a single coin. In this case, we have augmented the plate model by the parameter $\Theta \in [0, 1]$ to form a meta-network for the generative process, whereby the assumption that the variables have an identical distribution is made explicit. The expressive power of plate models goes far beyond the simple example considered here; the key idea is that plates can be combined and/or nested in any way to form richly structured distributions, which are able to utilize evidence that may otherwise be ignored [24]. Some examples are provided throughout the text, see e.g., Alg. 3.

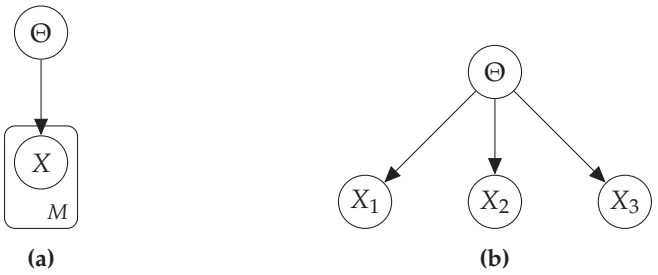


Figure 7: Plate model for multiple tosses of a single coin: (a) Compact representation, and (b) ground or unrolled Bayesian network for $M = 3$.

The basic plate concept as presented here may be extended in various ways. In the following, we will consider two such extensions, namely dynamic Bayesian networks (DBNs) for temporal modeling, and hierarchical Bayesian models for encoding multilevel dependencies.

DYNAMIC BAYESIAN NETWORKS

In this section, we define a template model representation for temporal systems, which is not possible with the basis plate representation, as a variable is not allowed to be a parent of itself at a future time step; a property we will need to encode temporal dynamics.

We define a ground random variable $X_i^{(t)}$ to be the instantiation of template variable X_i at time t, and $\boldsymbol{X}^{(t:t')}$ $(t < t')$ denote the set of variables $\{\boldsymbol{X}^{(t)} | t \in [t, t']\}$. Our goal is now to represent the joint distribution over a trajectory of assignments to each $\boldsymbol{X}^{(t)}$ for each relevant time instance t. In this regard, we will assume time to be discrete with granularity Δt, and thus system states are only observed in time slices. Moreover, we will assume the system to be Markovian, i.e., it satisfies the Markov assumption $(\boldsymbol{X}^{(t+1)} \perp \boldsymbol{X}^{(0:t-1)} | \boldsymbol{X}^{(t)})$, and stationary, also called time invariant or homogeneous, so that the transition model $P(\boldsymbol{X}^{(t+1)} | \boldsymbol{X}^{(t)})$ is the same for all time steps. Under these conditions, the distribution over trajectories sampled at

times $t = 0, 1, ..., T$ simplifies to

$$P(\boldsymbol{X}^{(0:T)}) = P(\boldsymbol{X}^{(0)}) \prod_{t=0}^{T-1} P(\boldsymbol{X}^{(t+1)}|\boldsymbol{X}^{(0:t)}) \tag{30a}$$

$$= P(\boldsymbol{X}^{(0)}) \prod_{t=0}^{T-1} P(\boldsymbol{X}^{(t+1)}|\boldsymbol{X}^{(t)}), \tag{30b}$$

where Eq. 30a encodes the joint distribution using the chain rule of probability, and Eq. 30b simplifies Eq. 30a by use of the Markov assumption. In most cases, the Markov assumption is reasonable, given that we consider a sufficiently rich state space description, e.g., for object localization, we could include both location and velocity in the state space description. Alternatively, we may employ a model that is semi-Markov, in which case the conditional independence assumptions of the model are relaxed [24].

Equation 30b allows us to define infinite trajectories using only an initial state distribution and a transition model, where the transition model is a conditional probability distribution, which we can define using a conditional BN. An example is shown in Fig. 8. Figure 8a shows a compact representation of the conditional model, called a 2-time-slice Bayesian network (2TBN), where we see the full representation at time $t + 1$, and only the interface variables at time t, thus the interface variables $\boldsymbol{X}_I \in \boldsymbol{X}$ are those variables at time t that have a direct influence on variables at time $t + 1$. Figure 8b shows the corresponding ground BN over three time slices. In general, the 2TBN representation therefore defines the conditional distribution, i.e.,

$$P(\boldsymbol{X}'|\boldsymbol{X}) = P(\boldsymbol{X}'|\boldsymbol{X}_I) = \prod_{i=1}^{M} P(X_i'|\mathbf{Pa}_i'), \tag{31}$$

where the template factors $P(X_i'|\mathbf{Pa}_i')$ are instantiated within each time slice at run time. Note that the initial state distribution does not need to encode the exact same dependencies as the transition model, as it appears to do in Fig. 8b [24].

In the 2TBN representation, edges connecting variables from one time slice to variables in the next are referred to as inter-time-slice edges, whereas edges connecting variables within a time slice are referred to as intra-time-slice edges. Moreover, inter-time-slice edges of the form $X \rightarrow X'$, thus edges connection a variable X at time t to its $t + 1$ representation X', are called persistence edges, and variables for which we have persistence edges are called persistence variables.

Many important special cases of DBNs are used extensively in practice. For example, variants of the hidden Markov model (HMM) are commonly used to represent the evolution of a discrete-state, latent Markov chains, and variants of linear dynamical systems, also referred to as Kalman filters or state-space models, are commonly use to represent the evolution of

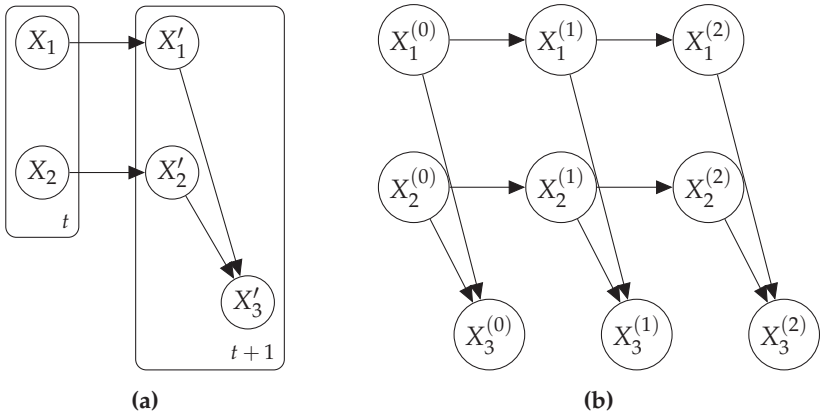


Figure 8: Dynamic Bayesian network: (a) 2-time-slice Bayesian network (2TBN) representation, and (b) ground or unrolled Bayesian network over three time slices.

a continuous-state, latent Markov chain. In fact, the model represented in Fig. 8 may e.g., be considered as an example of a factorial HMM, where X_3 is regarded as an observation variable. Note that tailored algorithms for efficient inference and learning are derives for these models. See e.g., [4, 66] for further details.

HIERARCHICAL BAYESIAN MODELS

In this section, we discuss the generalization of plate models to hierarchical Bayesian models, also known as multilevel models, in which we employ a soft version of parameter sharing, where parameters are encouraged to be similar through the use of hyper-priors and thus do not have to be identical [24]. Hierarchical priors allow us to introduce dependencies in parameter priors; a property that is particularly useful when we only have small amounts of training data and many parameters that can be assumed to be similar. In such situations, hierarchical priors increase the statistical power of the analysis by spreading the effect of observations between parameters with shared hyper-parameters [24].

Figure 9 shows an example of a varying coefficients model, which is the simplest example of a hierarchical Bayesian linear model [36]. The model depicts the generative process of the data x for N samples in M groups, with the following probabilistic description

$$x_i \sim \mathcal{N}(w_i, \sigma_i^2) \tag{32a}$$

$$w_i \sim \mathcal{N}(\mu_w, \sigma_w^2), \tag{32b}$$

along with appropriate (non-informative) prior distributions. We see that the hyper-parameter μ_w captures the general tendency of the w_i's over the groups, and σ_w expresses their variability.

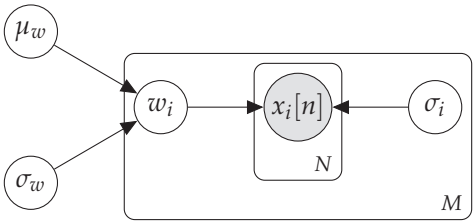


Figure 9: Meta-network for simple varying coefficients model.

In hierarchical models, we intentionally lose the property of parameter independence, which we explored extensively when formulating learning schemes for BNs in a complete data setting (Sec. 3.4). Thus, the posterior does not decompose into a product of independent terms, as the parameters are no longer independent in the prior and consequently in the posterior. Therefore, we generally need to resort to approximate inference for parameter learning, as in the case of parameter learning from incomplete data in Sec. 3.4 [24, 36]. Approximate inference algorithms can be slow to converge or get stuck when certain parameters (e.g., σ_w) follow a very short- or long-tailed distribution, thus the following reparameterization of Eq. 32 can be used in practice to decouple the model

$$x_i \sim \mathcal{N}(\mu_w + \sigma_w \eta_i, \sigma_i^2) \tag{33a}$$
$$\eta_i \sim \mathcal{N}(0, 1), \tag{33b}$$

where $w_i = \mu_w + \sigma_w \eta_i$ [36]. For example, Gelman et al. [36] recommend using this reparameterization when employing a Hamiltonian Monte Carlo (HMC) inference method, which is a class of state-of-the-art Markov chain Monte Carlo (MCMC) inference methods.

3.6 MODEL-BASED MACHINE LEARNING

Parts of this section appear in Paper D.

In this section, we describe an emerging methodology for applying machine learning (ML) called model-based machine learning (MBML) [32]. In traditional ML, the practitioner typically selects a suitable ML algorithm from the literature when faced with a new problem. If the algorithm requires modification to comply with the problem at hand, the practitioner must either modify the existing algorithm or combine the algorithm with other ML algorithms from the literature, both of which can be challenging. In contrast,

when applying MBML, a custom ML algorithm is formulated for a given problem by decomposing the algorithm construction into two distinct parts, namely (i) probabilistic model representation, and (ii) inference engine. The model representation covers the set of application specific assumptions made about the problem domain, i.e., the process that gave rise to the data, where any assumptions involving uncertainty are expressed using probabilities. The model representation is typically implemented in a compact modeling language from which custom code for learning and reasoning can be generated automatically based the chosen inference method. This is referred to as the inference engine. In some cases, of course, the MBML algorithm might correspond to an existing ML algorithm, while in other cases it will not. The important distinction here is that a MBML algorithm makes the model assumptions explicit through the model representation, while in traditional ML algorithms these are often implicitly defined [32, 67].

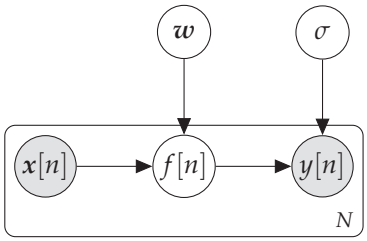
As indicated above, the MBML framework offers several advantages when defining a ML algorithm for a given problem, e.g., the ease with which highly tailored models can be created for specific applications, rapid model prototyping and modification for model comparison, compact model representation that permits debugging and collaboration, and the fact that practitioners can focus their attention on understanding a single modeling environment, as many traditional ML algorithms will appear as special cases of the MBML framework [32]. The framework could in principle be implemented using a variety of different approaches [32], but we will focus on an approach that leverages Bayesian inference in probabilistic graphical models, e.g., BNs, and recent developments in probabilistic programming (Paper D).

When implementing a MBML algorithm, we therefore first need to represent our assumptions about the problem domain using e.g., a BN. This means specifying the DAG $\mathcal{G}$, and a prior for the parameter vector $\boldsymbol{\Theta}_{\mathcal{G}}$, which holds the parameters of the corresponding (conditional) probability distributions. In this section, we will not distinguish between discrete and continuous probability distributions, or combinations hereof, as the considerations apply to either case. Thus, in general $\boldsymbol{\Theta}_{\mathcal{G}}$ simply holds the parameters needed to specify the model. As an example, the model representation for a Bayesian linear regression is shown in Alg. 3(1). Note that we have introduced a plate notation in the DAG to show that the relationship holds for every instance in the training set, and we account for the intercept term by including an additional element in the input vectors $\boldsymbol{x}$, which is equal to 1.

Second, we need to choose an inference engine, and, as mentioned, we will explore recent developments in probabilistic programming. Probabilistic programming is a programming paradigm that compiles a probabilistic model into a computer program from which inference code can be generated automatically, based on the chosen inference method. Moreover, it enables the combination of probabilistic and conventional deterministic code,

Algorithm 3: MBML algorithm statement defining a Bayesian linear regression.

1 **Probabilistic model representation**:



$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$$
$$y \sim \mathcal{N}(f(\boldsymbol{x}), \sigma)$$
$$w_j \sim \mathcal{N}(0, 10)$$
$$\sigma \sim \text{half-Cauchy}(0, 5)$$

2 **Inference engine**: e.g., Stan using Hamiltonian Monte Carlo

which provides a flexibility of modeling that reaches far beyond conventional graphical model notation [23, 32]. Probabilistic programming languages are programming languages that facilitate for probabilistic programming. Some examples are BUGS [68], Infer.NET [69], JAGS [70] and Stan [71]. As an example, we could implement our Bayesian linear regression model in Stan and use a Hamiltonian Monte Carlo inference method, see Alg. 3(2).

The Stan code for our Bayesian linear regression is shown in Lst. 1. From Lst. 1, we see that all we need to implement this model in Stan is 16 lines of code. Furthermore, we also see how easy it is to change the modeling assumptions. For instance, if we want to change the parametric form of the coefficients from normal to uniform, we simply have to exchange Lst. 1(13) to e.g., $\mathrm{w}[k] \sim \text{uniform}(-10, 10)$. For a general discussion on prior choice in probabilistic models, the interested reader is referred to e.g., [72].

Listing 1: Stan code for a Bayesian linear regression.

```
1  data {
2    int<lower=0> N;    // number of data items
3    int<lower=0> K;    // number of inputs
4    matrix[N, K] x;    // input matrix
5    vector[N] y;       // output vector
6  }
7  parameters {
8    vector[K] w;       // coefficients for inputs
9    real<lower=0> sigma;  // error scale
10 }
11 model {
12   for(k in 1:K)
13     w[k] ~ normal(0, 10); // prior for coefficients
14   sigma ~ cauchy(0, 5); // prior for error scale
15   y ~ normal(x * w, sigma);  // likelihood
16 }
```

At the present stage, MBML is a very powerful framework for modeling in problem domains, where the graph structure and the parametric form of the (conditional) probability distributions involved can be specified up-front in the probabilistic model representation. This is in contrast to the approach defined in Sec. 3.4, where we consider how to learn both the structure and the parameters of a BN model from data, of course, with the cost of having to discretize all continuous variables.

4 SYSTEM REPRESENTATIONS USING GAUSSIAN PROCESSES

As mentioned, supervised learning may be split into regression and classification problems, where the output in regression problems is the prediction of continuous quantities, and the output in classification problems is the prediction of discrete class labels. In this section, we consider how Gaussian processes (GPs), which are Bayesian non-parametric models, can be used for both supervised learning tasks. In Sec. 4.1, we consider single-output, as well as multi-output, GP regression, and in Sec. 4.2, we consider GP classification. Moreover, in Sec. 4.3, we show how the regression framework of GPs can be used to optimize black-box cost functions under the Bayesian optimization framework. Note that tutorials supporting the material covered in this section are available at the GitHub repository.

4.1 GAUSSIAN PROCESSES FOR REGRESSION

Parts of this section appear in Paper F.

In this section, we consider the regression of some response variable(s) y based on covariate(s) $\boldsymbol{x}$ of the form

$$y = f(\boldsymbol{x}) + \epsilon, \tag{34}$$

where we assume that the observed output y differ from the functional value $f(\boldsymbol{x})$ by additive noise ϵ. In the following, we show how the functional relationship f may be established by use of Gaussian process regression. First, we introduce Gaussian processes in a single-output setting and second, we proceed to cover the multi-output setting.

SINGLE-OUTPUT GAUSSIAN PROCESSES

A Gaussian process (GP) is a generalization of the multivariate Gaussian distribution over random variables to a probability distribution over functions, indexed by e.g., time or space, for which any finite subset of variables follows a Gaussian distribution. In this case, a random variable represents the value of the function $f(\boldsymbol{x})$ at location $\boldsymbol{x}$, which thus follows a Gaussian distribution. Just as a Gaussian distribution is completely specified by its mean and covariance, a GP is completely specified by its mean function m and covariance or kernel function k. With basis in Eq. 34, we assume f to be a non-linear, non-parametric function with a GP prior, i.e.,

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')), \tag{35}$$

where $m(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})]$ is the mean function, and $k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{C}[f(\boldsymbol{x}), f(\boldsymbol{x}')]$ is the positive semi-definite kernel, or covariance, function. This definition allows us to evaluate the mean function at an arbitrary input setting and assess how the value of the function at an input point covary with the value of the function at other points in input space. Therefore, we can think of a GP as defining a distribution over functions, and inference taking place directly in the space of functions [73].

Given a data set $\mathcal{D} = \{\hat{\boldsymbol{X}}, \hat{\boldsymbol{y}}\} = \{\boldsymbol{x}[n], y[n]\}_{n=1}^{N}$ of (potentially) vector-valued inputs and scalar outputs, we construct the GP prior by evaluating the mean and covariance function at the data points, which leads to a multivariate Gaussian distribution over the corresponding function values, i.e.,

$$f(\hat{\boldsymbol{X}}) \sim \mathcal{N}(m(\hat{\boldsymbol{X}}), k(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}})), \tag{36}$$

where $f(\hat{\boldsymbol{X}}) = \{f(\boldsymbol{x}[n])\}_{n=1}^{N}$ (Paper F). In Fig. 10, the GP model is depicted as a DAG [74]. The figure shows how the observed inputs $\hat{\boldsymbol{X}}$ are related to the observed outputs $\{y[n]\}_{n=1}^{N}$ through the latent function f, given the parameters of the latent function and the likelihood $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_{hid}, \boldsymbol{\Theta}_{obs}\}$. Note that the plate notation indicates independence amongst the outputs $y[n]$, given the corresponding functional value $f[n]$, where $f[n]$ is a shorthand notation for $f(\boldsymbol{x}[n])$. The likelihood $p(y|f)$ does not have to be Gaussian, but exact inference is generally not possible for non-Gaussian likelihoods [73].

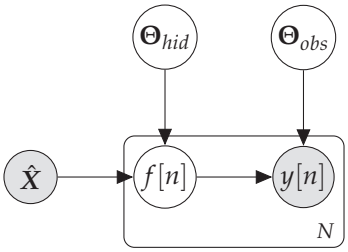


Figure 10: Meta-network of a Gaussian process.

Under proper normalization of the data, the expected value of the process can be assumed to be zero without loss of generality. The covariance function should then capture basic aspects of the process, such as stationarity, isotropicity, smoothness, and periodicity (Paper F). Here we will consider the Matérn family of kernels, which is a commonly used class of stationary kernels that are shift invariant. This kernel family includes a so-called smoothness parameter $\nu > 0$ that controls the degree to which samples from a GP are differentiable. Thus, samples from a GP with such a kernel are differentiable $\nu - 1$ times. As an example, the exponential and squared exponential kernel are special cases of Matérn kernels with $\nu = 1/2$ and $\nu \to \infty$,

respectively. Some commonly used Matérn kernels, labeled by their smoothness parameter, are shown below

$$k_{3/2}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \left(1 + \sqrt{3r^2}\right) \exp\left(-\sqrt{3r^2}\right) \tag{37a}$$

$$k_{5/2}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \left(1 + \sqrt{5r^2} + \frac{5}{3}r^2\right) \exp\left(-\sqrt{5r^2}\right) \tag{37b}$$

$$k_{SE}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}r^2\right), \tag{37c}$$

where $r^2 = \sum_i (x_i - x_i')^2 / l_i^2$. This class of kernels is thus parameterized by $\Theta_{hid} = \{\sigma_f, \boldsymbol{l}\}$, where $\sigma_f > 0$ is an amplitude parameter, and $\boldsymbol{l}$ is a vector of characteristic length-scale parameters $l_i > 0$. Kernel functions with learnable length-scale parameters are also known as automatic relevance detection (ARD) kernels, as the inverse of the length-scale indicates how relevant an input is with respect to inferences. Thus, if a length-scale has a very large value, the kernel will become almost independent of the corresponding input, and effectively remove it when performing inferences [73, 75, 76].

Now, assume that we want to make a prediction $f(\boldsymbol{x}_\star)$ at a new input $\boldsymbol{x}_\star$. First, we need to update our prior over f (Eq. 36) based on the observations using Bayes' rule. The posterior distribution $p(f(\hat{\boldsymbol{X}})|\mathcal{D})$ then reads

$$p(f(\hat{\boldsymbol{X}})|\mathcal{D}) = \frac{p(\hat{\boldsymbol{y}}|f(\hat{\boldsymbol{X}}))p(f(\hat{\boldsymbol{X}})|\hat{\boldsymbol{X}})}{p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{X}})}, \tag{38}$$

where the denominator, which is called the marginal likelihood or evidence to express the fact that the latent function f is marginalized out, is given by

$$p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{X}}) = \int p(\hat{\boldsymbol{y}}|f(\hat{\boldsymbol{X}}))p(f(\hat{\boldsymbol{X}})|\hat{\boldsymbol{X}})df(\hat{\boldsymbol{X}}). \tag{39}$$

Second, the posterior predictive distribution for $f(\boldsymbol{x}_\star)$, which is the one we are seeking, may be defined as

$$p(f(\boldsymbol{x}_\star)|\mathcal{D}, \boldsymbol{x}_\star) = \int p(f(\boldsymbol{x}_\star)|f(\hat{\boldsymbol{X}}), \hat{\boldsymbol{X}}, \boldsymbol{x}_\star)p(f(\hat{\boldsymbol{X}})|\mathcal{D})df(\hat{\boldsymbol{X}}). \tag{40}$$

Note that we have left out the dependence on $\boldsymbol{\Theta}$ in the above expressions to keep them uncluttered [77].

One attractive feature of the GP formulation is that exact inference is tractable under a Gaussian likelihood assumption, i.e.,

$$\hat{\boldsymbol{y}} \sim \mathcal{N}(f(\hat{\boldsymbol{X}}), \sigma^2 \mathbf{I}), \tag{41}$$

where $\mathbf{I}$ is the identity matrix. For this case, a closed form expression for Eq. 40 may be derived by direct application of the standard rules for conditioning of Gaussian distributed random variables [6]. Thus, we can write the

joint distribution of the observations $\hat{\boldsymbol{y}}$ and the function evaluated at the test location under the prior as

$$\begin{bmatrix} \hat{\boldsymbol{y}} \\ f(\boldsymbol{x}_\star) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) + \sigma^2\mathbf{I} & k(\hat{\boldsymbol{X}}, \boldsymbol{x}_\star) \\ k(\boldsymbol{x}_\star, \hat{\boldsymbol{X}}) & k(\boldsymbol{x}_\star, \boldsymbol{x}_\star) \end{bmatrix}\right), \tag{42}$$

where we have assumed the data to be properly normalized, so that $m(\boldsymbol{x}) = 0$ is appropriate. We can then restrict this joint prior distribution to contain only those functions that agree with the observations, i.e., by conditioning the prior for $f(\boldsymbol{x}_\star)$ on the observed data points. The closed form expression of the posterior predictive distribution for $f(\boldsymbol{x}_\star)$ (Eq. 40) then becomes

$$p(f(\boldsymbol{x}_\star)|\mathcal{D}, \boldsymbol{x}_\star, \boldsymbol{\Theta}) = \mathcal{N}(m_\star(\boldsymbol{x}_\star), k_\star(\boldsymbol{x}_\star, \boldsymbol{x}_\star)), \tag{43}$$

where $\boldsymbol{\Theta}$ denotes the set of model parameters, and $m_\star$ and $k_\star$ are defined as

$$m_\star(\boldsymbol{x}_\star) = \boldsymbol{k}_{\boldsymbol{x}_\star}(k(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) + \sigma^2\mathbf{I})^{-1}\hat{\boldsymbol{y}} \tag{44a}$$

$$k_\star(\boldsymbol{x}_\star, \boldsymbol{x}_\star) = k(\boldsymbol{x}_\star, \boldsymbol{x}_\star) - \boldsymbol{k}_{\boldsymbol{x}_\star}(k(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) + \sigma^2\mathbf{I})^{-1}\boldsymbol{k}_{\boldsymbol{x}_\star}^T, \tag{44b}$$

with $\boldsymbol{k}_{\boldsymbol{x}_\star}$ as a shorthand notation for $k(\boldsymbol{x}_\star, \hat{\boldsymbol{X}})$. Note that if we are interested in the corresponding noisy prediction $y_\star$, we simply have to add σ^2 to the predictive variance expression above. See [6, 73] for further details (Paper F).

Learning in a GP regression setting amounts to specifying the parameters of the covariance function k and the noise process ϵ in $\boldsymbol{\Theta}$. This may be achieved by use of a Bayesian approach, or by maximizing the log likelihood of the evidence as

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} \log p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{X}}, \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} -\frac{1}{2}\hat{\boldsymbol{y}}^T(k(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) + \sigma^2\mathbf{I})^{-1}\hat{\boldsymbol{y}} \\ &\quad - \frac{1}{2}\log|k(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) + \sigma^2\mathbf{I}| - \frac{N}{2}\log 2\pi. \end{aligned} \tag{45}$$

The right-hand side of the expression has three terms. The first term is the Mahalanobis distance between model predictions and the data, i.e., it quantifies the model fit to data. The second term penalizes model complexity, as smaller determinants (penalties) are found for smoother covariance matrices. The third and last term is a linear function of the data set size, which shows that the likelihood of the data generally decreases with increasing data set size. That is, the marginal likelihood provides an inherent trade-off between fit to data and model complexity [73, 76].

One approach for solving Eq. 45 is to use a gradient descent algorithm based on the partial derivative of the likelihood with respect to the model parameters $\boldsymbol{\Theta}$, which can be analytical derived, see e.g., [4, 6, 73], i.e.,

$$\frac{\partial}{\partial\theta_j}\log p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{X}}) = \frac{1}{2}\hat{\boldsymbol{y}}^T\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}\mathbf{K}_y^{-1}\hat{\boldsymbol{y}} - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}\right) \tag{46a}$$

$$\frac{\partial}{\partial \theta_j} \log p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{X}}) = \frac{1}{2}\mathrm{tr}\left(\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \mathbf{K}_y^{-1}\right)\frac{\partial \mathbf{K}_y}{\partial \theta_j}\right), \tag{46b}$$

where $\mathbf{K}_y = (k(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) - \sigma^2\mathbf{I})$, $\boldsymbol{\alpha} = \mathbf{K}_y^{-1}\hat{\boldsymbol{y}}$, and $\partial \mathbf{K}_y/\partial\theta_j$ is a matrix of derivatives with respect to θ_j evaluated at the input $\hat{\boldsymbol{X}}$. The form of $\partial \mathbf{K}_y/\partial\theta_j$ depends on the form of the input kernel and with respect to which parameter we are computing the gradient. As an example, consider the square exponential kernel Eq. 37c with a one dimensional input. For this case, we may write the output kernel as

$$K_y(x, x') = \sigma_f^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right) + \sigma^2\delta_{x,x'}, \tag{47}$$

where $\delta_{x,x'}$ is a delta function that includes σ^2 in the expression only when $x = x'$. The derivatives with respect to the kernel parameters then become

$$\frac{\partial K_y}{\partial \sigma} = 2\sigma\delta_{x,x'} \tag{48a}$$

$$\frac{\partial K_y}{\partial \sigma_f} = 2\sigma_f \exp\left(-\frac{(x-x')^2}{2l^2}\right) \tag{48b}$$

$$\frac{\partial K_y}{\partial l} = \sigma_f^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right)\left(\frac{(x-x')^2}{l^3}\right), \tag{48c}$$

where the expression for $\partial K_y/\partial l$ is found by direct application of the chain rule as

$$\begin{aligned}
\frac{\partial K_y}{\partial l} &= \frac{\partial K_y}{\partial g}\frac{\partial g}{\partial l} \\
&= \sigma_f^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right)\frac{\partial g}{\partial l} \\
&= \sigma_f^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right)\frac{\partial}{\partial l}\left(-\frac{(x-x')^2}{2}l^{-2}\right) \\
&= \sigma_f^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right)\left((x-x')^2 l^{-3}\right).
\end{aligned}$$

We often have constraints on the hyper-parameters, such as $\sigma \geq 0$. For this case, we can define $\theta = \log\sigma$, and then use the chain rule [4].

We note that a similar optimization problem is approached from a neural network perspective in Sec. 5. Section 5.1 provides some recommendations on how to avoid overfitting in high dimensional problems, see also [78], and Sec. 5.2 provides some guidance on how to optimize the expression. Moreover, in practical implementations of GPs we use the Cholesky decomposition, instead of directly inverting the predictive covariance matrix, as

it is faster and numerically more stable [73], see Alg. 4. Furthermore, for large N, an approximation scheme may be needed to reduce the computational burden of inference, due to the inversion of the predictive covariance, see e.g., [76] for further details.

Algorithm 4: Pseudo-code for inference and log marginal likelihood estimation in GP regression.

Input: $\mathcal{D}$, k, σ, $\boldsymbol{x}_\star$
Output: $m_\star(\boldsymbol{x}_\star)$, $k_\star(\boldsymbol{x}_\star, \boldsymbol{x}_\star)$, $\log p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{X}})$

$\mathbf{K}_y \leftarrow k(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) + \sigma^2\mathbf{I}$
$\mathbf{L} \leftarrow \text{cholesky}(\mathbf{K}_y)$
$\boldsymbol{\alpha} \leftarrow \mathbf{L}^T \backslash (\mathbf{L} \backslash \hat{\boldsymbol{y}})$; as $\mathbf{K}_y^{-1} = (\mathbf{L}^{-1})^T(\mathbf{L}^{-1})$
$\boldsymbol{k}_{\boldsymbol{x}_\star} \leftarrow k(\boldsymbol{x}_\star, \hat{\boldsymbol{X}})$
$m_\star(\boldsymbol{x}_\star) \leftarrow \boldsymbol{k}_{\boldsymbol{x}_\star}\boldsymbol{\alpha}$
$\mathbf{v} \leftarrow \mathbf{L} \backslash \boldsymbol{k}_{\boldsymbol{x}_\star}^T$
$k_\star(\boldsymbol{x}_\star, \boldsymbol{x}_\star) \leftarrow k(\boldsymbol{x}_\star, \boldsymbol{x}_\star) - \mathbf{v}^T\mathbf{v}$
$\log p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{X}}) \leftarrow -\frac{1}{2}\hat{\boldsymbol{y}}^T\boldsymbol{\alpha} - \sum_n \log \mathbf{L}_{n,n} - \frac{N}{2}\log 2\pi$

MULTI-OUTPUT GAUSSIAN PROCESSES

In this section, we extend the framework presented in the foregoing section to cover multi-output processes. In the general case, the different outputs might have different training set cardinalities, different input points, or even different input spaces. Thus, we have a training data set $\mathcal{D}_d = \{\hat{\boldsymbol{X}}_d, \hat{\boldsymbol{y}}_d\} = \{\boldsymbol{x}_d[n_d], y_d[n_d]\}_{n_d=1}^{N_d}$ for each output function f_d. In the geostatistics literature, a situation where each output has the same set of inputs is called isotopic, and a situation where each output is associated with a different set of inputs is called heterotopic [79]. The heterotopic case is also sometimes referred to as multi-task learning, but the distinction between multi-output learning and multi-task learning is not rigorously defined in the literature and varies from author to author. Álverez et al. [79] use the term multi-output learning or vector-valued learning to define the general class of problems, and the term multi-task learning for the class of problems where each component has different inputs, i.e., the heterotopic case.

Gaussian process modeling for vector-valued functions $\boldsymbol{f} = \{f_d\}_{d=1}^D$ follows the same approach as for the single-output case. In the single-output case, we consider a single process f evaluated at different values of $\boldsymbol{x}$, and in the multi-output case, we consider a set of processes $\boldsymbol{f}$ evaluated at different values of $\boldsymbol{x}$. The vector-valued function $\boldsymbol{f}$ is assumed to follow a GP, i.e.,

$$\boldsymbol{f}(\boldsymbol{x}) \sim \mathcal{GP}(\boldsymbol{m}(\boldsymbol{x}), \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}')), \tag{49}$$

where $\boldsymbol{m}(\boldsymbol{x}) = \{m_d(\boldsymbol{x})\}_{d=1}^D$ is the mean function of the outputs, and $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}')$ is a positive semi-definite, matrix-valued function with entries $(\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}'))_{d,d'}$, such that the entries correspond to the covariances between the outputs $f_d(\boldsymbol{x})$ and $f_{d'}(\boldsymbol{x}')$ [79].

The prior distribution over $\boldsymbol{f}$ takes the form

$$\boldsymbol{f}(\hat{\boldsymbol{X}}) \sim \mathcal{N}(\boldsymbol{m}(\hat{\boldsymbol{X}}), \boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}})), \tag{50}$$

where $\boldsymbol{f}(\hat{\boldsymbol{X}}) = \{f_d(\boldsymbol{x}[n]) | n = 1, ..., N; d = 1, ..., D\}$, $\boldsymbol{m}(\hat{\boldsymbol{X}})$ is a vector that concatenates the mean vectors of the outputs, which under proper normalization of the data can be assumed to be the zero vector without loss of generality, and $\boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}})$ is a block partitioned matrix defined as

$$\boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) = \begin{bmatrix} (\boldsymbol{K}(\hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_1))_{1,1} & \cdots & (\boldsymbol{K}(\hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_D))_{1,D} \\ (\boldsymbol{K}(\hat{\boldsymbol{X}}_2, \hat{\boldsymbol{X}}_1))_{2,1} & \cdots & (\boldsymbol{K}(\hat{\boldsymbol{X}}_2, \hat{\boldsymbol{X}}_D))_{2,D} \\ \vdots & \ddots & \vdots \\ (\boldsymbol{K}(\hat{\boldsymbol{X}}_D, \hat{\boldsymbol{X}}_1))_{D,1} & \cdots & (\boldsymbol{K}(\hat{\boldsymbol{X}}_D, \hat{\boldsymbol{X}}_D))_{D,D} \end{bmatrix}. \tag{51}$$

To simplify the following notations, we will assume that $\{N_d\}_{d=1}^D = N$, but this is not a necessary condition, and the formulations may readily be adjusted to cover the general case [79] (Paper F).

If we again assume a Gaussian likelihood model, i.e.,

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{f}(\boldsymbol{x}), \Sigma), \tag{52}$$

where Σ is a diagonal matrix of elements $\{\sigma_d^2\}_{d=1}^D$, and we assume the mean function $\boldsymbol{m}(\boldsymbol{x}) = \boldsymbol{0}$, the predictive distribution for a new $\boldsymbol{x}_\star$ data point has a closed form solution, i.e.,

$$p(\boldsymbol{f}(\boldsymbol{x}_\star) | \mathcal{D}, \boldsymbol{x}_\star, \boldsymbol{\Theta}) = \mathcal{N}(\boldsymbol{m}_\star(\boldsymbol{x}_\star), \boldsymbol{K}_\star(\boldsymbol{x}_\star, \boldsymbol{x}_\star)) \tag{53}$$

where $\boldsymbol{\Theta}$ denotes the set of model parameters, and $\boldsymbol{m}_\star$ and $\boldsymbol{K}_\star$ are defined as

$$\boldsymbol{m}_\star(\boldsymbol{x}_\star) = \boldsymbol{K}_{\boldsymbol{x}_\star}(\boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) + \boldsymbol{\Sigma})^{-1} \hat{\boldsymbol{y}}_c \tag{54a}$$

$$\boldsymbol{K}_\star(\boldsymbol{x}_\star, \boldsymbol{x}_\star) = \boldsymbol{K}(\boldsymbol{x}_\star, \boldsymbol{x}_\star) - \boldsymbol{K}_{\boldsymbol{x}_\star}(\boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) + \boldsymbol{\Sigma})^{-1} \boldsymbol{K}_{\boldsymbol{x}_\star}^T \tag{54b}$$

with $\hat{\boldsymbol{y}}_c$ being a vector of length $N \times D$ that concatenates the observed output vectors, $\boldsymbol{\Sigma} = \Sigma \otimes \mathbf{I}_N$ is the Kronecker product between the noise covariance matrix and an identity matrix of size N, and $\boldsymbol{K}_{\boldsymbol{x}_\star} = (\boldsymbol{K}(\boldsymbol{x}_\star, \hat{\boldsymbol{X}}_d))_{d,d'}$. Note that if we are interested in the corresponding noisy predictions $\boldsymbol{y}_\star$, we simply have to add Σ to the predictive variance expression above [79].

If we further assume that the kernel function is separable, we can form the kernel as a product between an input kernel and an output kernel as

$$(\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}'))_{d,d'} = k_x(\boldsymbol{x}, \boldsymbol{x}') k_y(d, d'), \tag{55}$$

where k_x and k_y encode the covariances between the inputs and outputs, respectively. This is referred to as the intrinsic coregionalization model (ICM) in the Bayesian literature. Other more general kernel structures are sums of separable kernel, as in the linear model of coregionalization (LMC) and process convolutions. See e.g., [79, 80] for further details (Paper F).

As for single-output GPs, learning in a multi-output GP setting amounts to specifying the parameters of the covariance function $\boldsymbol{K}$ and the noise process $\boldsymbol{\epsilon}$ in $\boldsymbol{\Theta}$, which may again be achieved by maximizing the log likelihood of the evidence, or by means of Bayesian inference [81].

4.2 GAUSSIAN PROCESSES FOR CLASSIFICATION

In this section, we consider how to perform classification with a GP. We start by considering the case of binary classification, i.e., $y \in \{0, 1\}$, thus we wish to represent a binary random variable. For this case, our model should output a single number, i.e., $P(y = 1|\boldsymbol{x}) \in [0, 1]$, but the GPs considered so far make predictions in $\mathbb{R}$. However, we can adapt a GP to the classification setting by transforming the output of the GP using an appropriate output transformation function like the logistic sigmoid as

$$f(\boldsymbol{x}) = \text{sigm}(z(\boldsymbol{x})) = \frac{1}{1 + \exp(-z(\boldsymbol{x}))}, \tag{56}$$

which satisfies $0 \leq f \leq 1$ such that $P(y = 0|\boldsymbol{x}) = 1 - f(\boldsymbol{x})$, and $z(\boldsymbol{x})$ represents the GP. Thus, we obtain a non-Gaussian stochastic process over function f [6]. Note that because the Gaussian prior is not conjugate to the Bernoulli likelihood, the standard rules for conditioning of Gaussian distributed random variables, as defined for the regression setting, on longer apply, and thus exact inference is not feasible [4]. This is not a unique feature of GP classification problems but also applies when a non-Gaussian likelihood is considered in the regression setting [73].

In this setup, z is a nuisance function, which is a function that is of no immediate interest by itself, but which needs to be accounted for in the analysis of function f. That is, given a data set $\mathcal{D} = \{\hat{\boldsymbol{X}}, \hat{\boldsymbol{y}}\} = \{\boldsymbol{x}[n], y[n]\}_{n=1}^{N}$, the posterior predictive distribution for $f(\boldsymbol{x}_\star)$ is given by

$$p(f(\boldsymbol{x}_\star)|\mathcal{D}, \boldsymbol{x}_\star, \boldsymbol{\Theta}) = \int p(f(\boldsymbol{x}_\star)|z(\boldsymbol{x}_\star))p(z(\boldsymbol{x}_\star)|\mathcal{D}, \boldsymbol{x}_\star, \boldsymbol{\Theta})dz(\boldsymbol{x}_\star), \tag{57}$$

where $p(f(\boldsymbol{x}_\star)|z(\boldsymbol{x}_\star)) = \text{sigm}(z(\boldsymbol{x}_\star))$, and $p(z(\boldsymbol{x}_\star)|\mathcal{D}, \boldsymbol{x}_\star, \boldsymbol{\Theta})$ is the posterior predictive distribution for $z(\boldsymbol{x}_\star)$ (Eq. 40). Note that there are no additional parameters in this model than the ones defining the GP, as the output values are assumed to be correctly labeled [6]. As mentioned, this integral is analytically intractable, and thus we need to resort to an approximate inference method, such as e.g., Monte Carlo sampling, variational inference, or the

Laplace approximation [6, 73]. The interested reader is referred to e.g., [73], as well as Appendix A, for further details.

In the case of standard multi-class classification, each input is assigned to one of D mutually exclusive classes. For this case, the output variables $y_d \in \{0, 1\}$ have a 1–D encoding, and the model should output $\mathbf{f} = \{f_d(\boldsymbol{x}) = P(y_d = 1|\boldsymbol{x})\}_{d=1}^{D}$, thus we wish to represent a discrete random variable with D states. This can be accomplished with a softmax output transformation function as

$$f_d(\boldsymbol{x}) = \text{softmax}(\boldsymbol{z}(\boldsymbol{x}))_d = \frac{\exp(z_d(\boldsymbol{x}))}{\sum_{k'} \exp(z_{d'}(\boldsymbol{x}))}, \tag{58}$$

which satisfies $0 \leq f_d \leq 1$ and $\sum_d f_d = 1$. The softmax function can thus be viewed as a generalization of the sigmoid function, which is used above to define the probability distribution over a binary random variable [6, 82].

As indicated, the basis for the multi-class case is a multi-output GP $\boldsymbol{z}$, where the individual GPs z_d may be assumed to be independent, i.e., $\boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}})$ in Eq. 51 is a block-diagonal matrix. This again produces a non-Gaussian process over functions, and because the Gaussian prior is not conjugate to the categorical likelihood, we need to rely on the methods explained above to conduct approximate inferences on $\boldsymbol{f}$ [6, 73].

4.3 GAUSSIAN PROCESSES FOR OPTIMIZATION

Parts of this section appear in Paper E.

In this section, we consider the problem of finding a global minimizer of a function f defined by covariate(s) $\boldsymbol{x}$ as

$$\boldsymbol{x}_{\min} = \arg\min_{\boldsymbol{x} \in \boldsymbol{X}} f(\boldsymbol{x}). \tag{59}$$

Bayesian optimization (BO) is a sequential model-based approach for optimizing a cost function, which is computationally expensive to evaluate and/or has no closed-form expression, but from which we can obtain (noisy) observations [76]. BO techniques are some of the most efficient optimization techniques in terms of the number of functional evaluations required, due to their use of Bayesian updating, i.e.,

$$p(f|\mathcal{D}) \propto p(\mathcal{D}|f)p(f), \tag{60}$$

where $\mathcal{D} = \{\boldsymbol{x}[n], y[n]\}_{n=1}^{N}$ is a data set of observations of the cost function [83]. One example is the optimization of hyper-parameters for a general machine learning model, where the objective is to find the hyper-parameters that result in the lowest validation loss. Traditionally, strategies such as manual-, grid- and random-search are employed for the optimization, where

random-search is found superior to grid-search [84], but BO techniques have been shown to outperform manual- and random-search in terms of both performance and efficiency, see e.g., [85–87] (Paper E).

Bayesian optimization using GPs leverage Bayes rule to build a surrogate model of the cost function with a prior over functions and combines it with new observations to form a posterior over functions, in an online fashion. This allows for a utility-based selection of the next point to sample from the cost function, which should account for the trade-off between exploration (sampling from areas where the uncertainty is high) and exploitation (sampling from areas that are likely to provide an improvement over the current best setting $\boldsymbol{x}_{\text{min}}^{(t)}$) [76, 83].

Fitting a surrogate model given a data set is an exercise of GP regression, as presented in Sec. 4.1, but finding the next sample point from the cost function requires a utility function, which is commonly referred to as an acquisition function in the BO literature. The acquisition function takes into account the mean and variance information of the predictions, over the domain of interest, to model the utility of new sampling points, such that a high acquisition value corresponds to potentially low cost values, either because the prediction is low or the uncertainty is high, or both. The argmax value of the acquisition function is chosen as the new sampling point of the cost function, and the process is repeated, considering the data set $\mathcal{D}$ augmented with the new sample point $\{\boldsymbol{x}[N+1], y[N+1]\}$ [83] (Paper E). Pseudo-code for a BO implementation is provided in Alg. 5.

Algorithm 5: Pseudo-code for Bayesian optimization.

Input: $\mathcal{D}$, f, $p(f)$, A, T
Output: $\boldsymbol{x}_{\text{min}}$

Initialization: $(f_{\text{min}}^{(0)}, \boldsymbol{x}_{\text{min}}^{(0)})$
for $t = 0, ..., T$ **do**
  Select $\boldsymbol{x}[N+1]$ by optimizing acquisition function $\text{A}(\boldsymbol{x}|\mathcal{D})$ over current surrogate model; $\boldsymbol{x}[N+1] = \arg\max_{\boldsymbol{x}} \text{A}(\boldsymbol{x}|\mathcal{D})$
  Query cost function at $\boldsymbol{x}[N+1]$ to obtain $y[N+1]$
  Update data set; $\mathcal{D} \leftarrow \{\mathcal{D}, \{\boldsymbol{x}[N+1], y[N+1]\}\}$, $N \leftarrow N+1$
  Update surrogate model; $p(f|\mathcal{D}) \leftarrow p(\mathcal{D}|f)p(f)/p(\mathcal{D})$
  Update $(f_{\text{min}}^{(t+1)}, \boldsymbol{x}_{\text{min}}^{(t+1)})$; $f_{\text{min}}^{(t+1)} \leftarrow f(\boldsymbol{x}_{\text{min}}^{(t+1)}) = \min_{\boldsymbol{x} \in \hat{\boldsymbol{X}}} f(\boldsymbol{x})$
end

Some improvement-based acquisition functions used for Bayesian optimization are the probability of improvement A_{PI}, and the expected improvement A_{EI}. The probability of improvement is defined as

$$\text{A}_{PI}(\boldsymbol{x}) = P(f(\boldsymbol{x}) < f_{\text{min}}^{(t)}), \tag{61}$$

where $f_{\min}^{(t)} = \min_{\boldsymbol{x} \in \hat{\boldsymbol{X}}} f(\boldsymbol{x})$ is the best current value; and the expected improvement is defined as

$$\mathrm{A}_{EI}(\boldsymbol{x}) = \mathbb{E}\left[\max(0, f_{\min}^{(t)} - f(\boldsymbol{x}))\right]. \tag{62}$$

Note that analytical expressions can be derived for both A_{PI} and A_{EI}, when a GP surrogate is used; and e.g., $f_{\min}^{(t)} = \min_{\boldsymbol{x} \in \hat{\boldsymbol{X}}} f_\star(\boldsymbol{x})$ can be used in case of noisy observations [76, 77]. Another idea is to explore the lower confidence bounds of the surrogate model to construct an acquisition function, i.e.,

$$\mathrm{A}_{LCB}(\boldsymbol{x}) = -f_\star(\boldsymbol{x}) + \kappa \tau_\star(\boldsymbol{x}), \tag{63}$$

where $\tau_\star(\boldsymbol{x}) = \sqrt{k_\star(\boldsymbol{x}, \boldsymbol{x})}$, and κ is a hyper-parameter to balance exploitation and exploration [77]. See e.g., [76] for a detailed listing of acquisition functions used in practice. Note that the choice of probabilistic model is often considered more important than the choice of acquisition function [76], and in this regard the Matérn 5/2 kernel is recommended [75]. Moreover, both the expected improvement and the confidence lower bound acquisition function have proven to be efficient in the number of functional evaluations required to find the global optimum of a variety multi-modal, black-box functions [75, 88, 89].

Section 4.1 considers how to handle the hyper-parameters of the GP surrogate, but we have not considered how to optimize the acquisition function. In practice, a common approach is to use a quasi-Newton hill-climbing search with multiple restarts when the gradients can be analytically derived or numerically approximated, and a derivative-free optimizer, like the divided rectangles approach, when this is not the case. See e.g., [76] for a review of other approaches used in practice for optimizing acquisition functions.

In this section, BO is posed as an unconstrained, sequential optimization problem, where an experiment is completed before the next is proposed, but procedures for both parallel implementation and constraint optimization are found in the literature, together with approaches that include the evaluation time of an input configuration in the utility assessment, see e.g., [75, 76, 90]. For further details on BO in general, and BO using GPs in particular, the interested reader is referred e.g., [76, 77, 83].

5 SYSTEM REPRESENTATIONS USING NEURAL NETWORKS

In the era of deep learning, it is almost unthinkable not to consider the most widely used modeling approach for this purpose, namely (artificial) neural networks (NNs). A NN is a data processing model composed of a layered set of interconnected processing units (neurons), which is inspired by the way the human brain process information. In Sec. 5.1, we introduce the multilayer perceptron for both regression and classification tasks, and we consider some ways to regularize NNs. This is followed by Sec. 5.2, where we consider how to learn NNs. Finally, Secs. 5.3 and 5.4 introduce some other common classes of NN architectures and Bayesian NNs, respectively. Note that tutorials supporting the material covered in this section are available at the GitHub repository.

5.1 MULTILAYER PERCEPTRONS

The most widely used and archetypal NN architecture is the multilayer perceptron (MLP), also sometimes referred to as the (classical) feed-forward neural network. We can generally view a MLP as a parametric non-linear function f from a vector $\boldsymbol{x}$ of realized input variables to a vector $\boldsymbol{y}$ of realized output variables [6]. MLPs allow signals to travel one way only, i.e., from input to output, and there are no feedback loops, i.e., the output of any layer does not affect that same layer [82]. Figure 11 shows this model as a directed acyclic graph (DAG), with fully connected layers. The DAG illustrates the functional representation of MLPs, i.e., $f(\boldsymbol{x}) = f^{[2]}(f^{[1]}(\boldsymbol{x}))$. In this case, $f^{[1]}$ is referred to as the first layer, and $f^{[2]}$ is referred to as the second layer. The length of this chain is called the depth of the model, and the number of units in each hidden layer specifies the width [82]. Generally, the layers $f^{[j]}$ perform an affine transformation defined by parameters $\boldsymbol{W}^{[j]}$, followed by a fixed non-linear transformation using a so-called activation function $g^{[j]}$. Thus, the mapping within the first layer becomes

$$\boldsymbol{h} = f^{[1]}(\boldsymbol{x}) = g^{[1]}((\boldsymbol{W}^{[1]})^T\boldsymbol{x}) = g^{[1]}(\boldsymbol{z}), \tag{64}$$

where we assume that $\boldsymbol{x}$ is augmented by the value $x_0 = 1$ to include a bias term, and $\boldsymbol{z} = (\boldsymbol{W}^{[1]})^T\boldsymbol{x}$. The mapping from input to output performed by the function f can now be formulated as

$$f(\boldsymbol{x}) = g^{[2]}(\boldsymbol{W}^{[2]})^T\boldsymbol{h}) = g^{[2]}\left((\boldsymbol{W}^{[2]})^T g^{[1]}((\boldsymbol{W}^{[1]})^T\boldsymbol{x})\right), \tag{65}$$

where $g^{[2]}$ is the output activation function, and we assume $\boldsymbol{x}$, as well as $\boldsymbol{h}$, to be augmented by the value 1 to include a bias term. Moreover, as we only

consider one output in this case, the matrix $\boldsymbol{W}^{[2]}$ of the second layer becomes a vector. The output unit activation function defines the task performed by the MLP, e.g., regression or classification, and it needs to be chosen in consistency with the cost function used to fit the model [6, 82].

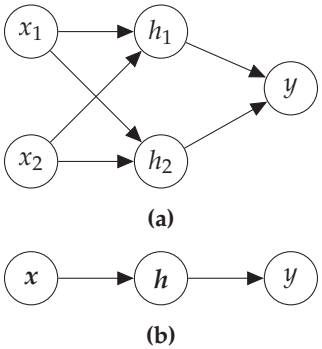


Figure 11: Example of a MLP model: (a) graph showing the individual nodes of each layer, (b) graph showing the nodes in each layer as vectors.

When implementing a NN, three important design choices need to be made, namely (i) the activation function of the hidden layers, (ii) the cost function to be optimized, and (iii) the network structure. The default recommendations are to use the rectified linear unit (ReLU) as activation function, and the cross-entropy as cost function, but a lot of different combinations can be found in the literature. Generally, though, NN technologies employ a gradient-based optimization procedure for training, which requires a smooth cost function. Furthermore, when designing a MLP, a uniform width can be assumed as default for all hidden layers, in order to reduce the set of possible network structures to consider in a cross-validation scheme. This way, only one width and one depth parameter need to be selected [82].

REGRESSION

In this section, we revisit the regression problem of Eq. 34, thus we consider the regression of some response variable(s) y based on covariate(s) x of the form

$$y = f(\boldsymbol{x}) + \epsilon,$$

under a Gaussian noise assumption, i.e., $y \sim \mathcal{N}(f(\boldsymbol{x}), \sigma^2)$. Given a data set $\mathcal{D} = \{\hat{\boldsymbol{X}}, \hat{\boldsymbol{y}}\} = \{\boldsymbol{x}[n], y[n]\}_{n=1}^{N}$ of (potentially) vector-valued inputs and scalar outputs, the likelihood of the model decomposes as

$$p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{X}}, \boldsymbol{\Theta}) = \prod_{n=1}^{N} p(y[n]|\boldsymbol{x}[n], \boldsymbol{\Theta}), \tag{66}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{w}, \sigma^2\}$ is the set of model parameters; $\boldsymbol{w}$ is the set of all network weights and σ^2 is the output variance [82].

The convention in the NN community is to minimize a cost function rather than maximizing a (log) likelihood, so we will follow this convention [4, 6]. Taking the negative logarithm on both sides of Eq. 66, we arrive at the negative log likelihood

$$-\log p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{X}}, \boldsymbol{\Theta}) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y[n] - f(\boldsymbol{x}[n]))^2 + \frac{N}{2} \log(2\pi\sigma^2). \tag{67}$$

The cost function used for training most modern NNs is the negative log likelihood, which is equivalent to the cross entropy between the training data and the model distribution [82].

We first consider the estimation of $\boldsymbol{w} \in \boldsymbol{\Theta}$. Maximizing the likelihood function with respect to $\boldsymbol{w}$ is equivalent to minimizing the sum-of-squares cost function, i.e.,

$$L(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} (y[n] - f(\boldsymbol{x}[n]))^2. \tag{68}$$

Minimization of Eq. 68 produces a maximum likelihood estimate $\hat{\boldsymbol{w}}$ [6]. The variance parameter can now be estimated by minimizing the negative log likelihood (Eq. 67) to give

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{n=1}^{N} (y[n] - f(\boldsymbol{x}[n]))^2. \tag{69}$$

This approach is prone to overfitting, thus regularization measures should always be considered when fitting a model using this approach. Taking a Bayesian perspective, a specific kind of regularization emerges naturally for this problem, namely weight decay, also known as L^2 norm, ridge, and Tikhonov regularization. Combining Eq. 66 with a Gaussian prior on $\boldsymbol{w}$, i.e.,

$$p(\boldsymbol{w}|\tau^2) = \mathcal{N}(\boldsymbol{0}, \tau^2\mathbf{I}), \tag{70}$$

the posterior for $\boldsymbol{w}$ takes the following form

$$p(\boldsymbol{w}|\mathcal{D}, \sigma^2, \tau^2) \propto p(\boldsymbol{w}|\tau^2) p(\mathcal{D}|\boldsymbol{w}, \sigma^2). \tag{71}$$

This gives rise to the following cost function for the weights

$$L(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} (y[n] - f(\boldsymbol{x}[n]))^2 + \frac{\lambda}{2} \boldsymbol{w}^T \boldsymbol{w}, \tag{72}$$

where $\lambda = \sigma^2/\tau^2$ [4, 6, 91]. The minimization of Eq. 72 thus provides a maximum a-posteriori (MAP) estimate of $\boldsymbol{w}$. Note that often the bias parameters

are omitted from the regularizer, as their inclusion causes the solution to depend on the choice of origin for the output variable, or they may be included with their own regularization parameters. Furthermore, it is recommended to use individual regularization terms for each layer. See e.g., [6] for further details.

Some other ways to prevent overfitting are early stopping, data set augmentation, and dropout. In early stopping, the validation error is monitored during training, and the training process is terminated at the point of smallest validation error. It can be shown that early stopping behaves like weight decay regularization [6]. In data set augmentation, the training data are augmented by perturbed versions of itself. In image classification this perturbation could be different transformations, like translation and rotation, on the data set. In cases where it is not apparent how to best transform the data, input noise injection can be performed under the assumption that the model should still be able to perform the task with random noise added to the inputs [82]. For dropout regularization, we employ a minibatch-based learning algorithm that takes small steps, like stochastic gradient decent, and at each training step, we randomly remove non-output units from the network structure. In this regard, an input unit is typically included with probability 0.8, and a hidden unit is included with probability 0.5. This can be seen as an approximation to bootstrap aggregating (bagging), which is an ensemble method that performs model averaging [82, 92]. Note that bagging is also covered in Appendix B.

CLASSIFICATION

In this section, we consider how to perform classification with a MLP. We start by considering the case of binary classification, i.e., $y \in \{0, 1\}$, thus we wish to represent a binary random variable. For this case, our model should output a single number, i.e., $P(y = 1|\boldsymbol{x})$, which we can accomplish with e.g., the logistic sigmoid output unit as

$$f(\boldsymbol{x}) = \text{sigm}(z) = \frac{1}{1 + \exp(-z)}, \tag{73}$$

which satisfies $0 \leq f \leq 1$ such that $P(y = 0|\boldsymbol{x}) = 1 - f(\boldsymbol{x})$ [6, 82]. Given a data set $\mathcal{D} = \{\hat{\boldsymbol{X}}, \hat{\boldsymbol{y}}\} = \{\boldsymbol{x}[n], y[n]\}_{n=1}^{N}$, this setup reveals a likelihood function based on the Bernoulli distribution, i.e.,

$$P(\hat{\boldsymbol{y}}|\hat{\boldsymbol{X}}, \boldsymbol{\Theta}) = \prod_{n=1}^{N} f(\boldsymbol{x}[n])^{y[n]} (1 - f(\boldsymbol{x}[n]))^{(1-y[n])}. \tag{74}$$

Taking the negative logarithm on both sides, we arrive at the negative log likelihood

$$-\log P(\hat{\boldsymbol{y}}|\hat{\boldsymbol{X}},\boldsymbol{\Theta}) = -\sum_{n=1}^{N} y[n]\log f(\boldsymbol{x}[n]) + (1-y[n])\log(1-f(\boldsymbol{x}[n])), \quad (75)$$

which equivalently defines the cross-entropy cost function for the weight parameters of the model, i.e.,

$$L(\boldsymbol{w}) = -\sum_{n=1}^{N} y[n]\log f(\boldsymbol{x}[n]) + (1-y[n])\log(1-f(\boldsymbol{x}[n])). \quad (76)$$

Note that there are no additional parameters in this model, i.e., $\boldsymbol{\Theta} = \boldsymbol{w}$, as the output values are assumed to be correctly labeled [6, 82]. Moreover, the cost function may be rewritten in terms of the softplus function, see [82] for further details.

In the case of standard multi-class classification, each input is assigned to one of D mutually exclusive classes. For this case, the output variables $y_d \in \{0,1\}$ have a 1–D encoding, and the model should output $f(\boldsymbol{x}) = \{P(y_d = 1|\boldsymbol{x})\}_{d=1}^{D}$, thus we wish to represent a discrete random variable with D states. This can be accomplished using the softmax output activation function as

$$f_d(\boldsymbol{x}) = \text{softmax}(\boldsymbol{z})_d = \frac{\exp(z_d)}{\sum_{d'}\exp(z_{d'})}, \quad (77)$$

which satisfies $0 \leq f_d \leq 1$ and $\sum_d f_d = 1$ [6, 82]. Given a data set $\mathcal{D} = \{\hat{\boldsymbol{X}},\hat{\boldsymbol{Y}}\} = \{\boldsymbol{x}[n],\boldsymbol{y}[n]\}_{n=1}^{N}$, we arrive at a likelihood function based on the categorical distribution, also called the multinoulli distribution, i.e.,

$$P(\hat{\boldsymbol{Y}}|\hat{\boldsymbol{X}},\boldsymbol{\Theta}) = \prod_{n=1}^{N}\prod_{d=1}^{D} f_d(\boldsymbol{x}[n])^{y_d[n]}. \quad (78)$$

Taking the negative logarithm on both sides, we get the negative log likelihood

$$-\log P(\hat{\boldsymbol{Y}}|\hat{\boldsymbol{X}},\boldsymbol{\Theta}) = -\sum_{n=1}^{N}\sum_{d=1}^{D} y_d[n]\log f_d(\boldsymbol{x}[n]), \quad (79)$$

which equivalently defines the cross-entropy cost function for the weight parameters of the model, i.e.,

$$L(\boldsymbol{w}) = -\sum_{n=1}^{N}\sum_{d=1}^{D} y_d[n]\log f_d(\boldsymbol{x}[n]). \quad (80)$$

Note that again there are no additional parameters in the model, i.e., $\boldsymbol{\Theta} = \boldsymbol{w}$, as the output values are assumed to be correctly labeled [6, 82].

It is interesting to note that the output activation functions used in this section correspond to the output transformations for Gaussian process (GP) classification studied in Sec. 4.2, but the resemblance does not end here. It has been known for some time that a single-layer MLP with an i.i.d. prior on the weights is equivalent to a GP, in the limit of infinite width, see e.g., [93], and the same property has recently been shown to hold for deep NNs, see e.g., [94]. This correspondence has led to new research activities on how to convert deep NNs into GPs in order to perform exact Bayesian inference for regression tasks, see e.g., [94–96].

5.2 LEARNING NEURAL NETWORKS

Learning, or training, is the process of determining the weights of the model, along with any additional parameters, by introducing a training set $\mathcal{D}$. The optimal choice of weights is the one that produces the smallest aggregated error, i.e., the lowest cost, over the training set, while monitoring target metrics on the validation set, e.g., accuracy, to avoid overfitting.

As apparent from Sec. 5.1, the appropriate choice of cost function depends on the output unit activation function used in the network, and thus the problem at hand. In general, the non-linearities in the network function $f(\boldsymbol{x})$ causes the cost function $L(\boldsymbol{w})$ to become non-convex, and thus makes the task of finding the weights an optimization problem. This implies finding a stationary point (minimum) in weight space, i.e., $\nabla_{\boldsymbol{w}} L(\boldsymbol{w}) \approx \mathbf{0}$, by an iterative numerical procedure, i.e., $\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + \Delta \boldsymbol{w}^{(t)}$ where t is the iteration step [6, 97].

The optimization is performed in a sequence of steps, where each step involves two distinct phases: (i) Finding the derivatives of the cost function with respect to the weights, and (ii) adapting the weights by applying the derivatives. Normally, the back-propagation technique is adopted in phase (i), while minibatch-based learning algorithms, like stochastic gradient decent, are applied in phase (ii) [6, 82, 97].

BACKPROPAGATION

In MLPs, information is propagated through the network from inputs to outputs leading to a scalar cost for each training example. This is called forward-propagation. The back-propagation algorithm, often referred to as backprop, formalizes how information is propagated backwards from the costs through the network in order to compute the gradients with respect to the parameters of the network [82].

By use of the chain rule of calculus, it is straightforward to express the gradients, but numerical evaluation of such expressions can be computationally expensive. Back-propagation is a specific instance of dynamic program-

ming that avoids repeating common sub-expression evaluations by storing intermediate results in the evaluation of the gradient $\nabla_{\boldsymbol{w}} L(\boldsymbol{w})$ and filling in the components successively, as they are computed, when passing backward through the network. See e.g., [82] for further details.

OPTIMIZATION ALGORITHMS

As apparent from Sec. 5.1, the cost function typically decomposes as a sum in the training examples. Therefore, optimization algorithms adopted for NN learning usually compute each parameter update based only on a randomly selected subset of the training examples. Among others, this reduces the computational cost of each parameter update compared to using the entire training set, which is computationally expensive for large data sets [82].

Optimization algorithms that operate on randomly selected subsets of the training data are called minibatch (stochastic) methods. The canonical example of a minibatch method is stochastic gradient decent (SGD). In SGD, the gradient of minibatch k is calculated as

$$\nabla_{\boldsymbol{w}}^{(k)} = \frac{1}{N_k} \sum_{n=1}^{N_k} \nabla L(\boldsymbol{w}^{(k)} | \boldsymbol{x}^{(k)}[n], \boldsymbol{y}^{(k)}[n]), \tag{81}$$

where $\nabla_{\boldsymbol{w}}^{(k)}$ is the gradient of minibatch k of size N_k, and $\nabla L(\boldsymbol{w}|\boldsymbol{x}^{(k)}[n], \boldsymbol{y}^{(k)}[n])$ are the gradient contributions from the individual members of minibatch k. It is common to use a power of 2 batch size in the range from 32 to 256. The updating rule then becomes

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} - r\nabla_{\boldsymbol{w}}^{(k)}, \tag{82}$$

where r is the learning rate, which may be gradually decreased during training. The typical recommendation is to cycling through the training set several times, unless the training set is extremely large. When multiple epochs are used, though, only the first epoch provides unbiased estimates of the gradients, but the additional epochs usually provide enough benefit in terms of error reduction to make up for this fact [82].

More advanced optimization algorithms leverage e.g., information contained in the second-order derivatives of the cost function, but this comes with the cost of a larger batch size requirement. Furthermore, we note that a reasonably new idea in deep learning called batch normalization [98], which adaptively reparametrizes the layers, can improve the behavior of the optimization algorithm. At the same time, batch normalization has a slight regularizing effect on the network, which depends on the minibatch size, i.e., smaller minibatches introduce more randomness in the normalization, and thus adds more regularization. See e.g., [6, 82] for further details.

5.3 OTHER NEURAL NETWORK ARCHITECTURES

There is a wide variety of different NN architectures targeting different applications, and more are arising every month, but some base classes need to have a brief mentioning here, namely convolutional neural networks (CNNs), recurrent neural networks (RNNs), autoencoders, and generative adversarial networks (GANs).

CNNs can be applied to any domain that has a regular grid-like topology, e.g., images. They take their name from the convolution operation, which is used in place of the matrix multiplication in at least one of the layers of a traditional NN, e.g., the MLP. Thus, inputs passed to a convolutional layer are convolved with a kernel function, which usually has a dimensionality that is much smaller than the input, to produce a so-called feature map. Due to this inherent, sparse connectivity, and parameter sharing, CNNs can scale well to large domains. Furthermore, they generalize well, as they provide a representation that is invariant to spatial translations of the input, which is not the case for traditional NNs like MLPs, and they have fewer parameters to learn due to their sparse representation. See e.g., [82] for further details.

RNNs are a class of NNs specifically designed for sequence modeling. Note that whenever the sequence is defined on a regular grid, CNNs can also be used for this task. Just like CNNs scale well to grid-like typologies, RNNs can scale to long data sequences, and like both DBNs and CNNs, RNNs share parameters across different parts of the network. In case of RNNs, and DBNs, parameters sharing occurs between time slices, which makes it possible to apply and extend these models to data sequences of different length and generalize across them. See e.g., [82] for further details.

Autoencoders learn latent representations of the original input data, also called codings, in an unsupervised learning mode, and as the dimensionality of latent representation is typically lower than the inputs (undercomplete), autoencoders provide a means for dimensionality reduction. The architecture of autoencoders has two parts: An encoder that maps the inputs to the latent space and a decoder that maps the latent representations to the outputs, where the number neurons in the output layer must equal the input size, as the quality of the mapping is measured by the reconstruction of the inputs provided by the output layer. Some other applications of autoencoders are denoising, where the network learns to recover the noise-free inputs provided a noisy version, and generative modeling, where new data similar to the training data are generated. See e.g., [82, 92] for further details.

GANs are generative models that learn a dense representation of the data in a unsupervised learning mode, like autoencoders. The architecture is composed of two NNs: A generator network that generates new data, which are similar to the training data, and a discriminator network that discriminates simulated (fake) data from real data. During training the two networks com-

pete in a zero-sum game in which the generator tries to fool the discriminator. This is referred to as adversarial training, and it is considered one of the most important, recent ideas in deep learning [92]. See also [82] for further details.

5.4 BAYESIAN NEURAL NETWORKS

Section 5.1 shows that using weight decay regularization on the cost function, we arrive at a MAP estimate of the weights. However, if we adhere to a more complete Bayesian treatment, we need to marginalize over the parameters in order to make predictions. Bayesian neural networks (BNNs) are neural networks trained using Bayesian inference, where stochastic elements in terms of activation functions and/or weights are introduced in the network. Thus, BNNs be may considered as a special case of ensemble learning, see e.g., Sec. 7.1 and Appendix B.

For many models, it is infeasible to evaluate the posterior distribution or to compute expectations with respect to it, and thus we need to resort to an approximation method, e.g., MCMC or variational inference. As sampling methods can be computationally demanding, especially for high dimensional problems, we will focus on how to approximate the posterior distribution using variational inference, also known as variational Bayes [6]. Note that Appendix A also considers variational inference.

Our starting point is the posterior distribution for the weights, i.e., Eq. 71, which is restated in the general form below

$$p(\boldsymbol{w}|\mathcal{D}) = \frac{p(\boldsymbol{w})p(\mathcal{D}|\boldsymbol{w})}{p(\mathcal{D})}. \tag{83}$$

In this setting, variational inference provides a locally optimal, exact analytical solution to an approximation of the posterior distribution, through minimization of the Kullback-Leibler (KL) divergence between a variational distribution $q(\boldsymbol{w}|\boldsymbol{\nu})$ and the posterior distribution $p(\boldsymbol{w}|\mathcal{D})$, i.e.,

$$\begin{aligned}\mathbb{KL}[q(\boldsymbol{w}|\boldsymbol{\nu}) \parallel p(\boldsymbol{w}|\mathcal{D})] =& \mathbb{E}_q\left[\log \frac{q(\boldsymbol{w}|\boldsymbol{\nu})p(\mathcal{D})}{p(\boldsymbol{w})p(\mathcal{D}|\boldsymbol{w})}\right] \\ =& \mathbb{E}_q\left[\log q(\boldsymbol{w}|\boldsymbol{\nu}) + \log p(\mathcal{D}) - \log p(\boldsymbol{w}) - \log p(\mathcal{D}|\boldsymbol{w})\right] \\ =& \mathbb{KL}[q(\boldsymbol{w}|\boldsymbol{\nu}) \parallel p(\boldsymbol{w})] - \mathbb{E}_q[\log p(\mathcal{D}|\boldsymbol{w})] + \log p(\mathcal{D}).\end{aligned} \tag{84}$$

Since $\log p(\mathcal{D})$ is a normalizing constant with respect to $q(\boldsymbol{w}|\boldsymbol{\nu})$, we can define the following cost function for the variational parameters $\boldsymbol{\nu}$, i.e.,

$$L(\boldsymbol{\nu}) = \mathbb{KL}[q(\boldsymbol{w}|\boldsymbol{\nu}) \parallel p(\boldsymbol{w})] - \mathbb{E}_q[\log p(\mathcal{D}|\boldsymbol{w})]. \tag{85}$$

Eq. 85 is sometimes referred to as the variational free energy, and its negation $\mathrm{L}(\boldsymbol{\nu})$ as the energy functional or evidence lower bound (ELBO) [4, 91]. The

last name emphasizes that it is a lower bound on the log model evidence, i.e., from Eq. 84 we have that

$$\mathbb{KL}[q(\boldsymbol{w}|\boldsymbol{\nu}) \parallel p(\boldsymbol{w}|\mathcal{D})] = -\mathrm{L}(\boldsymbol{\nu}) + \log p(\mathcal{D}) \tag{86}$$

or equivalently

$$\log p(\mathcal{D}) = \mathbb{KL}[q(\boldsymbol{w}|\boldsymbol{\nu}) \parallel p(\boldsymbol{w}|\mathcal{D})] + \mathrm{L}(\boldsymbol{\nu}), \tag{87}$$

and since $\mathbb{KL}[q \parallel p] \geq 0$, with equality only if $q(\boldsymbol{w}|\boldsymbol{\nu}) = p(\boldsymbol{w}|\mathcal{D})$, it follows that $\mathrm{L}(\boldsymbol{\nu}) \leq \log p(\mathcal{D})$ [6].

In practice, we may e.g., utilize an approach called Bayes by backprop [91], which uses unbiased estimates of the gradients of the cost function. For this case, Eq. 85 is approximated as

$$L(\boldsymbol{\nu}) \approx \sum_{t=1}^{T} \log q(\boldsymbol{w}^{(t)}|\boldsymbol{\nu}) - \log p(\boldsymbol{w}^{(t)}) - \log p(\mathcal{D}|\boldsymbol{w}^{(t)}), \tag{88}$$

where $\boldsymbol{w}^{(t)}$ is a Monte Carlo sample drawn from the variational distribution $q(\boldsymbol{w}|\boldsymbol{\nu})$. Note that this procedure does not assume a specific prior family [91].

If we further consider a Gaussian mean field approximation, we can use the local reparameterization trick [99] to translate uncertainty about global parameters into local noise, i.e.,

$$\boldsymbol{w} = \boldsymbol{\mu}_w + \boldsymbol{\sigma}_w \odot \boldsymbol{\epsilon}, \text{ with } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{89}$$

where $\odot$ is element-wise multiplication. This means that instead of parameterizing the neural network with weights $\boldsymbol{w}$ directly, we parameterize it with parameter vectors $\boldsymbol{\mu}_w$ and $\boldsymbol{\sigma}_w$, which thus leads to a doubling of the original number of parameters. In return, though, we get an infinite ensemble representation instead of a single network model, where each network of the ensemble has its weights drawn from a shared, learnt probability distribution [91].

A convenient result of this parameterization is that the only additional thing we need to learn a variational model compared to a conventional NNs model, as discussed in Sec. 5.2, is an updating rule for $\boldsymbol{\nu} = \{\boldsymbol{\mu}_w, \boldsymbol{\sigma}_w\}$ based on the weight gradients provided by conventional back-propagation, which in this case simply amounts to scaling and shifting the weight gradients. Provided this updating rule, we can use the learning scheme presented in Sec. 5.2 to calculate the weight gradients with respect to Eq. 88, and conduct the parameter updates [91]. For a general review on BNNs and their applications, the interested reader is referred to e.g., Jospin et al. [100].

6 SYSTEMS ANALYSIS

The analysis of system characteristics is key in understanding the behavior of systems, and it facilitates a comprehensive study of system constituents, their relationship, and associated uncertainties. Section 6.1 introduces the two pillars of systems analysis, namely (i) uncertainty analysis and (ii) sensitivity analysis, which are also commonly framed as elements of uncertainty quantification [101]. After introducing uncertainty analysis, which is also a subject of the forgoing sections on system representations, along with some key references, we proceed to discuss sensitivity analysis, which is the subject of Secs. 6.2 and 6.3. In Sec. 6.2 we consider (global) variance-based sensitivity analysis in case of independent and dependent inputs, respectively; and in Sec. 6.3 we discuss regionalized sensitivity analysis and show how cluster analysis may be used for this purpose. Note that tutorials supporting the material covered in this section are available at the GitHub repository.

6.1 INTRODUCTION TO SYSTEMS ANALYSIS

The analysis of systems naturally involves a consideration of how uncertainty in the inputs propagates through a system to uncertainty in the outputs. This is referred to as uncertainty analysis (UA). Sensitivity analysis (SA) propagates uncertainty in the opposite direction, i.e., from outputs to inputs, while focusing on how the uncertainty in system outputs can be appointed to different sources of uncertainty in the inputs. Thus, ideally, a UA study should precede a SA study, as uncertainty can only be appointed once it is estimated. In a quantitative analysis of systems uncertainty, we can only vary a subset of the model assumptions, where an assumption refers to any choice of e.g., type and structure of model, parameters, resolution, and calibration data. This subset is referred to as the input factors, and the results of the model for any combination of the input factors is, as always, referred to as the model output. Thus, when performing a UA and SA, we should keep in mind that the uncertainty in the assumptions that are not part of the set of input factors are not explored [102, 103].

The quantification of uncertainties related to system representations allows us to bridge our computational modeling universe and the actual, physical systems represented by accounting for the different sources of variability in our models, which include (i) uncertainties in input factors, such as parameters, initial conditions and boundary conditions; (ii) discrepancies between the model and the true system; (iii) uncertainties due to a limited computational budget, e.g., limited number of simulations; and (iiii) solution and coding errors [101]. In this regard, a UA may be undertaken using e.g., a perturbation-based method, where a lower-order Taylor expansion is used to derive approximate expressions for the statistical moments of the model out-

puts, or a sampling-based approach, where a number of joint samples of the input factors are drawn, and the model output evaluated for each joint sample. The inserted reader is referred to e.g., [101] for a comprehensive coverage of UA techniques and the general field of uncertainly quantification.

SA techniques generally fall in two categories: local and global techniques. Local techniques typically rely on the first partial derivatives of the response function with respect to the input factors at a given design point. Two important consequences of this definition of sensitivity are (i) if the functional relationship f between inputs and outputs is nonlinear, then the derivatives will change with the design point, and (ii) if input interactions are present in f, then the partial derivative for input factor X_i will change depending on the remaining inputs $\boldsymbol{X}_{\sim i}$. Thus, the first partial derivatives are only valid measures of sensitivity when the model is linear, in which case the partial derivatives will remain constant over the range of $\boldsymbol{X}$ [102].

Because of the limitations of local sensitivity techniques, Saltelli et al. [102] advocate the use of global SA techniques, when f is nonlinear and/or inputs interact. One approach to perform global SA is the analysis of variance (ANOVA), which informs the analyst about the global contribution of input factors, in terms of variance, to the model outputs, including the effect of interactions among inputs [104]. Some other global approaches to SA include e.g., the elementary effect approach [105], global derivative-based measures [106], and moment-independent methods [107]. See e.g., [101] for further details.

Following Saltelli et al. [103], a non-exhaustive list of possible settings for a global SA, and corresponding objectives, is given below

Factor Prioritization (FP): Which factor (or group of factors) produces the greatest reduction in output variance, if set to its true value?

Factor Fixing (FF): Which factors (or group of factors) provides no significant contribution to the output variance, if set to an arbitrary value in its range?

Variance cutting (VC): How can we bring the output variance below a given threshold by acting on the smallest possible number of factors?

Factor mapping (FM): Which factors are driving a specific target behavior of the system?

The FP setting allows us to rank the factors according to their impact on the output variance. The FF setting provides an assessment of which factors could be fixed, within its range, without significantly influencing the output variance. The VC setting allows us to focus our effort to bring the output variance below a certain threshold. The FM setting allows us to identity the

most influential factors in driving the model output into a specific range of output space.

The FP, FF, and VC settings fall in the category of variance-based SA, whereas the FM setting relies on Monte Carlo filtering. In this study, we will focus on two avenues for system analysis using global SA, namely variance-based SA for factor prioritization and factor fixing, see Sec. 6.2, and regionalized SA, including cluster analysis, for factor mapping, see Sec. 6.3.

6.2 VARIANCE-BASED SENSITIVITY ANALYSIS

Parts of this section appear in Paper B.

In this section, we outline how variance-based sensitivity analysis can be performed via the functional ANOVA decomposition for the case of independent inputs and via the functional ANCOVA decomposition for the case of dependent inputs, respectively.

ANOVA DECOMPOSITION

In variance-based SA, we assess how the variance of the output depends on the uncertain input factors, or variables, by considering how the output variance can be decomposed. The basis is a high-dimensional model representation (HDMR) [108], where a response function $Y = f(X_1, X_2, ..., X_M)$ is decomposed into a set of functions of increasing dimensionality, see Eq. 90.

$$f = f_0 + \sum_i f_i + \sum_i \sum_{j>i} f_{ij} + ... + f_{12...M}, \tag{90}$$

where the individual terms are square integrable over the domain of existence, and only a function of the factors in their index, thus $f_i = f(X_i)$, $f_{ij} = f(X_i, X_j)$ and so on.[9] Given that each term in Eq. 90 is defined to have zero mean, e.g., $\int f_i(x_i)dx_i = 0$ and $\int f_i(x_i)f_j(x_j)dx_i dx_j = 0$, the terms are orthogonal in pairs, and the individual terms can be uniquely calculated by use of the conditional expectation of the model output, e.g.,

$$f_0 = \mathbb{E}[Y] \tag{91a}$$

$$f_i = \mathbb{E}[Y|X_i] - f_0 \tag{91b}$$

$$f_{ij} = \mathbb{E}[Y|X_i, X_j] - f_i - f_j - f_0. \tag{91c}$$

The so-called first-order sensitivity index corresponds to the variance of the univariate terms $\mathbb{V}_i = \mathbb{V}[f_i] = \mathbb{V}[\mathbb{E}[Y|X_i]]$ scaled by the unconditional output variance $\mathbb{V}[Y]$, i.e.,

$$S_i = \frac{\mathbb{V}_{X_i}[\mathbb{E}_{\mathbf{X}_{\sim i}}[Y|X_i]]}{\mathbb{V}[Y]}, \tag{92}$$

[9] Note that Eq. 90 is not a series expansion, as it has a finite number of terms [103].

where $\boldsymbol{X}_{\sim i}$ denotes all variables except X_i. The index Eq. 92 represents the main effect contribution from factor i to the output variance [103] (Paper B).

Two factors are said to interact when their effect on Y cannot be expressed as a sum of single effects. For independent input factors, the output variance decomposes as

$$\mathbb{V}[Y] = \sum_i \mathbb{V}_i + \sum_i \sum_{j>i} \mathbb{V}_{ij} + ... + \mathbb{V}_{12...M}, \tag{93}$$

where the terms $\mathbb{V}_{ij}$, $\mathbb{V}_{ijk}$ et cetera correspond to interaction terms. Equation 93 is commonly referred to as the analysis of variance (ANOVA) decomposition. Dividing both sides of Eq. 93 by the output variance, the following relationship appears

$$1 = \sum_i S_i + \sum_i \sum_{j>i} S_{ij} + ... + S_{12...M}. \tag{94}$$

Based on Eq. 94, a set of properties can now be derived for the first-order sensitivity indices [103], see Tab. 1.

Table 1: Properties of first-order sensitivity indices (Paper B).

$\sum_i S_i \leq 1$	Always
$\sum_i S_i = 1$	Additive models
$1 - \sum_i S_i$	Indicates presence of interactions

The so-called total effect index represents the joint effect of all contributions related to a factor. That is, the first-order effect of a factor and higher-order effect due to its interactions with the remaining factor. As an example, the total effect index of X_1 in a three factor model is

$$S_{T_1} = S_1 + S_{12} + S_{13} + S_{123}.$$

The terms in Eq. 94 could in principle be used to construct the total effect indices, as in the example above, but then $2^k - 1$ terms must be calculated. That is, this procedure suffers under the curse of dimensionality. Instead, we explore the law of total variance, i.e.,

$$\mathbb{V}[Y] = \mathbb{V}_{X_i}[\mathbb{E}_{\boldsymbol{X}_{\sim i}}[Y|X_i]] + \mathbb{E}_{X_i}[\mathbb{V}_{\boldsymbol{X}_{\sim i}}[Y|X_i]], \tag{95}$$

or equivalently

$$\mathbb{V}[Y] = \mathbb{V}_{\boldsymbol{X}_{\sim i}}[\mathbb{E}_{X_i}[Y|\boldsymbol{X}_{\sim i}]] + \mathbb{E}_{\boldsymbol{X}_{\sim i}}[\mathbb{V}_{X_i}[Y|\boldsymbol{X}_{\sim i}]]. \tag{96}$$

In both factorizations, the first term represents the variance due the conditioning set, and the second term represents the residual variance, i.e., the

variance due to variables not in the conditioning set. Note that the first-order sensitivity index corresponds to the first term in Eq. 95 divided by the output variance, see Eq. 92. By use of Eq. 96, the total effect index of variable i may be represented by the residual variance divided by the output variance, i.e.,

$$S_{T_i} = \frac{\mathbb{E}_{\boldsymbol{X}_{\sim i}}[\mathbb{V}_{X_i}[Y|\boldsymbol{X}_{\sim i}]]}{\mathbb{V}[Y]} = 1 - \frac{\mathbb{V}_{\boldsymbol{X}_{\sim i}}[\mathbb{E}_{X_i}[Y|\boldsymbol{X}_{\sim i}]]}{\mathbb{V}[Y]}. \tag{97}$$

Equation Eq. 97 provides a more efficient way of calculating the total effect index than the brute force formulation based on Eq. 94 [103] (Paper B).

In a variance-based sensitivity assessment, the set of all S_i and S_{T_i} indices provides a reasonable good description of the model sensitivity at a computationally cost that is tractable for most models. Thus, variance-based main effects are suitable in a factor prioritization setting, while the total effects address a factor fixing setting. In case evaluation of S_i and S_{T_i} is computationally intractable, there exists a set of proxies for these indices. The elementary effect measure μ_i^* can, for instance, be used as a proxy for S_{T_i} [103] (Paper B).

In practical applications, it is often convenient to build a surrogate model to approximate outputs from the underlying computational model, as such models tend to be computationally expensive to run, and Monte Carlo estimation of the Sobol indices requires a large amount of samples. One surrogate modeling scheme that lends itself directly to SA applications is polynomial chaos expansion (PCE) [109], as the model representation immediately provides the quantities needed to compute the Sobol indices [110].

PCE is a spectral method that represents f as an infinite sum of multivariate orthonormal polynomials. In practice, this representation is truncated, which leads to a finite approximation of the form

$$f(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}), \tag{98}$$

where $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_k)$ is a multi-index or tuple for which $\alpha_i \in \mathbb{N}$, and $\mathcal{A}$ is a truncation scheme, e.g., $\mathcal{A} = \{(0,0), ..., (1,2), ...(0,3)\}$ for degree 3 in two variables. The polynomial degree is typically chosen to be 3 to 5, and/or a more advanced truncation scheme is applied, to cope with the curse of dimensionality. Furthermore, $\{y_{\boldsymbol{\alpha}}\}$ is the set of parameters, which may be interpreted as the coordinates of Y in the basis, and $\{\Psi_{\boldsymbol{\alpha}}\}$ is the set of multivariate polynomials defined as

$$\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \prod_{i=1}^{M} \psi_{\alpha_i}^{(i)}(x_i), \tag{99}$$

where $\{\psi_{\alpha_i}^{(i)}\}$ is the set of orthonormal, univariate polynomials used as the basis for the expansion. For example, if $X_i \sim \mathcal{N}(0,1)$ the corresponding polynomial family is Hermite polynomials [111, 112].

For any subset $\boldsymbol{u} = \{i_1, ..., i_s\} \subseteq \{1, .., M\}$, the multivariate polynomials that depend only on $\boldsymbol{u}$ are

$$\mathcal{A}_{\boldsymbol{u}} = \{\boldsymbol{\alpha} \in \mathcal{A} : \alpha_i \neq 0 \quad \text{iff} \quad i \in \boldsymbol{u}\}. \tag{100}$$

Thus, we can rewrite Eq. 98 as

$$f(\boldsymbol{X}) = y_0 + \sum_{\substack{\boldsymbol{u} \subseteq \{1,..,M\} \\ \boldsymbol{u} \neq \emptyset}} \left[\sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\boldsymbol{u}}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}_{\boldsymbol{u}}) \right], \tag{101}$$

where $\boldsymbol{X}_{\boldsymbol{u}}$ is the subset of $\boldsymbol{X}$ with index $i \in \boldsymbol{u}$. Now, due to orthogonality of the PCE basis, we can decompose the output variance as

$$\mathbb{V}[\boldsymbol{Y}] = \mathbb{V}\left[\sum_{\substack{\boldsymbol{u} \subseteq \{1,..,M\} \\ \boldsymbol{u} \neq \emptyset}} \left[\sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\boldsymbol{u}}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}_{\boldsymbol{u}}) \right] \right] = \sum_{\substack{\boldsymbol{u} \subseteq \{1,..,M\} \\ \boldsymbol{u} \neq \emptyset}} \mathbb{V}\left[\sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\boldsymbol{u}}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}_{\boldsymbol{u}}) \right], \tag{102}$$

where the partial variance terms $\mathbb{V}[f_{\boldsymbol{u}}]$ reduces to

$$\mathbb{V}(f_{\boldsymbol{u}}) = \mathbb{V}\left[\sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\boldsymbol{u}}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}_{\boldsymbol{u}}) \right] = \sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\boldsymbol{u}}} y_{\boldsymbol{\alpha}}^2. \tag{103}$$

Based on Eq. 103, the Sobol indices may be defined by

$$S_{\boldsymbol{u}} = \frac{1}{\mathbb{V}[Y]} \sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\boldsymbol{u}}} y_{\boldsymbol{\alpha}}^2, \tag{104}$$

with the corresponding indices given in Tab. 2 [111]. In this study, we only present a concise description of PCEs that evolves around SA. The interested reader may refer to e.g., [111] for more general details on PCEs.

Table 2: First-order and total sensitivity index based on PCE surrogate model.

S_i	$\mathcal{A}_{\boldsymbol{u}} = \mathcal{A}_i = \{\boldsymbol{\alpha} \in \mathcal{A} : \alpha_i > 0, \alpha_{i \neq j} = 0\}$
S_i^T	$\mathcal{A}_{\boldsymbol{u}} = \mathcal{A}_i^T = \{\boldsymbol{\alpha} \in \mathcal{A} : \alpha_i > 0\}$

ANCOVA DECOMPOSITION

The basis for variance decomposition in the case of correlated inputs is again the HDMR (Eq. 90), but for dependent input variables, it is not possible to derive a unique decomposition in terms of orthogonal summands of increasing order [111, 113]. The unconditional variance of the model output may now be written as

$$\mathbb{V}[Y] = \mathbb{E}\left[(Y - \mathbb{E}[Y])^2 \right] \tag{105a}$$

$$\mathbb{V}[Y] = \mathbb{E}\left[(Y - f_0)\Big(\sum_{\boldsymbol{u}\subseteq\{1,..,k\}} f_{\boldsymbol{u}}\Big)\right] \tag{105b}$$

$$= \mathbb{C}\left[Y, \sum_{\boldsymbol{u}\subseteq\{1,..,M\}} f_{\boldsymbol{u}}\right] \tag{105c}$$

$$= \sum_{\boldsymbol{u}\subseteq\{1,..,M\}} \mathbb{C}\left[Y, f_{\boldsymbol{u}}\right] \tag{105d}$$

$$= \sum_{\boldsymbol{u}\subseteq\{1,..,M\}} \left[\mathbb{V}[f_{\boldsymbol{u}}] + \mathbb{C}\left[f_{\boldsymbol{u}}, \sum_{\boldsymbol{v}\subseteq\{1,..,M\}, \boldsymbol{v}\cap\boldsymbol{u}=\emptyset} f_{\boldsymbol{v}}\right]\right] \tag{105e}$$

where each function $\{f_{\boldsymbol{u}} | \boldsymbol{u} \subseteq \{1,..,M\}\}$ represents the combined contribution of the variables $\boldsymbol{X}_{\boldsymbol{u}}$ to Y. Equation 105b holds because Y contains its functional decomposition, Eq. 105c reframes the variance of Y as the covariance of Y with itself, Eq. 105d explores the properties of covariance, and Eq. 105e splits the covariance terms into a variance and a covariance term. Equation 105e is commonly denoted the analysis of covariance (ANCOVA) decomposition. Note that when the inputs are independent, the functions $f_{\boldsymbol{u}}$ and $f_{\boldsymbol{v}}$ are orthogonal. Thus, the covariance terms of Eq. 105e evaluate to zero, and only the variance terms are left in the equation. Under these conditions, the ANCOVA decomposition is equivalent to the ANOVA decomposition, thus ANOVA is a particular case (independent inputs) of ANCOVA [114, 115] (Paper B).

The ANCOVA decomposition in Eq. 105e facilitates a separation of the uncorrelated and correlated effects in the following sensitivity indices

$$S_{\boldsymbol{u}} = \frac{\mathbb{C}[Y, f_{\boldsymbol{u}}]}{\mathbb{V}[Y]} \tag{106a}$$

$$S_{\boldsymbol{u}}^{U} = \frac{\mathbb{V}[f_{\boldsymbol{u}}]}{\mathbb{V}[Y]} \tag{106b}$$

$$S_{\boldsymbol{u}}^{C} = \frac{\mathbb{C}\left[f_{\boldsymbol{u}}, \sum_{\boldsymbol{v}\subseteq\{1,..,M\}, \boldsymbol{v}\cap\boldsymbol{u}=\emptyset} f_{\boldsymbol{v}}\right]}{\mathbb{V}[Y]}. \tag{106c}$$

From this definition of indices, it is seen that $S_{\boldsymbol{u}}$ represents the total contribution to output variance due to $\boldsymbol{X}_{\boldsymbol{u}}$, $S_{\boldsymbol{u}}^{U}$ represents the uncorrelated, or structural, share of output variance due to $\boldsymbol{X}_{\boldsymbol{u}}$, and $S_{\boldsymbol{u}}^{C}$ represents the correlated share of output variance due to $\boldsymbol{X}_{\boldsymbol{u}}$, i.e., the contribution due to correlations between $\boldsymbol{X}_{\boldsymbol{u}}$ and the remaining inputs. Moreover, the relationship between the indices is

$$S_{\boldsymbol{u}} = S_{\boldsymbol{u}}^{U} + S_{\boldsymbol{u}}^{C}. \tag{107}$$

As a result of this definition, $S_{\boldsymbol{u}}^{U}$ is always positive, the sign of $S_{\boldsymbol{u}}^{C}$ depends on the nature of the correlation between $\boldsymbol{X}_{\boldsymbol{u}}$ and the remaining inputs, and thus

the sign of $S_{\boldsymbol{u}}$ depends on which of the structural contribution $S_{\boldsymbol{u}}^U$ and correlative contribution $S_{\boldsymbol{u}}^C$ is largest. In this context, $S_{\boldsymbol{u}}^C$ should be understood as a corrective term that indicates whether the total contribution is overestimated or underestimated because of the correlation between the inputs. If $|S_{\boldsymbol{u}}^C|$ is small the correlation has a weak influence of the contribution of $\boldsymbol{X}_{\boldsymbol{u}}$, and if it is large the correlation has a strong influence of the contribution of $\boldsymbol{X}_{\boldsymbol{u}}$ [114, 115] (Paper B).

Saltelli ét al. [116] argue that the condition $\mathbb{E}[\mathbb{V}[Y|\mathbf{X}_{\sim i}]] = 0$ is a necessary and sufficient condition to deem X_i non-influential, under any model or correlation/dependency structure among the inputs. Moreover, if we add the first order index and the higher-order indices of a variable together, we arrive at an index S_i^T that is consistent with the Sobol total effect index for the case of independent inputs [115]. Note that in case of correlation among the inputs, the total effect terms can be smaller than the first-order terms (Paper B). There is, however, confusion about the distinction between interaction and correlation effects, when higher-order indices are calculated using Eqs. 106a–106c. That is, if X_i and X_j are correlated, the term f_{ij} still appears in the uncorrelative index (Eq. 106b). Furthermore, the correlative index (Eq. 106c) may detect a correlation between $X_{k\in\boldsymbol{u},k\neq i}$ and $X_{l\in\boldsymbol{v},k\neq i}$, although X_i is neither correlated with X_k nor X_l, see [115, 116] for further details.

We also consider how PCE can be used to provide estimates for the sensitivity induces, when the inputs are dependent. In this case, the recommended procedure [113, 115] follows two distinct steps. First, we build a PCE surrogate as if the input vector $\boldsymbol{X}$ has independent components (independent copula). Second, we evaluate the variances and covariances by simulating realizations of $\boldsymbol{X}$, which accommodate for the dependency structure (dependent copula). The sensitivity indices may then be estimated as

$$S_{\boldsymbol{u}} = \frac{\sum_{t=1}^{T}(f_{\mathcal{A}}(\boldsymbol{x}^{(t)}) - \bar{y}_{\mathcal{A}})(f_{\boldsymbol{u}}(\boldsymbol{x}_{\boldsymbol{u}}^{(t)}) - \bar{y}_{\boldsymbol{u}})}{\sum_{t=1}^{T}(f_{\mathcal{A}}(\boldsymbol{x}^{(t)}) - \bar{y}_{\mathcal{A}})^2} \tag{108a}$$

$$S_{\boldsymbol{u}}^U = \frac{\sum_{t=1}^{T}(f_{\boldsymbol{u}}(\boldsymbol{x}_{\boldsymbol{u}}^{(t)}) - \bar{y}_{\boldsymbol{u}})^2}{\sum_{t=1}^{T}(f_{\mathcal{A}}(\boldsymbol{x}^{(t)}) - \bar{y}_{\mathcal{A}})^2} \tag{108b}$$

$$S_{\boldsymbol{u}}^C = S_{\boldsymbol{u}} - S_{\boldsymbol{u}}^U, \tag{108c}$$

with $\bar{y}_{\mathcal{A}} = \frac{1}{T}\sum_{t=1}^{T} f_{\mathcal{A}}(\boldsymbol{x}^{(t)})$ and $\bar{y}_{\boldsymbol{u}} = \frac{1}{T}\sum_{t=1}^{T} f_{\boldsymbol{u}}(\boldsymbol{x}_{\boldsymbol{u}}^{(t)})$, see e.g, [113, 115] for further details.

Finally, note that other approaches for generalizing the Sobol indices are found in the literature, see e.g., [117, 118]. These approaches generally provide different results when the inputs are correlated, but they are consistent with the Sobol indices when the inputs are independent. Thus, several generalizations of the Sobol indices for models with dependent inputs are available in the literature, but it remains an active research topic [115].

6.3 REGIONALIZED SENSITIVITY ANALYSIS

Parts of this section appear in Paper B.

Regionalized sensitivity analysis (RSA) is a Monte Carlo (MC) filtering approach, where Monte Carlo simulations are filtered, based on the model output realizations, into a "behavioral" ($\mathcal{B}$) and a "non-behavioral" ($\bar{\mathcal{B}}$) set. The behavioral samples correspond to output realizations exhibiting a certain target behavior, which in this study are realizations leading to (partial) system failure, see Paper B. Based on this partitioning, RSA aims to identify which factors are most important in driving output realizations into $\mathcal{B}$ or $\bar{\mathcal{B}}$. In practice, this is typically attained by comparing the subsets $\{X_i|\mathcal{B}\}$ and $\{X_i|\bar{\mathcal{B}}\}$ for all input factors, under the intuition that if the two subset are dissimilar to one another, then the factor is influential [103].

STATISTICAL TESTING

The assessment of similarity between the behavioral and non-behavioral set may be conducted by means of hypothesis testing using e.g., the Smirnov two-sample test (two-sided) [103]. For this case, the test statistic is defined as

$$\mathrm{d}_{\mathcal{B},\bar{\mathcal{B}}}(X_i) = \sup ||\hat{F}(X_i|\mathcal{B}) - \hat{F}(X_i|\bar{\mathcal{B}})||, \tag{109}$$

where $\hat{F}$ is the empirical CDF. We then assess at which significance level α the null hypothesis, i.e., $F(X_i|\mathcal{B}) = F(X_i|\bar{\mathcal{B}})$, is rejected. The larger $\mathrm{d}_{\mathcal{B},\bar{\mathcal{B}}}(X_i)$, or equivalently the smaller α, the more important the parameter is in driving the model behavior. The Smirnov test provides a means for assessing whether the factor under analysis is important, but it does not provide a necessary condition for deeming a factor unimportant, i.e., its non-significance does not ensure that a factor is non-influential. Furthermore, low success rates may occur for rare model behaviors, which will lead to lack of statistical power [103].

CLUSTER ANALYSIS

In some cases, we may be interested in analyzing whether all behavioral samples in a MC experiment corresponds to the same latent grouping, or whether multiple latent groupings exist in the data. For this purpose, cluster analysis may be employed. Furthermore, a comparison of the cluster representation of $\{X_i|\mathcal{B}\}$ and $\{X_i|\bar{\mathcal{B}}\}$ may reveal distinct patterns in the two subsets.

Cluster analysis is a general class of techniques used to classify items in a database into relative groups called clusters, when no information about the cluster memberships is available a-priori. Thus, the basic assumption in cluster analysis is that the considered data set $\mathcal{D}$ is sampled from a finite set of

distinct base models, and the target of the analysis is to infer the most likely generating base model for each realization, i.e., the latent cluster assignment. In this regard, we assume that the data set $\mathcal{D} = \{\boldsymbol{x}[n]\}_{n=1}^{N}$ consists of N i.i.d. observations of a random vector $\boldsymbol{X} \in \mathbb{R}^M$. Model-based clustering is commonly used as a basis for cluster analysis, as it provides a framework for choosing the relevant number of clusters in the data, as well as assessing the resulting partitioning of the data, see e.g., [119, 120] (Paper B).

In a model-based setting, if it is further assumed that the base model for each of the clusters is Gaussian, the joint distribution can be represented as a Gaussian mixture model (GMM) of the form

$$p(\boldsymbol{x}|\boldsymbol{\Theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{110}$$

where $\boldsymbol{\Theta}$ represent the collection of all model parameters, $\boldsymbol{\mu}_k$ is the mean vector of cluster k, $\boldsymbol{\Sigma}_k$ is the covariance matrix of cluster k, and π_k is the mixing weight or probability of cluster k, such that $\sum_k \pi_k = 1$ with $0 \leq \pi_k \leq 1$.

The generative model for the data is shown in Fig. 12, where $\boldsymbol{z}[n] \in \{0, 1\}$ is a binary random variable with a 1-of-K encoding in which a particular element z_k is equal to 1 and all other elements are equal to zero. This variable represent the latent cluster assignment for data item n, with a marginal distribution specified by the mixing weights, such that $p(z_k = 1) = \pi_k$ and $p(\boldsymbol{z}) = \prod_k \pi_k^{z_k}$ [6]. Thus, the corresponding joint distribution $p(\boldsymbol{x}, \boldsymbol{z})$ is defined in terms of the marginal distribution $p(\boldsymbol{z})$ and a conditional distribution $p(\boldsymbol{x}|\boldsymbol{z}) = \prod_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}$. The marginal distribution for $p(\boldsymbol{x})$ (Eq. 110) thus appears by marginalizing out $\boldsymbol{z}$ in $p(\boldsymbol{x}, \boldsymbol{z})$, i.e., $p(\boldsymbol{x}) = \sum_z p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$ (Paper B).

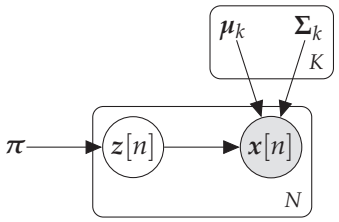


Figure 12: Meta-network of a Gaussian mixture model (Paper B).

The log-likelihood of the data under this model is

$$\log p(\mathcal{D}|\boldsymbol{\Theta}) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}[n]|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right). \tag{111}$$

No closed form solution can be derived for the maximization of this expression with respect to the parameters, due to the summation over k that appears

inside the logarithm, and thus it is necessary to resort to an iterative scheme, e.g., expectation maximization (EM), to estimate the parameters of the distribution, see e.g., [6] (Paper B).

Expectation maximization As described in Sec. 3.4, EM is a method for finding point estimates of distribution parameters, i.e., maximum likelihood (MLE) or maximum a-posteriori (MAP), for models with latent variables. In Sec. 3.4, the latent variables are missing data in the data set, and in this application they are latent cluster assignments, i.e., $z[n]$ in Fig. 12.

First, we consider the solution conditions for the means, thus setting the derivatives of Eq. 111 with respect to the mean components $\boldsymbol{\mu}_k$ to zero, we obtain the following equality

$$0 = \sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(\boldsymbol{x}[n]|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\boldsymbol{x}[n]|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}}_{\gamma(z_k[n])} \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}[n] - \boldsymbol{\mu}_k), \tag{112}$$

where we can view π_k as the prior probability of $z_k = 1$ and the quantity $\gamma(z_k)$ as the corresponding posterior probability after observing $\boldsymbol{x}$. The $\gamma(z_k)$ components are also called responsibilities, as they express the responsibility of the individual components in explaining an observation $\boldsymbol{x}$ [6]. Now, multiplying by $\boldsymbol{\Sigma_k}$ and rearranging, we obtain an expression for $\boldsymbol{\mu}_k$, i.e.,

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_k[n]) \boldsymbol{x}[n], \tag{113}$$

where $N_k = \sum_n \gamma(z_k[n])$ may be interpreted as the effective number of points assigned to cluster k [6]. Next, we consider the solution conditions for the covariances, thus setting the derivatives of Eq. 111 with respect to the covariance components $\boldsymbol{\Sigma}_k$ to zero and rearranging, we obtain the following expression for $\boldsymbol{\Sigma_k}$, i.e.,

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_k[n])(\boldsymbol{x}[n] - \boldsymbol{\mu}_k)(\boldsymbol{x}[n] - \boldsymbol{\mu}_k)^T. \tag{114}$$

Finally, we need to maximize Eq. 111 with respect to the mixing coefficients π_k, accounting for the constraint $\sum_k \pi_k = 1$. This may be archived by augmenting Eq. 111 with a Lagrange multiplier component, and maximizing the augmented log-likelihood to arrive at the following expression for π_k, i.e.,

$$\pi_k = \frac{N_k}{N}. \tag{115}$$

That is, the mixing component of cluster k reflects the average responsibility of that cluster in explaining the training data [6].

The EM algorithm proceeds as follows: We first initialize the parameters in $\boldsymbol{\Theta}$ to some appropriate values; commonly the related K-means algorithm [6] is used for the initialization of the parameters. Second, we alternate between updating the expected responsibilities $\gamma(z_k[n])$ for each cluster (E-step), and updating the parameters in $\boldsymbol{\Theta} = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k\}_{k=1}^K$ (M-step). Pseudo-code for the EM algorithm is provided in Alg. 6.

Algorithm 6: Pseudo-code of the EM algorithm for GMMs.

Input: $\mathcal{D}$
Output: $\hat{\boldsymbol{\theta}}^{(t)}$
Initialization: $\hat{\boldsymbol{\theta}}^{(0)}$
for $t = 1, ...,$ *until convergence* **do**
  E-step:
  for $n = 1, ..., N$ **do**
    for $k = 1, ..., K$ **do**
      $\gamma(z_k[n]) \leftarrow \frac{\pi_k \mathcal{N}(\boldsymbol{x}[n]|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\boldsymbol{x}[n]|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}$
    end
  end
  M-step:
  for $k = 1, ..., K$ **do**
    $N_k^{(t)} \leftarrow \sum_{n=1}^N \gamma(z_k[n])$
    $\boldsymbol{\mu}_k^{(t)} \leftarrow \frac{1}{N_k} \sum_{n=1}^N \gamma(z_k[n]) \boldsymbol{x}[n]$
    $\boldsymbol{\Sigma}_k^{(t)} \leftarrow \frac{1}{N_k} \sum_{n=1}^N \gamma(z_k[n]) (\boldsymbol{x}[n] - \boldsymbol{\mu}_k^{(t)})(\boldsymbol{x}[n] - \boldsymbol{\mu}_k^{(t)})^T$
    $\pi_k^{(t)} \leftarrow \frac{N_k}{N}$
  end
end

The EM algorithm may also be derived from an alternative view point, where we express our state of knowledge of both observed and unobserved variables in Fig. 12, i.e., $\{\boldsymbol{x}[n], \boldsymbol{z}[n]\} = \{\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}}\}$, in a direct manner. The set $\{\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}}\}$ is commonly referred to as the "complete" data set, and the corresponding log-likelihood $\log p(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}}|\boldsymbol{\Theta})$ is referred to as the complete-data log-likelihood. As $\hat{\boldsymbol{Z}}$ is unobserved, our state of knowledge about $\hat{\boldsymbol{Z}}$ is expressed by the posterior $p(\hat{\boldsymbol{Z}}|\hat{\boldsymbol{X}}, \boldsymbol{\Theta})$. The EM algorithm then computes the expectation (E-step) of the complete-data log-likelihood under this posterior, and its subsequent maximization (M-step) with respect to the parameters. Thus, in the E-step, the posterior of the latent variables is computed based on the current parameter setting, i.e. $\hat{\boldsymbol{\theta}}^{(t-1)}$, which in turn is used to express the expectation of the complete-data log-likelihood for a general parameter

setting $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. This expectation may be written as

$$Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(t-1)}) = \sum_{\hat{\boldsymbol{Z}}} p(\hat{\boldsymbol{Z}}|\hat{\boldsymbol{X}}, \hat{\boldsymbol{\theta}}^{(t-1)}) \log p(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}}|\boldsymbol{\theta}), \tag{116}$$

and the subsequent M-step proceeds as

$$\hat{\boldsymbol{\theta}}^{(t)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(t-1)}). \tag{117}$$

Note that the logarithm in $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(t-1)})$ acts directly on the likelihood [6].

Now, we consider how to derive the different components involved in this formulation of the EM algorithm. First, using Bayes' theorem, we can express the posterior of the latent variables as

$$\begin{aligned} p(\hat{\boldsymbol{Z}}|\hat{\boldsymbol{X}}, \boldsymbol{\Theta}) &\propto p(\hat{\boldsymbol{X}}|\hat{\boldsymbol{Z}}, \boldsymbol{\Theta}) p(\hat{\boldsymbol{Z}}|\boldsymbol{\Theta}) \\ &\propto \prod_{n=1}^{N} \prod_{k=1}^{K} \left(\pi_k \mathcal{N}(\boldsymbol{x}[n]|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right)^{z_k[n]}. \end{aligned} \tag{118}$$

Note that under the posterior, the $\boldsymbol{z}[n]$ vectors are independent, as it appears in Fig. 12. The expected value of the latent component $z_k[n]$ under this posterior distribution can now be computed as

$$\begin{aligned} \mathbb{E}[z_k[n]] &= \frac{\sum_{\boldsymbol{z}[n]} z_k[n] \ \prod_{k'} \left[\pi_{k'} \mathcal{N}(\boldsymbol{x}[n]|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})\right]^{z_{k'}[n]}}{\sum_{\boldsymbol{z}[n]} \prod_{k'} \left[\pi_{k'} \mathcal{N}(\boldsymbol{x}[n]|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})\right]^{z_{k'}[n]}} \\ &= \frac{\pi_k \mathcal{N}(\boldsymbol{x}[n]|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\boldsymbol{x}[n]|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})} = \gamma(z_k[n]), \end{aligned} \tag{119}$$

whereby we retrieve the responsibilities $\{\gamma(z_k[n])\}$ in Eq. 112 [6]. The expected complete-data log-likelihood may thus be written as

$$\mathbb{E}_{\hat{\boldsymbol{Z}}} \left[\log p(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}}|\boldsymbol{\Theta})\right] = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_k[n]) \left[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}[n]|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right]. \tag{120}$$

For this case, the EM algorithm thus amounts to successively compute the responsibilities in Eq. 119 based on the current parameter setting (E-step) and optimizing Eq. 120 with respect to the parameters in $\boldsymbol{\Theta}$ (M-step), which leads to the expressions in Eqs. 113–115 as before [6].

Finally, we may link the second deviation of the EM algorithm to our discussion of variational inference in Sec. 5.4 (Eq. 87) by writing the log-likelihood as

$$\begin{aligned} \log p(\hat{\boldsymbol{X}}|\boldsymbol{\theta}) &= \mathrm{L}_q(\boldsymbol{\theta}) + \mathbb{KL}[q \parallel p] \\ &= \sum_{\hat{\boldsymbol{Z}}} q(\hat{\boldsymbol{Z}}) \log \frac{p(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}}|\boldsymbol{\theta})}{q(\hat{\boldsymbol{Z}})} - \sum_{\hat{\boldsymbol{Z}}} q(\hat{\boldsymbol{Z}}) \log \frac{p(\hat{\boldsymbol{Z}}|\hat{\boldsymbol{X}}, \boldsymbol{\theta})}{q(\hat{\boldsymbol{Z}})}, \end{aligned} \tag{121}$$

where $q(\hat{\boldsymbol{Z}})$ is some distribution defined over the latent variables, i.e., similar to the variational distribution of Sec. 5.4; $\mathbb{KL}[q \parallel p]$ is the Kullback-Leibler divergence between $q(\hat{\boldsymbol{Z}})$ and $p(\hat{\boldsymbol{Z}}|\hat{\boldsymbol{X}}, \boldsymbol{\theta})$; and $\mathrm{L}_q(\boldsymbol{\theta})$ is the evidence lower bound, i.e., $\mathrm{L}_q(\boldsymbol{\theta}) \leq \log p(\hat{\boldsymbol{X}}|\boldsymbol{\theta})$. That is, maximizing $\mathrm{L}_q(\boldsymbol{\theta})$ is equivalent to minimizing $\mathbb{KL}(q \parallel p)$. Note that Eq. 87 may easily be verified by plugging in $\log p(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}}|\boldsymbol{\theta}) = \log p(\hat{\boldsymbol{Z}}|\hat{\boldsymbol{X}}, \boldsymbol{\theta}) + \log p(\hat{\boldsymbol{X}}|\boldsymbol{\theta})$ in the expression for $\mathrm{L}_q(\boldsymbol{\theta})$, whereby the $\mathbb{KL}[q|p]$ term cancels out, leaving only $\mathbb{E}_q[\log p(\hat{\boldsymbol{X}}|\boldsymbol{\theta})] = \log p(\hat{\boldsymbol{X}}|\boldsymbol{\theta})$ on the right-hand side [6].

For this case, the EM algorithm therefore proceeds by sequentially optimizing $\mathrm{L}_q(\boldsymbol{\theta}^{(t-1)})$ with respect to $q(\hat{\boldsymbol{Z}})$ while holding $\boldsymbol{\theta}^{(t-1)}$ fixed (E-step), and optimizing $\mathrm{L}_q(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ while holding $q(\hat{\boldsymbol{Z}})$ fixed (M-step). In the E-step, $\mathrm{L}_q(\boldsymbol{\theta}^{(t-1)})$ is optimized when $\mathbb{KL}[q \parallel p]$ vanishes, which occurs when $q(\hat{\boldsymbol{Z}}) = p(\hat{\boldsymbol{Z}}|\hat{\boldsymbol{X}}, \boldsymbol{\theta}^{(t-1)})$. Thus, we arrive at the following expression for $\mathrm{L}_q(\boldsymbol{\theta})$ after the E-step

$$\begin{aligned}\mathrm{L}_q(\boldsymbol{\theta}) &= \sum_{\hat{\boldsymbol{Z}}} p(\hat{\boldsymbol{Z}}|\hat{\boldsymbol{X}}, \boldsymbol{\theta}^{(t-1)}) \log p(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}}|\boldsymbol{\theta}) - \sum_{\hat{\boldsymbol{Z}}} p(\hat{\boldsymbol{Z}}|\hat{\boldsymbol{X}}, \boldsymbol{\theta}^{(t-1)}) \log p(\hat{\boldsymbol{Z}}|\hat{\boldsymbol{X}}, \boldsymbol{\theta}^{(t-1)}) \\ &= Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)}) + \mathbb{H}[q], \end{aligned} \tag{122}$$

where the second term is the entropy of $q(\hat{\boldsymbol{Z}})$, which is a constant with respect to $\boldsymbol{\theta}$, thus we recover Eqs. 116–117 in the maximization [6].

Model selection The EM algorithm solves the problem of parameter estimation, but one additional problem persists, namely how to choose the number of mixture components K, i.e., the number of clusters represented in the data. One approach to address this issue is to define a likelihood-based score metric that penalizes model complexity, as the likelihood will simply increase as more mixture components are considered, which eventually will lead to overfitting. Two such metrics are the Bayesian information criterion (BIC) [121] and the integrated complete-data likelihood (ICL) [122], i.e.,

$$\mathrm{BIC}(\mathcal{M}) = \log p(\mathcal{D}|\hat{\boldsymbol{\theta}}) - \frac{\nu}{2} \log N \tag{123a}$$

$$\mathrm{ICL}(\mathcal{M}) = \mathrm{BIC}(\mathcal{M}) + \sum_{n=1}^{N} \sum_{k=1}^{K} \hat{z}_k[n] \log \gamma(z_k[n]), \tag{123b}$$

where $\mathcal{M}$ reflects a given model in terms of the number of mixture components and covariance structure, $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimate for the parameter vector under the model, ν is the number of free parameters in the model, $\gamma(z_k[n])$ is the responsibility of mixture component k in explaining $\boldsymbol{x}[n]$ (Eq. 119), and $\hat{z}_k[n]$ is the hard cluster assignment of $\boldsymbol{x}[n]$ based on $\gamma(\boldsymbol{z}[n])$, i.e.,

$$\hat{z}_k[n] = \begin{cases} 1 & \text{if } \arg\max_{k'} \gamma(z_{k'}[n]) = k \\ 0 & \text{otherwise.} \end{cases} \tag{124}$$

We note that the hard cluster assignment $\hat{z}_k$ in Eq. 123b may be replaced with the soft cluster assignment, i.e., $\gamma(z_k[n])$, leading to a reduction in the entropy penalty for uncertain cluster assignments [119] (Paper B).

As the BIC score tends to select the number of mixture components needed to reasonably approximate the density, rather than the number of clusters, the ICL score is used for model selection in the present study. It appears from Eq. 123b that the ICL score is a penalized version of the BIC score, which adds further penalization through an additional entropy term that reflects cluster overlap. In a Bayesian setting, the BIC metric appears by maximizing an approximation to the integrated (observed) likelihood, i.e., $p(\hat{\boldsymbol{X}}) = \int p(\hat{\boldsymbol{X}}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$, with non-informative priors, leading to a Laplace approximation at the MLE solution. The ICL metric appears by maximizing an approximation to the integrated complete-data likelihood, i.e., $p(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}}) = \int p(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$, under the same assumptions as for BIC, see [119, 120, 123] for further details (Paper B).

Some practical considerations In practical applications, we need to address two potential issues when seeking an MLE solution for mixture models, namely singularities and identifiability. Identifiability issues arise in mixture models, as there for a K-component mixture are $K!$ equivalent solutions, thus reflecting the $K!$ ways in which we can assigning K sets of parameters to K components. However, for the purpose of defining a density model, as we consider here, it does not matter which of the $K!$ equivalent solutions we pick [6]. Singularities occur when one of the clusters mean coincides with a point in the training data set and thus is an example of overfitting. One solution to this problem is to define a prior for the parameter vector $p(\Theta)$, in which case the EM algorithm leads to a MAP solution, which is equivalent to parameter regularization, see e.g., Sec. 5.1. Another solution is to use a suitable heuristic to detect singularities in the model during optimization, and if they occur, we reset the parameters of that cluster and continue with the optimization from that state [6].

7 MODEL SELECTION AND AVERAGING, AND DECISION OPTIMIZATION

In general systems modeling, the true causal relationships governing a given problem domain are often not known or result in representations that are not parsimonious. For such situations, model building typically leads to multiple plausible model hypotheses that provide adequate descriptions of the observed data, as only limited amounts of data are available and different modeling approaches may be utilized. In such situations, one best system representation is commonly selected from the ensemble according to some criterion, e.g., fit to data or predictive performance. After one model is selected, all inferences are made and conclusions drawn assuming that the selected model is the true model, thus ignoring model uncertainty (Paper F).

In this section, we consider two avenues for dealing with model uncertainty in inferential modeling and decision-making. In Sec. 7.1 and Appendix B, we discuss predictive inference using model averaging, and in Sec. 7.2, we introduce how the choice of system representation can be embedded inside an overarching decision context, which enables us to select the model best suited for a specific decision problem.

7.1 MODEL AVERAGING

Parts of this section appear in Papers D, and F.

Consider a finite ensemble of system representations $\boldsymbol{\mathcal{M}} = \{\mathcal{M}_u\}_{u=1}^{U}$, where each $\mathcal{M}_u$ corresponds to one system representation. Using model averaging, inferences are made by averaging over the ensemble of models as

$$Y_{\boldsymbol{\mathcal{M}}} = \sum_{u=1}^{U} w_u Y_u, \tag{125}$$

where $Y_{\boldsymbol{\mathcal{M}}}$ is the random outcome resulting from the weighted average of the individual, random outcomes $\{Y_u\}$ with corresponding model weights $\{w_u\}$ that add up to 1, i.e., $\sum_u w_u = 1$. The mean-squared error (MSE) for this average estimator may now be decomposed as

$$\begin{aligned}
\mathrm{MSE}(Y_{\boldsymbol{\mathcal{M}}}) &= \mathbb{E}\left[(Y_{\boldsymbol{\mathcal{M}}} - y^*)^2\right] \\
&= (\mathbb{E}[Y_{\boldsymbol{\mathcal{M}}}] - y^*)^2 + \mathbb{V}[Y_{\boldsymbol{\mathcal{M}}}] \\
&= \left(\sum_{u=1}^{U} w_u(\mathbb{E}[Y_u] - y^*)\right)^2 + \sum_{u=1}^{U}\sum_{u'=1}^{U} w_u w_{u'} \rho_{u,u'} \sigma_u \sigma_{u'}
\end{aligned} \tag{126}$$

where the first term in the decomposition corresponds to the bias between the average prediction and the truth, and the second term corresponds to the variance of the estimator. Moreover, $\rho_{u,u'}$ is the correlation coefficient between models $\mathcal{M}_u$ and $\mathcal{M}_{u'}$, and σ_u is the standard deviation of predictions from model $\mathcal{M}_u$ [124].

From Eq. 126, we see that for a given set of weights, the error of the average prediction depends on (i) the bias of the average model, which results from the biases of the averaged models, (ii) the predictive variance of the averaged models, and (iii) the covariances between averaged models. Under the assumption that the model bias of the averaged models tends to fall on both sides of the truth, the bias contribution of the average model will decrease compared to the individual model biases. Moreover, when the covariances between the averaged models are negligible, and the variances of the models are of similar size, i.e., σ, the resulting variance of the average model becomes σ^2/U, under the assumption of equal weights, i.e., $w_u = 1/U$. That is, the benefit of model averaging generally increase with decreasing covariance between the averaged models and decreasing mean bias of the averaged models [124].

We may further reduce the influence of poor models in the averaging by estimating the weights, but this, of course, introduces additional variability in the average model prediction, which can reduce the benefit of model averaging [124]. Following Dormann et al. [124], weighting schemes for model averaging can broadly be classified into four categories, which are listed below in order of increasing probabilistic interpretability.

Equal weighting: The simplest weighing scheme is to assign a uniform weight distribution over the averaged models, i.e., $w_u = 1/U$. This approach has been used with great success in ML applications, see Appendix B, and it is e.g., the approach taken in bagging [125, 126].

Heuristic weighting: We may also choose to optimize the model weights to achieve the best predictive performance of the average model by e.g., defining the averaging weights of each model in accordance with its predictive performance, usually obtained through cross-validation. In this regard, predictive performance may e.g., be defined as the (average) likelihood of a held-out data set under each of the averaged models. This approach has also been used successfully in ML applications, see Appendix B, and it is e.g., the approach taken in stacking [124, 126].

Information-theoretic weighting: Taking an information-theoretic prospective, the weights in model averaging reflect closeness, defined in terms of the Kullback-Leibler (KL) divergence, between the individual averaged models and the true, underlying data generating process. Common approximations to the KL-divergence is e.g., the Akaike informa-

tion criterion and the minimum description length, which negation as mentioned in Sec. 3.4 is equivalent to the Bayesian information criterion [124, 127].

Bayesian weighting: The Bayesian interpretation of the averaging weights is that they reflect the probability of the averaged models being the true model of the data generating process. In practice, inferences are made for the average model either by sampling from the joint distribution over model and parameters, i.e., $p(\mathcal{M}_u, \boldsymbol{\Theta}_u|\mathcal{D})$, using a Markov chain Monte Carlo scheme; or by approximating the posterior probability of each of the averaged models independently and using a normalized version of these posterior weights in Eq. 125, see e.g., [124, 126–128] for further details.

Note that Papers D, and F consider model averaging. As an example, Paper D uses a heuristic scheme, termed model-based model combination [124], to define the model weights through a radial-basis function kernel with distinct scaling parameters along the input dimensions. This weighting function is thus effectively equivalent to the automatic relevance detection kernel, as introduced in Sec. 4.1.

7.2 CONTEXT-SPECIFIC MODEL SELECTION

Parts of this section appear in Papers A, C, and F.

In a decision context, a system model $\mathcal{M}(a)$ provides a mapping from input to output, conditional on a decision alternative a, which is measured in terms of utility. In general, the system performance is uncertain, thus the optimal decision alternative needs to be selected in accordance with Bayesian decision theory [16] and the axioms of utility theory [17] by optimizing the expected utility, i.e.,

$$a^* = \arg\max_a(\mathbb{E}[\mathbb{U}(a)]). \tag{127}$$

The corresponding, principle decision event tree is illustrated in Fig. 13a. The figure shows the procedure for a so-called prior or posterior decision analysis, which only differ in the information available at the time of decision-making [129]. Thus, the decision maker has to choose between a set of decision alternatives (decision node) with uncertain outcomes (chance node) and associated utilities (utility node), and the rational of Eq. 127 is to choose the decision alternative that results in the maximum expected utility (benefit), see e.g., [27] (Papers A, C, and F).

In case the true underlying system is unknown, and it is unknown which is the most relevant representation of the system, the principle decision event tree for a prior or posterior decision analysis may be depicted as in Fig. 13b,

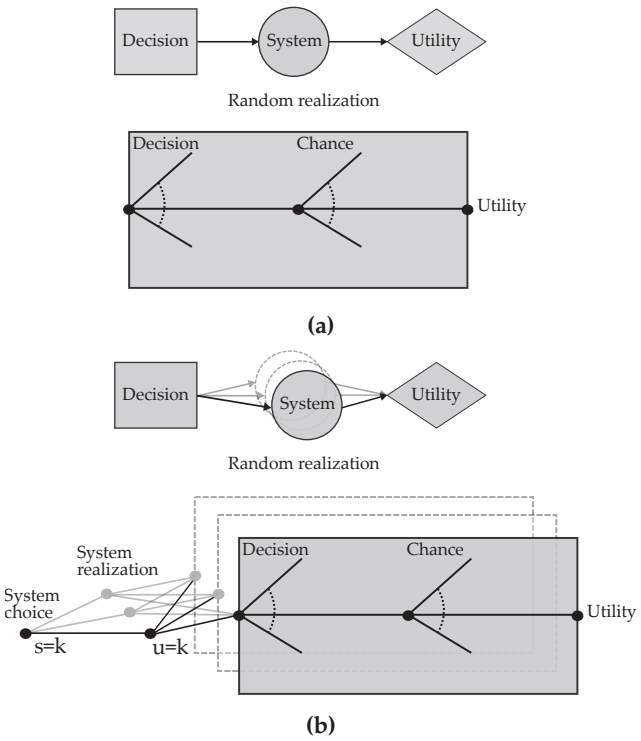


Figure 13: Systems role in decision analysis: (a) one possible system, (b) several possible systems (Papers C, and F).

see also [10, 12]. Following [10], the corresponding optimization for this case takes the following form

$$(s^*, a^*) = \arg\max_{s,a} \mathbb{U}(s,a) = \arg\max_{s} \Big(P(u=s) \arg\max_{a} \Big(\mathbb{E}_{\boldsymbol{X}|s}[\mathbb{U}(a,\boldsymbol{X})] \Big) + \mathbb{E}_{u' \in \boldsymbol{u} \setminus s} \Big[\mathbb{E}_{\boldsymbol{X}|u'}[\mathbb{U}(a^*,\boldsymbol{X})] \Big] \Big). \tag{128}$$

In this regard, the true system is represented by a random event with possible realizations $\boldsymbol{\mathcal{M}} = \{\mathcal{M}_u\}_{u=1}^{U}$ of known components indexed by u, and s represents the index of one choice of system representation in $\boldsymbol{\mathcal{M}}$. For this case, an additional complication arises, as some of the decision alternatives in $\boldsymbol{a}$ might not be admissible for some of the competing system representations. Therefore, the optimization of decision alternatives needs be be seen in consistency with the choice of system representation. For a given system choice, a^* is thus determined in accordance with Eq. 127 (Papers A, C, and F).

In Eq. 128, the robustness of the decision with regard to the choice of

system may be assessed as the ratio of the first term to the sum of the two terms, i.e.,

$$\text{Robustness}(s, a^*) = \frac{P(u = s)\mathbb{E}_{\boldsymbol{X}|s}[\mathbb{U}(a^*, \boldsymbol{X})]}{\mathbb{E}_{u' \in \boldsymbol{u}}\left[\mathbb{E}_{\boldsymbol{X}|u'}[\mathbb{U}(a^*, \boldsymbol{X})]\right]}. \tag{129}$$

This ratio takes a value between 0 and 1 (1=robust) that specify how sensitive the decision is to the possibility that the optimization is conducted under an erroneous system assumption [12, 130] (Paper C).

We may further have an option to collect additional information to support our decision analysis, see [12]. Thus, the pre-posterior decision analysis for a joint optimization considering a choice of (potential) additional information e, a choice of system s, and a choice of decision alternative a may be formulated as

$$\begin{aligned}(e^*, s^*, a^*) = \arg\max_e \mathbb{E}'_{\boldsymbol{Z}} \Big[\arg\max_s \Big(P''(u = s) \arg\max_a \Big(\mathbb{E}''_{\boldsymbol{X}|s}[\mathbb{U}(a, \boldsymbol{X})] \Big) \\ + \mathbb{E}''_{u' \in \boldsymbol{u} \setminus s} \Big[\mathbb{E}''_{\boldsymbol{X}|u'}[\mathbb{U}(a^*, \boldsymbol{X})] \Big] \Big) \Big],\end{aligned} \tag{130}$$

where $\boldsymbol{Z}$ are the random outcomes of the experimental strategies, and $\boldsymbol{z}$ are the corresponding realizations. Moreover, $\mathbb{E}''$ defines an expectation taken with respect to the updated probability assignments of the random variables included in the modeling, e.g., $P''(\boldsymbol{X}|s) = P'(\boldsymbol{X}|s, \boldsymbol{z})$ [12]. For this case, the corresponding, principle decision event tree is illustrated in Fig. 14.

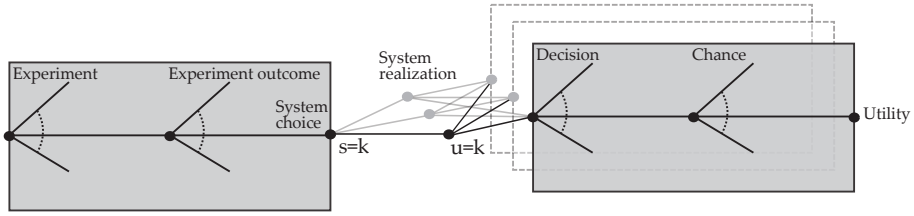


Figure 14: Pre-posterior decision analysis considering several possible systems.

This joint approach to decision optimization, when uncertainty exists on the optimal system representation, facilitates to integrate the model building process and the decision optimization, whereby the available knowledge can be fully utilized to optimize the expected utility associated with the considered system and consistently rank decision alternatives. This thus provides a principled means of addressing the trade-off between simplicity and complexity in modeling, as our models only need to be accurate in the domains of "reality" that matter for the decision subject to optimization. The interested

reader is referred to [12], as well as Papers A, C, and F, for further details on the approach and its applications.

8 APPLICATIONS IN OFFSHORE ENGINEERING

Parts of this section appear in Papers A–F.

In the preceding sections, we have studied the theoretical aspects of systems modeling and analysis with a special focus on the probabilistic representation of systems. In this section, we consider how the theory apply within the context of offshore engineering and point to the research papers in Part II, where it is applied to real-world problems and associated real data. Note again that practical implementations of the algorithms and numerical examples from the papers are found at the GitHub repository.

Environmental loads such as wind, waves, and current play key roles in the design and assessment of offshore structures [131, 132]. Figure 15 shows how structural responses are generated in storm events, which are the predominant, environmental exposure events for offshore structures. In this regard, the characterization of e.g., the wave loading in a storm follows a hierarchical approach, where we initially specify the sea state events (typically 1 or 3 hours) in terms of e.g., the significant wave height, zero-crossing period, direction, and directional spreading. Dependent on the sea state events, we then provided the statistical descriptions of short-term wave load characteristics of relevance for the structural responses, e.g., the crest height and shape as well as relevant time partial derivatives of the water particle positions over the water column. This approach has been significantly facilitated by wave tank experiments representing the sea surface elevation for given sea states in conjunction with theoretical models of the sea surface elevation within sea states, see e.g., [133–135].[10]

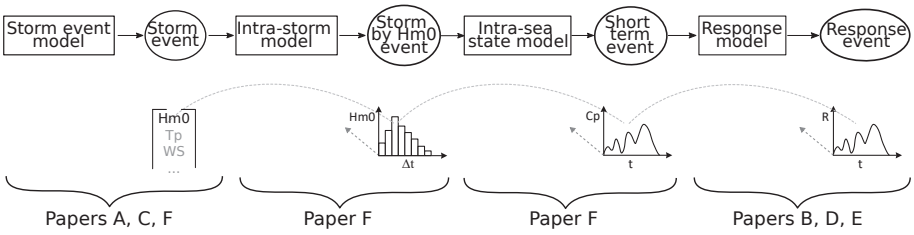


Figure 15: Flowchart illustrating the generation of structural responses as a result of exposure to environmental loads in a storm event.

Figure 15 focuses on how the environmental loads in a storm event lead to structural responses (far right), but the response model itself is an equally complex system composed of a set of subsystems, which are again composed

[10] `https://vbn.aau.dk/da/projects/load-environment-modeling-and-forecasting`

of sub-subsystems and so on. Thus, to target our efforts in the modeling of exposure events, it is imperative that we understand the drivers of critical system responses as well. Figure 15 frames the research papers in the context of offshore engineering as discussed in this section. In the following, the papers are shortly summarized.

Paper A: A framework for offshore load environment modeling. This paper is published in the proceeding of *the ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering (OMAE2018)*. The paper introduces a coherent framework for systems modeling and decision optimization, which may e.g., be utilized in offshore engineering applications, i.e., it builds on the material presented in Sec. 3 on Bayesian networks and Sec. 7.2 on context-specific model selection. The paper also includes a description of the North Sea region and presents the dominating phenomenology characterizing the offshore environment for this region. Moreover, the paper reviews current practices in design of offshore structures and related design information. A principle example is included to showcase the use of the framework. The example considers the modeling of environmental loads associated with storm events, and decision optimization in the context of the possible evacuation of an offshore facility in the face of an emerging storm event.

Paper B: Systems modeling using big data analysis techniques and evidence. This paper is published in the proceeding of *the IEEE 2019 4th International Conference on System Reliability and Safety (ICSRS2019)*. The paper explores how data mining and sensitivity analysis tools (Sec. 6) can be used as a means to improve understanding of complex probabilistic system representations. The proposed methodology to systems analysis includes the following steps: (i) generate system responses under given loading conditions, (ii) filter of system responses into failure scenarios, (iii) perform model-based cluster analysis on the realizations of the failure scenarios to detect distinct patterns and access regionalized sensitivities, and (iiii) access the sensitivity of system performance characteristics to the uncertain factors in the system representations. The individual steps of the methodology are illustrated in a principle example, which considers a portal frame structure under annual extreme loading conditions.

Paper C: A framework for offshore load environment modeling. This paper is published in the *Journal of Ocean, Offshore and Arctic Engineering*. The paper elaborates on the framework presented in Paper A and considers how to consistently account for parameter uncertainties in system representations using Bayesian networks, and propagate these

uncertainties in a decision optimization. This paper considers the same principle example as Paper A.

Paper D: On normalized fatigue crack growth modeling. This paper is published in the proceeding of *the ASME 2018 39th International Conference on Ocean, Offshore and Arctic Engineering (OMAE2020)*. The paper considers a new approach for the modeling of fatigue crack growth in structural details that accounts for the possible mutual dependencies in parameter space, which is typically ignored in today's best-practice fracture mechanics modeling approaches, where several parameters are assessed experimentally on an individual basis. We address the issue of consistent, evidence-based parameter estimation for the new so-called normalized fatigue crack growth model by framing it using parametric Bayesian hierarchical modeling and model-based machine learning (Secs. 3.5–3.6). The proposed probabilistic modeling scheme is presented and discussed based on an example considering fatigue crack growth for welds in K-joints. Finally, it is shown how the developed probabilistic crack growth model may be applied as basis for risk-based inspection and maintenance planning by utilizing Bayesian model averaging (Sec. 7.1).

Paper E: On a simple scheme for systems modeling and identification using big data techniques. This paper has been submitted to *Reliability Engineering & System Safety*. As the paper title indicates, the paper presents a simple scheme for systems modeling and identification, where focus is directed on the representation of our current state of knowledge about the considered system and letting the resulting so-called digital twin "speak" for itself through simulations of system responses. In this way, Monte Carlo simulation may be employed to establish the relevant scenarios of realizations of the random variables describing possible system states, including damage states, and system performance characteristics. On this basis, supervised classification provides a means for assessing the probability of the system being in a given state, provided new observations. Note that most applications on e.g., damage identification in civil engineering conduct the assessment somehow in reverse, i.e., by initially defining what to look for in terms of damage scenarios. This however has the disadvantage of not providing a means for assessing the relevance of the considered system states, measured in terms of likelihood of occurrence on the real system. The proposed scheme is illustrated through two principle examples considering damage identification in structural systems subject to extreme loading.

Paper F: On systems modeling and context-specific model selection in offshore engineering. This paper has been submitted to *Computer-Aided Civil and Infrastructure Engineering*. The paper builds on and extends the material presented in Papers A and C by first introducing the machinery behind the learning of Bayesian network (Sec. 3) and Gaussian process (Sec. 4) discrepancy modeling as a means for considering both measurements and corresponding simulation data in the Bayesian network modeling. Next, the paper goes on to present the competing systems frameworks of Sec. 7 and showcases how context-specific model selection may be applied on a simple principle example concerning the optimal design of a short concrete column, given a database of experimental outcomes from concrete compression strength tests. Finally, the paper considers how the modeling techniques and the decision analytical framework for model selection may be applied in a full-scale setting, where the objectives are (i) the formulation of a storm event model, based on measurements and corresponding hindcast data, and (ii) decision support on the potential evacuation of a set of offshore platforms, given observed storm characteristics.

9 CONCLUSIONS AND FUTURE WORK

This PhD study addresses the challenge of developing probabilistic models for the representation of systems in consistency with knowledge and information that might be or become available over time. Rather than following the traditional approach to model building, a fundamental and new perspective has been followed, where the probabilistic systems modeling should facilitate for the existence of several possible systems, which could underlie and explain the available knowledge and information. This perspective and conceptual abstraction might be considered to cause an increase in model complexity, which is thus in opposition to the general principle of simplicity in modeling, often expressed by Occam's razor. Therefore, the study seeks to understand how the trade-off between representativeness and simplicity might be formally operationalized by integrating the model building with the decision analysis, the modeling aims to inform.

Knowledge- and information-consistent systems modeling is paramount for integrity management in general, and for offshore engineering in particular. Through an honest representation of the current state of knowledge, and a deep understanding of the influences of and interactions between the variables governing system characteristics, the risks associated with different activities may be better understood. This in turn facilitates for the assessment of the expected value of benefits associated with different decision alternatives, and the corresponding ranking of these with the aim of managing the risks. In relation to the general objective of our group at the Danish Hydrocarbon Research and Technology Centre, namely to establish the knowledge basis for lifetime extensions of the existing offshore structures in the Danish North Sea, the research presented in this thesis brings us one step closer to this end.

9.1 CONCLUSIONS

Parts of this section appear in Papers A–F.

The main contribution of this PhD study is a principled framework for systems modeling and analysis, which is true to the prior knowledge, the available information, the possible competing system representations, and the decision context in which the system representations are applied. The principal novelty being the explicit accounting for model multiplicity and its integration into the overarching decision problem. It is emphasized that although the proposed framework is demonstrated within the context of (offshore) engineering, it is fully generic and applies to any context of model development, whereby it should be appreciated as a contribution to the general body of knowledge in model building.

In the following, the partial contributions within (offshore) engineering are categorized according to the general elements of systems modeling and analysis, which are addressed in the thesis, i.e., (i) representation of systems, (ii) analysis of systems, and (iii) model selection and decision optimization.

REPRESENTATION OF SYSTEMS

Storm event modeling Storm events are as mentioned the predominant environmental exposure events for offshore structures. As such, one outcome of this research is a generic approach for the modeling of storm events, which uses data-driven learning based on Bayesian networks (BNs) and provides a categorical representation of storm events. This approach further accommodates for a joint utilization of phenomenological understanding and information contained in databases by enabling prior knowledge to enter the data-driven modeling procedure at different stages, i.e., as constraints in structure learning and as priors in parameter learning. The potential of this approach is demonstrated in Papers A, C, and F, where it is shown how the approach may be used for inferring different (conditional) probability queries, for sampling storm events in relation to e.g., fatigue assessments, for extreme value assessments by conditioning on a category of storms likely to generate extreme responses, and for decision support in regard to the risk of exceedance of short-term load levels in a given storm event.

The approach thus consistently accounts for the available knowledge and information as well as the associated uncertainties, and together with the surveys provided in Paper A on the metocean environment of the North Sea, the traditional design practices in offshore engineering, and the commonly available metocean information sources, and the survey provided in Paper F on probabilistic approaches in offshore engineering, this research addressees all related research questions in Sec. 1.3 (Q1–Q4).

From the present research, the approach is found rather robust and efficient in representing storm event, and it is now at the stage of maturity, where it can be applied in large scale simulations for e.g., fatigue life assessments, provided the availability of an appropriate model for fatigue accumulation given a storm category (future work). Accompanying this work are two toolboxes for learning BNs – one for structure learning and automated discretization and one for parameters learning – both of which are able to handle a setting of fully observed data as well as partially observed data. The toolboxes are hosted at the GitHub repository along with tutorials on their use (Papers A, C, and F).

Normalized fatigue crack growth modeling Fatigue induced crack growth poses an important risk to the integrity of offshore structures. Therefore, one outcome of the study is a consistent approach for parameter estimation of a

new, so-called normalized fatigue crack growth model based on experimental evidence. The proposed approach is based on Bayesian hierarchical modeling in order to represent a super-population of experimental outcomes from different laboratories in a coherent manner, where information is allowed to flow between the models of the individual experiments by use of jointly modeled hyper-prior distributions. This allows for the best possible use of the data, as sparsely populated experiments gain statistical power in modeling from the other experiments under the assumption of similarity in crack propagation between experiments. Inferences for new details may be based on Bayesian model averaging over the hierarchically represented experiment models, and, for this purpose, a weighting scheme based on the radial-basis function kernel is proposed.

This work also addresses all related research questions in Sec. 1.3 (Q1–Q4) by showcasing how to consistently handle the available and potential future knowledge and information; by utilizing the state-of-the-art modeling methodology of model-based machine learning; and by illustrating that the proposed approach allow for a consistent representation of within and between group variability, contrary to the procedure currently in use, where uncertainties are represented on a point-by-point basis.

The research is documented in Paper D, and all figures and results related to the model development in the paper are reproduced in a notebook available at the GitHub repository. With this model, the basis for risk-based inspection and maintenance planning is established, and thus Paper D also provides a small principle example on how this may be performed using the model (Paper D).

Discrepancy modeling In this study, discrepancy modeling is used as a means for correcting simulator outputs based on corresponding measurements, i.e., combining/fusing different data sources. Thus, this research mainly addresses research questions Q1 and Q4 of Sec. 1.3 by considering how to combine different sources of information, and subsequently propagate uncertainties related to the discrepancy model. To this end, Gaussian processes (GPs) are found to be efficient emulators of the error function, which are capable of representing the uncertainty in the model representation.

An application of discrepancy modeling is shown in Paper F, where the errors in a hindcast data set of storm events are modeled by considering a data set of corresponding partial measurements of the storm events. Moreover, Paper E uses GPs to emulate cost functions related to general machine learning algorithms in a Bayesian optimization scheme for hyper-parameter tuning. At the GitHub repository, several notebooks considering GP surrogate modeling are available (Paper F).

As a spin-off on this work, a collaboration with DTU Mechanical Engineering has been initiated that until now has focused on discrepancy modeling related to Morrison's equation based on measurements of forces on a cylindrical body under different turbulence intensities. This collaboration has so far, apart from Paper F, resulted in a successful M.Sc. project on the subject, and currently, among others, this work is being extended in a joint research paper.

ANALYSIS OF SYSTEMS

Clustering and sensitivity assessments of system characteristics A system representation of a real (offshore) system is a complex object comprised of an ensemble of constituents interacting jointly to provide the functionalities of the system. Thus, understanding the nature and drives of response characteristics for such systems is generally challenging. This research takes up this challenge by studying how modern techniques of data mining may be used to this end. First, model-based cluster analysis is used to establish a probabilistic representation of realizations leading to system performances of interest. The research highlights that cluster analysis provides not only a strong means for checking the relevance and physical adequacy of complex system models, but also a significant insight into how complex models may be designed, modified, and/or maintained to achieve adequate and cost-efficient performance characteristics with respect to e.g., robustness and resilience.

Second, variance-based sensitivity analysis is used to decompose the variance of responses or their indicators in order to gain insight as to how uncertainties in the system constituents propagate to and influence uncertainties in system response characteristics. In this regard, the present research shows how sensitivity analysis may be used for model reduction as well as for identifying response characteristics that contain significant information about the state of the system, in the case of both uncorrelated system inputs (ANOVA) and correlated system inputs (ANCOVA). The identification of the uncertain system inputs driving the uncertainty in system outputs is of particular interest in e.g., structural health monitoring, where response characteristics that contain significant information about the state of the system must be identified and observed with the highest possible degree of accuracy.

This research, which is documented in Paper B, addresses both related research questions in Sec. 1.3 (Q5–Q6) by exploring state-of-the-art technologies for systems analysis, and by devising (together with Paper E) a consistent scheme for exploring the space of possible, competing systems and rank their relevance in terms of occurrence. Note that the GitHub repository contains tutorials on both cluster and sensitivity analysis, as they are explained and used in this thesis (Paper B).

As a spin-off on this work, the authors of Paper B (myself included) are

currently involved in a project named CodeWrapper undertaken by Total E&P Denmark. In this regard, a significant number of Monte Carlo based structural reliability assessments (SRAs) have been and are presently being performed in order to support the understanding of how the reliability of the considered portfolio of offshore structures varies over different structural layouts. One of our tasks in this project is the identification of design load cases, which among others entails the discovery of patterns of realizations in the database that give rise to similar SRA responses. For this purpose, cluster analysis as explained in Paper B is used on Monte Carlo filtered data of different structural layouts and response levels.

Structural damage identification The identification of structural changes due to different kinds of damages is one of two main pillars of a functioning structural health monitoring (SHM) system; the other being efficient and accurate harvesting of information in terms of damage-sensitive, structural response features, i.e., indicators for structural damages. Such SHM systems may e.g., be used for condition screening to provide near real time predictions regarding the integrity of structures in relation to the occurrence of extreme load events, as it is framed in Paper E. This situation generally reflects a special case of competing system representations, where each damage state represents one possible system, and the objective is to rank the systems according to their likelihood in generating the considered response and manage the system accordingly based on the associated risks.

When analyzing damage scenarios in a simulation study or using real data, these are most often unevenly distributed in likelihood of occurrence, thus resulting in an imbalanced database in the damage scenarios, which if not properly accounted for may result in a bias towards better represented damage scenarios when fitting a model to the data. Therefore, one outcome of this research is an efficient, random-walk Markov chain Monte Carlo scheme for generating additional realizations of poorly represented damage scenarios, which is explained and demonstrated in Paper E.

Based on the resulting balanced database of realizations of damage scenarios, a classifier can be fitted to the data, which enables a likelihood-based (soft) classification of new realizations from the structure. In this regard, Paper E displays how the hyper-parameters of state-of-the-art tree-based classification models may be tuned using GP-based Bayesian optimization to define well performing classifiers in terms of the cost function. The numerical examples in the paper show that the resulting classifiers (i) with a high level of precision (small type I and type II errors) identify the correct states of damages, (ii) scale well to the size of the training data set, and (iii) behave robustly to changes in the number of training samples per damage scenario and the observation noise level, respectively. Moreover, initial value of information

assessments point to the potential benefits of embedding the classification framework in a pre-posterior decision analysis, which enables the optimizing of strategies for collecting observations of system responses.

This research is as noted documented in Paper E, which addresses both related research questions in Sec. 1.3 (Q5–Q6) by exploring state-of-the-art methodologies for systems identification, and by devising (together with Paper B) a consistent scheme for exploring the space of possible competing systems and rank their relevance in terms of occurrence. The GitHub repository contains a set of tutorials on tree-based modeling using gradient boosting, as implemented in Paper E. These tutorials also consider different implementational details when defining a tree-based classifier and shows how GP-based Bayesian optimization for such models may be implemented (Paper E).

MODEL SELECTION AND DECISION OPTIMIZATION

Phenomenological understanding as well as analysis of databases often lead to the identification of several competing system representations that explain the data almost equally well in terms of data likelihood. Traditionally, such situations are handled by statistical model selection or averaging on the basis of a data fitness criterion. However, if we acknowledge that the reason for formulating system representations in the first place is to serve decision-making in the generic context of systems performance management, we may be able to choose the best suited system representation for the decision context at hand. Therefore, one outcome of this research is a novel decision analytical framework for systems modeling in the context of risk-informed integrity management of offshore facilities, where the problem of systems modeling is embedded within an optimization problem to be solved jointly with the ranking of decision alternatives (research question Q7 in Sec. 1.3). Note that this work should be seen in conjunction with and related to the early developments in [10], and the resent contributions in [12], where, as mentioned, I act as co-author.

The potential of this research is demonstrated in Papers A, C, and F, where it is shown how the optimization may be setup and undertaken in the case of previously identified competing system representations (research question Q8(i) in Sec. 1.3), as well as in case the set of possible system representations is dynamically adjusted to best accommodate the decision problem at hand (research question Q8(ii) in Sec. 1.3). The latter case directly targets the model building process by appreciating that our models only need to be accurate in the areas of the problem domain, which matter for the decision subject to optimization. This research thus presents a formal, theoretical basis for approaching the principle of simplicity (Occam's razor) in a consistent, quantifiable manner. Additionally, this work advances on the representation and propagation of uncertainties in decision problems and the implications of

doing so, both in terms of the interpretation of results and the computational efforts needed (Papers A, C, and F).

9.2 FUTURE WORK

This study brings us one step closer to information-consistent systems modeling and analysis in engineering under due consideration of the overarching decision context, but it also sheds light on the elements still missing, things that may be improved, and where to turn next. This section shares some thoughts and ideas on these aspects and points towards future work. This section is also organized according to the general elements of systems modeling and analysis addressed in this thesis, i.e., (i) representation of systems, (ii) analysis of systems, and (iii) model selection and decision optimization.

REPRESENTATION OF SYSTEMS

Storm event modeling Different improvements to and extensions of the presented storm event model have been discussed. Among others, full-scale examples on the use of the model and associated models in relation to fatigue assessments, conditional extreme values analysis, and forecasting and early warning systems. In this regard, a so-called storm evolution model, or intra-storm model, will be developed in order to define the storm content in terms of sea state parameters and their joint evolution over a storm event, given storm characteristic. Moreover, in collaboration with DTU Mechanical Engineering, a project on spatial and temporal modeling of extreme sea state characteristics and their associated effects on structural responses will be launched. This research will build on extensive laboratory experiments conducted at the Danish Hydraulic Institute.

Algorithmically, the structure learning and automated discretization toolbox (GitHub repository) will be extended by including an alternative to the current greedy hill-climbing optimization strategy for dynamic discretization. This alternative optimization algorithm will be based on dynamic-programming and will provide a globally optimal discretization policy. Note that this strategy will only be feasible for problems with a limited number of domain variables. Also, a reimplementation of many of the existing functions in the toolbox is needed to accommodate custom objective functions, which will allow for a set of possible, competing system representations to be dynamically adjusted during an enveloping decision optimization. Early developments along this line are documented in Paper F.

Normalized fatigue crack growth modeling The implementation of the normalized fatigue crack growth model is currently being and will further be extended in the coming years, both by me and my co-workers, but also

by a new project at the Danish Hydrocarbon Research and Technology Centre (DHRTC) called Corrosion and Fatigue of Offshore Structures. Currently, we are working on harvesting more information in terms of fatigue experiments for the model formulation, as well as algorithmic changes to the current framework, which will make it more targeted and computationally efficient. Also, we are considering non-parametric alternatives to the current model formulation.

Discrepancy modeling As mentioned, a joint research paper with DTU Mechanical Engineering on discrepancy modeling related to Morrison's equation is on its way, and other related ideas on the modeling of the force field on offshore structures are taking shape. This also includes ideas on multi-fidelity modeling, which can be thought of as a way embedding discrepancy modeling in the modeling of systems when data are available on different fidelity levels, as for example in the database considered in Paper F, where both hindcast and partial measurements are considered.

ANALYSIS OF SYSTEMS

Clustering and sensitivity assessments of system characteristics The potential of using data mining and sensitivity analysis techniques as a means for understanding complex systems is by no means exhausted in this and other related studies. Among other, the authors of Paper B are currently using these techniques for quantifying robustness and resilience as well as understanding how we can most efficiently improve these properties of a structural system.

Structural damage identification Damage and anomaly detection using statistical pattern recognition approaches have been active research topics for at least two decades and important grounds have been covered, but work is still needed in order to make technologies such as structural health monitoring accurate and efficient for the multitude of system changes experienced in complex real-world systems. In this regard, the framework of Paper E for the identification of relevant damage scenarios is currently being considered in more complex settings with different failure modes. Also, approaches to transfer findings from one system to other similar systems will be explored in order to reduce the computational burden related to the identification of relevant damage scenarios for similar systems. This is referred to as transfer learning in machine learning.

A new project called InnoSHM has just been launched. In this project, we are working together with among others Total E&P Denmark, Ramboll, and DTU Civil Engineering on devising a structural health monitoring campaign, where special attention is given to damage feature design, e.g., identification

of potential indicators for different failures in offshore structures, feature selection, and data fusion and normalization. Especially, feature normalization poses a major challenge when realizing a structural health monitoring campaign for offshore structures, as many benign system changes are experienced on a daily basis e.g., due to mass changes in the form of filled/empty tanks, and varying staffing and equipment.

MODEL SELECTION AND DECISION OPTIMIZATION

The early developments on the dynamic adjustment of the set of potential, relevant competing system representations, as presented in this study (Paper F), are currently being investigated for both simple, stereo-type design settings, where there are potential for defining analytical solutions, and more complex settings, e.g., the storm event modeling. Also, developments regarding the operationalization of the related ideas on consistent handing of (conflicting) information and fake news presented in [12] are on the menu for future research.

A INFERENCE ALGORITHMS FOR BAYESIAN NETWORKS

In this appendix, we highlight some commonly used methods for performing inference in Bayesian networks (BNs) and provide some key references. In Sec. A.1, we consider methods for exact inference, and in Sec. A.2, we consider methods for approximate inference.

A.1 EXACT INFERENCE

VARIABLE ELIMINATION

Variable elimination (VE) is a general and simple exact inference algorithm for probabilistic graphical models, such as Bayesian and Markov networks. It computes conditional probability queries by pushing summations into the factor product Eq. 3, whereby the variables are marginalized out one by one in smaller sub-products of the factors. That is, VE allows us to perform local operations on relevant sub-products of factors, instead of having to work with the entire joint distribution [24, 28].

BELIEF PROPAGATION

Belief propagation (BP) is an exact inference algorithm for probabilistic graphical models when applied on junction trees, also known as clique trees. A junction tree $\mathcal{T}$ is a secondary computational structure representing a BN model, with nodes corresponding to subsets of the domain variables $\boldsymbol{X}$, i.e., cliques. Evidence is propagated in a junction tree by passing messages between the cliques in two sweeps. First, one clique is selected as the root, and messages are passed from the leaves towards the root. This is referred to as collection of information. Second, messages are passed in the opposite direction, i.e., from the root towards the leaves. This is referred to as distribution of information. After the evidence has been propagated, the junction tree is said to be in equilibrium, and a query $P(Y|\boldsymbol{\mathcal{E}} = \boldsymbol{\varepsilon})$ may be calculated from any clique potential in the tree containing Y. In case we are interested in a query $P(\boldsymbol{Y}|\boldsymbol{\mathcal{E}} = \boldsymbol{\varepsilon})$, where the query variables $\boldsymbol{Y}$ span several cliques, i.e., they are not present in the same clique, VE is performed on the unnormalized joint distribution formed by the relevant clique potentials corresponding to the (connected) subtree $\mathcal{T}'$, where $\boldsymbol{Y} \subseteq Scope(\mathcal{T}')$ [24, 27].

A.2 APPROXIMATE INFERENCE

PARTICLE-BASED INFERENCE

Particle-based methods use the probabilistic model (generative model) to generate instances (particles) from the distribution $P(\boldsymbol{Y}|\boldsymbol{\mathcal{E}} = \boldsymbol{\varepsilon})$, and answer queries based on these particles. Thus, particle-based methods perform approximate inference, and the accuracy of the approximation depends on the efficiency of the algorithm applied as well as the number of particles sampled from the distribution. There are a vast number of algorithms for generation particles in BNs, e.g., rejection sampling and Gibbs sampling. In rejection sampling, we sample directly from the desired distribution by disregarding the samples that do not comply with our current query specification $\boldsymbol{\mathcal{E}} \neq \boldsymbol{\varepsilon}$. Thus, this procedure is computationally expensive, as we only expect to accept $P(\boldsymbol{\varepsilon})$ of the particles we generate. In Gibbs sampling, we construct a Markov chain, which eventually samples from the target distribution, see e.g., Eq. 19. A Gibbs sampler is relatively easy to implement and computationally efficient to sample from but mixing may be slow, especially in models where the variables are highly correlated [24, 36, 41]

VARIATIONAL INFERENCE AND EXPECTATION PROPAGATION

Variational inference (VI), also called variational Bayes, and expectation propagation (EP) are two alternatives to particle-based methods for performing full Bayesian inference over complex distributions that are hard to directly evaluate or sample from. Whereas particle-based techniques provide a numerical approximation to the exact posterior using a set of samples, VI and EP provide a locally optimal, exact analytical solution to an approximation of the posterior, which is termed the variational distribution or local likelihood approximation. The variational distribution is defined by considering a tractable family of distributions for which the parameters are optimized to approximate the true posterior. Thus, VI and EP turn inference into an optimization problem. One score metric that is often used to assess the quality of the variational distribution is the Kullback–Leibler (KL) divergence, which generally measures the closeness of two distributions. In this regard, VI minimizes $\mathbb{KL}[Q \parallel P]$, where $P(\boldsymbol{X})$ is the true distribution and $Q(\boldsymbol{X})$ is the variational distribution, and EI minimizes $\mathbb{KL}[P \parallel Q]$. Compared to particle-based methods, both methods are fast and scales well to large data sets, but the VI algorithm is mode seeking, and thus it may underestimate the variance of the true posterior, whereas the EP algorithm is moment matching, and thus may lead to a poor approximation, if the true posterior is multimodal [6, 73, 136, 137].

B MODEL AVERAGING IN MACHINE LEARNING

Parts of this section appear in Paper D.

The machine learning literature contains a variety of general-purpose procedures for stabilizing model predictions of statistical learners. Among the most popular are bagging, boosting, and stacking [126, 138]. In this appendix, we will briefly discuss these concepts and relate them to the general introduction of model averaging in Sec. 7.1.

In bagging, or bootstrap aggregation, the original data set $\mathcal{D}$ is re-sampled uniformly with replacement U times to produce a bootstrap sample $\{\mathcal{D}_u\}_{u=1}^{U}$ of data sets, and a model (weak/base learner) $\mathcal{M}_u$ is trained on each bootstrap replicate. Predictions for new data points are made by averaging the outputs from each model in a regression setting or e.g., majority voting among the outputs in a classification setting [125]. Random forest is a popular variant of bagging that leverages classification and regression trees (CARTs) [126] and random selection among the input variables before each split in the tree-growing process to further de-correlate the ensemble models. After U trees are grown, the bagged (average) estimator follow from Eq. 125. Note that this procedure corresponds to model averaging with equal weights.

In boosting, we also typically build an ensemble of tree models, but opposite to bagging, where the trees are grown in parallel, we grow the trees sequentially, where each tree is grown using information on the predictive performance of previous trees in the sequence. Thus, at each stage, we fit a new model that focuses on the errors of the current ensemble model and consequently add this new CART to the ensemble model. After U trees are grown, the boosted (average) estimator follow from Eq. 125. No simple algorithm exists for solving the sequential optimization problem in boosting for general loss criteria, and we thus need to resort to numerical optimization procedures like (functional) gradient decent [126, 138]. This gives rise to a very popular variant of boosting called gradient boosting (machines), which is continuously coming out in the top of machine learning competitions like Kaggle;[11] not at least to due efficient implementations such as XGBoost [139]. Note that this procedure corresponds to model averaging with optimized weights (Paper E).

Where bagging and boosting typically consider an ensemble of homogeneous base models, e.g., each member of the ensemble is a CART, stacking typically considers an ensemble of heterogeneous base models, e.g., we may want to combine predictions from a linear regression and an neural network, and the weights are chosen such that a hold-out cross validation error metric

[11] `https://www.kaggle.com/`

is minimized. This procedure thus corresponds to model averaging with optimized weights. After U models are fitted in parallel on the original training set $\mathcal{D}$, one possible stacked (average) estimator follows from Eq. 125, but the stacking framework is more general; in principle any learning algorithm from the literature could be used in place of Eq. 125 to map predictions from the base models to a (joint) stacked prediction [126]. For example, in a regression setting, we may want to stack predictions from e.g., a k-nearest-neighbor regression, a linear regression and a support vector regression using a neural network, see e.g., [4, 6, 126] for a reference on the individual models. The neural network then takes the outputs from the base models as inputs and provides a stacked prediction.

REFERENCES

[1] J. Tychsen, S. Risvig, H. Fabricius Hansen, N. Ottesen Hansen, and F. Stevanato, "Summary of the impact on structural reliability of the findings of the tyra field extreme wave study 2013-2015," in *Third Offshore Structural Reliability Conference, OSRC2016, Stavanger, Norway*, 2016, pp. 1–12.

[2] J. Tychsen and M. Dixen, "Wave kinematics and hydrodynamic loads on intermediate water depth structures inferred from systematic model testing and field observations–tyra field extreme wave study 2013-15," in *Third Offshore Structural Reliability Conference*, 2016, pp. 1–10.

[3] M. Kuhn and K. Johnson, *Feature engineering and selection: A practical approach for predictive models*. CRC Press, 2019.

[4] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[5] I. Newton, *The mathematical principles of natural philosophy*. Benjamin Motte, 1729, vol. 2.

[6] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[7] K. Menger, "A counterpart of occam's razor in pure and applied mathematics ontological uses," *Synthese*, vol. 12, no. 4, pp. 415–428, 1960.

[8] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.

[9] S. Theodoridis, *Machine learning: a Bayesian and optimization perspective*. Academic press, 2020.

[10] M. H. Faber and M. A. Maes, "Epistemic uncertainties and system choice in decision making," in *Ninth International Conference on Structural Safety and Reliability, ICOSSAR*, 2005, pp. 3519–3526.

[11] Joint Committee on Structural Safety (JCSS), *Risk assessment in engineering: principles, system representation & risk criteria*. Ed. M. H. Faber, 2008.

[12] L. Nielsen, S. T. Glavind, J. Qin, and M. H. Faber, "Faith and fakes–dealing with critical information in decision analysis," *Civ Eng Environ Syst*, vol. 36, no. 1, pp. 32–54, 2019.

[13] M. H. Faber, "On the treatment of uncertainties and probabilities in engineering decision analysis," *Journal of Offshore Mechanics and Arctic Engineering*, vol. 127, no. 3, pp. 243–248, 2005.

[14] J. D. Sørensen and H. Toft, "Probabilistic design of wind turbines," *Energies*, vol. 3, no. 2, pp. 241–257, 2010.

[15] A. Der Kiureghian and O. D. Ditlevsen, "Aleatoric or epistemic? does it matter?" *Structural Safety*, vol. 31, no. 2, pp. 105–112, 2009.

[16] H. Raiffa and R. Schlaifer, *Applied statistical decision theory*. MIT Press, 1961.

[17] J. von Neumann and O. Morgenstern, *Theory of games and economic behavior*. Princeton University Press, 1953.

[18] L. Wasserman, *All of Statistics : A Concise Course in Statistical Inference*. Springer New York, 2004.

[19] M. Kuhn and K. Johnson, *Applied predictive modeling*. Springer, 2013.

[20] V. Vapnik, *Statistical learning theory*. Wiley New York, 1998.

[21] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Advances in Neural Information Processing Systems*, vol. 14, 2002, pp. 841–848.

[22] D. Heckerman, "Bayesian networks for data mining," *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 79–119, 1997.

[23] ——, "A tutorial on learning with bayesian networks," 2020, (accessed 16 October 2020). [Online]. Available: https://arxiv.org/abs/2002.00269

[24] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.

[25] J. Pearl, *Probabilistic Reasoning in Intelligent Systems. Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[26] ——, *Causality*. Cambridge university press, 2009.

[27] U. B. Kjærulff and A. L. Madsen, *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*. Springer New York, 2013.

[28] F. V. Jensen and T. D. Nielsen, *Bayesian networks and decision graphs*. Springer Science & Business Media, 2007.

[29] R. Daly, Q. Shen, and S. Aitken, "Learning bayesian networks: approaches and issues," *The Knowledge Engineering Review*, vol. 26, no. 2, p. 99–157, 2011.

[30] J. Zhu, J. Chen, W. Hu, and B. Zhang, "Big learning with bayesian methods," *National Science Review*, vol. 4, no. 4, pp. 627–651, 2017.

[31] M. Scutari and J.-B. Denis, *Bayesian networks: with examples in R*. CRC press, 2014.

[32] C. M. Bishop, "Model-based machine learning," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, p. 20120222, 2013.

[33] R. Nagarajan, M. Scutari, and S. Lèbre, *Bayesian Networks in R: with Applications in Systems Biology*. Springer, 2013.

[34] M. Ueno, "Learning networks determined by the ratio of prior and data," in *26th Conference on Uncertainty in Artificial Intelligence*, 2010, pp. 1–8.

[35] D. Barber, *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[36] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*. CRC press, 2013.

[37] M. Scutari, "Understanding bayesian networks with examples in r," (accessed 1 October 2018). [Online]. Available: http://www.bnlearn.com/about/teaching/slides-bnshort.pdf

[38] K. B. Korb and A. E. Nicholson, *Bayesian artificial intelligence*. CRC press, 2010.

[39] Z. Ji, Q. Xia, and G. Meng, "A review of parameter learning methods in bayesian network," in *Advanced Intelligent Computing Theories and Applications*, 2015, pp. 3–12.

[40] M. Scutari, "Bayesian network models for incomplete and dynamic data," *Statistica Neerlandica*, vol. 74, no. 3, pp. 397–419, 2020.

[41] C. P. Robert and G. Casella, *Introducing Monte Carlo Methods with R*. Springer New York, 2010.

[42] R. E. Neapolitan, *Learning bayesian networks*. Pearson Prentice Hall, 2004.

[43] D. M. Chickering, "Learning bayesian networks is np-complete," in *Learning from Data: Artificial Intelligence and Statistics V*. Springer-Verlag, 1996, pp. 121–130.

[44] T. Silander and P. Myllymäki, "A simple approach for finding the globally optimal bayesian network structure," in *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, 2006, pp. 445–452.

[45] T. Jaakkola, D. Sontag, A. Globerson, and M. Meila, "Learning bayesian network structure using lp relaxations," in *Thirteenth International Conference on Artificial Intelligence and Statistics*, vol. 9, 2010, pp. 358–365.

[46] M. Studený and D. Haws, "Learning bayesian network structure: Towards the essential graph by integer linear programming tools," *International Journal of Approximate Reasoning*, vol. 55, no. 4, pp. 1043 – 1071, 2014.

[47] T. J. Koski and J. M. Noble, "A review of bayesian networks and structure learning," *Mathematica Applicanda*, vol. 40, no. 1, pp. 51–103, 2012.

[48] M. Drton and M. H. Maathuis, "Structure learning in graphical modeling," *Annual Review of Statistics and Its Application*, vol. 4, no. 1, pp. 365–393, 2017.

[49] D. M. Chickering, "Optimal structure identification with greedy search," *Journal of machine learning research*, vol. 3, pp. 507–554, 2002.

[50] M. Scutari, C. E. Graafland, and J. M. Gutiérrez, "Who learns better bayesian network structures: Accuracy and speed of structure learning algorithms," *International Journal of Approximate Reasoning*, vol. 115, pp. 235–253, 2019.

[51] N. Friedman, M. Goldszmidt, and A. Wyner, "Data analysis with Bayesian networks: A bootstrap approach," in *Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 196–205.

[52] G. Melançon and F. Philippe, "Generating connected acyclic digraphs uniformly at random," *Information Processing Letters*, vol. 90, no. 4, pp. 209–213, 2004.

[53] J. S. Ide and F. G. Cozman, "Random generation of bayesian networks," in *Brazilian symposium on artificial intelligence*. Springer, 2002, pp. 366–376.

[54] G. Melançon, I. Dutour, and M. Bousquet-Mélou, "Random generation of directed acyclic graphs," *Electronic Notes in Discrete Mathematics*, vol. 10, pp. 202–207, 2001.

[55] C. Riggelsen, "MCMC learning of bayesian network models by markov blanket decomposition," in *European Conference on Machine Learning*. Springer, 2005, pp. 329–340.

[56] N. Friedman and D. Koller, "Being bayesian about network structure," in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2000, pp. 201–210.

[57] P. Myllymaki, T. Silander, H. Tirri, and P. Uronen, "B-course: a web-based tool for bayesian and causal data analysis," *International Journal on Artificial Intelligence Tools*, vol. 11, no. 3, pp. 369–387, 2002.

[58] L. Uusitalo, "Advantages and challenges of bayesian networks in environmental modelling," *Ecological Modelling*, vol. 203, no. 3-4, pp. 312–318, 2007.

[59] S. Monti and G. F. Cooper, "A multivariate discretization method for learning bayesian networks from mixed data," in *Fourteenth International Conference on Uncertainty in Artificial Intelligence*, 1998, pp. 404–413.

[60] K. Vogel, "Applications of bayesian networks in natural hazard assessments," Ph.D. dissertation, University of Potsdam, 2013.

[61] N. Friedman and M. Goldszmidt, "Discretizing continuous attributes while learning bayesian networks," in *Thirteenth International Conference on Machine Learning*, 1996, pp. 157–165.

[62] N. Friedman, "Learning belief networks in the presence of missing values and hidden variables," in *Fourteenth International Conference on Machine Learning*, 1997, pp. 125–133.

[63] ——, "The Bayesian structural EM algorithm," in *Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1998, pp. 129–138.

[64] M. A. Tanner and W. H. Wong, "The calculation of posterior distributions by data augmentation," *Journal of the American statistical Association*, vol. 82, no. 398, pp. 528–540, 1987.

[65] M. Scutari, "Learning bayesian networks with the bnlearn R package," *Journal of Statistical Software*, vol. 35, no. 3, pp. 1–22, 2010.

[66] K. P. Murphy, "Dynamic bayesian networks: representation, inference and learning," Ph.D. dissertation, University of California, Berkeley, CA, 2002.

[67] J. Winn, C. M. Bishop, T. Diethe, J. Guiver, and Y. Zaykov, "Model-based machine learning," (accessed 15 July 2019). [Online]. Available: http://mbmlbook.com/

[68] D. J. Lunn, A. Thomas, N. Best, and D. Spiegelhalter, "Winbugs - a bayesian modelling framework: Concepts, structure, and extensibility," *Statistics and Computing*, vol. 10, no. 4, pp. 325–337, 2000.

[69] Infer.NET Development Team, "Infer.NET 0.3."

[70] M. Plummer, "JAGS: A program for analysis of bayesian graphical models using gibbs sampling," in *Proceedings of the 3rd international workshop on distributed statistical computing*, vol. 124, no. 125. Vienna, Austria., 2003, pp. 1–10.

[71] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, and A. Riddell, "Stan: A probabilistic programming language," *Journal of Statistical Software*, vol. 76, no. 1, 2017.

[72] Stan Development Team, "Prior choice recommendations," 2020, (accessed on 1 December 2020). [Online]. Available: https://github.com/stan-dev/stan/wiki/Prior-Choice-Recommendations

[73] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT press, 2006.

[74] P. Li and S. Chen, "A review on gaussian process latent variable models," *Caai Transactions on Intelligence Technology*, vol. 1, no. 4, pp. 366–376, 2016.

[75] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, 2012, pp. 2960–2968.

[76] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of bayesian optimization," in *Proceedings of the IEEE*, vol. 104, no. 1, 2016, pp. 148–175.

[77] R. B. Gramacy, *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. Chapman and Hall/CRC, 2020.

[78] G. Yi, J. Q. Shi, and T. Choi, "Penalized gaussian process regression and classification for high-dimensional nonlinear data," *Biometrics*, vol. 67, no. 4, pp. 1285–1294, 2011.

[79] M. A. Álvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review," *Foundations and Trends in Machine Learning*, vol. 4, no. 3, pp. 195–266, 2012.

[80] E. V. Bonilla, K. M. Chai, and C. Williams, "Multi-task gaussian process prediction," in *Advances in neural information processing systems*, 2008, pp. 153–160.

[81] H. Liu, J. Cai, and Y.-S. Ong, "Remarks on multi-output gaussian process regression," *Knowledge-Based Systems*, vol. 144, pp. 102–121, 2018.

[82] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[83] E. Brochu, V. M. Cora, and N. de Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," 2010, (accessed on 10 August 2020). [Online]. Available: https://arxiv.org/abs/1012.2599

[84] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[85] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in Neural Information Processing Systems*, 2011.

[86] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proceedings of the 30th International Conference on Machine Learning*. International Machine Learning Society (IMLS), 2013, pp. 115–123.

[87] W. M. Czarnecki, S. Podlewska, and A. J. Bojarski, "Robust optimization of svm hyperparameters in the classification of bioactive compounds," *Journal of Cheminformatics*, vol. 7, no. 1, p. 38, 2015.

[88] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *Proceedings of the 27th International Conference on Machine Learning*. International Machine Learning Society (IMLS), 2010, pp. 1015–1022.

[89] A. D. Bull, "Convergence rates of efficient global optimization algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2879–2904, 2011.

[90] J. González, Z. Dai, P. Hennig, and N. Lawrence, "Batch bayesian optimization via local penalization," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, vol. 51. PMLR, 2016, pp. 648–657.

[91] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," 2015.

[92] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[93] R. M. Neal, *Bayesian learning for neural networks*. Springer, 1996.

[94] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, "Deep neural networks as gaussian processes," in *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018*. International Conference on Learning Representations, ICLR, 2018.

[95] J. Lee, L. Xiao, S. S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington, "Wide neural networks of any depth evolve as linear models under gradient descent," 2019.

[96] R. Novak, L. Xiao, J. Hron, J. Lee, A. A. Alemi, J. Sohl-Dickstein, and S. S. Schoenholz, "Neural tangents: Fast and easy infinite neural networks in python," 2019.

[97] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.

[98] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[99] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," 2015.

[100] L. V. Jospin, W. Buntine, F. Boussaid, H. Laga, and M. Bennamoun, "Hands-on bayesian neural networks – a tutorial for deep learning users," 2020.

[101] R. Ghanem, H. Owhadi, and D. Higdon, *Handbook of uncertainty quantification*. Springer, 2017.

[102] A. Saltelli, K. Aleksankina, W. Becker, P. Fennell, F. Ferretti, N. Holst, S. Li, and Q. Wu, "Why so many published sensitivity analyses are false: A systematic review of sensitivity analysis practices," *Environmental Modelling and Software*, vol. 114, pp. 29–39, 2019.

[103] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

[104] G. E. P. Box, J. S. Hunter, and W. G. Hunter, *Statistics for experimenters : design, innovation, and discovery*. Wiley-Interscience, 2005.

[105] M. D. Morris, "Factorial sampling plans for preliminary computational experiments," *Technometrics*, vol. 33, no. 2, pp. 161–174, 1991.

[106] I. M. Sobol' and S. Kucherenko, "Derivative based global sensitivity measures and their link with global sensitivity indices," *Mathematics and Computers in Simulation*, vol. 79, no. 10, pp. 3009–3017, 2009.

[107] S. Da Veiga, "Global sensitivity analysis with dependence measures," *Journal of Statistical Computation and Simulation*, vol. 85, no. 7, pp. 1283–1305, 2015.

[108] I. M. Sobol, "On sensitivity estimation for nonlinear mathematical models," *Mathematical Models and Computer Simulations*, vol. 1, no. 4, pp. 407–414, 1993.

[109] C. Soize and R. Ghanem, "Physical systems with random uncertainties: chaos representations with arbitrary probability measure," *SIAM Journal on Scientific Computing*, vol. 26, no. 2, pp. 395–410, 2004.

[110] B. Sudret, "Global sensitivity analysis using polynomial chaos expansions," *Reliability engineering & system safety*, vol. 93, no. 7, pp. 964–979, 2008.

[111] ——, "Polynomial chaos expansions and stochastic finite element methods," in *Risk and reliability in geotechnical engineering*, K.-K. Phoon and J. Ching, Eds. CRC Press, 2015, ch. 6, pp. 265–300.

[112] L. L. Gratiet, S. Marelli, and B. Sudret, "Metamodel-based sensitivity analysis: polynomial chaos expansions and gaussian processes," in *Handbook of Uncertainty Quantification*, R. Ghanem, H. Owhadi, and D. Higdon, Eds. Springer, 2017, ch. 38, pp. 1289–1325.

[113] B. Sudret and Y. Caniou, "Analysis of covariance (ANCOVA) using polynomial chaos expansions," in *Proceedings of the 11th International Conference on Structural Safety and Reliability*. Taylor & Francis, 2013, pp. 3275–3281.

[114] G. Li, H. Rabitz, P. E. Yelvington, O. O. Oluwole, F. Bacon, C. E. Kolb, and J. Schoendorf, "Global sensitivity analysis for systems with independent and/or correlated inputs," *Journal of Physical Chemistry*, vol. 114, no. 19, pp. 6022–6032, 2010.

[115] Y. Caniou, "Global sensitivity analysis for nested and multiscale modelling," Ph.D. dissertation, Blaise Pascal University – Clermont II, 2012.

[116] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto, "Sensitivity analysis in practice: a guide to assessing scientific models," *Chichester, England*, 2004.

[117] S. Kucherenko, S. Tarantola, and P. Annoni, "Estimation of global sensitivity indices for models with dependent variables," *Computer Physics Communications*, vol. 183, no. 4, pp. 937–946, 2012.

[118] T. A. Mara, S. Tarantola, and P. Annoni, "Non-parametric methods for global sensitivity analysis of model output with dependent inputs," *Environmental Modelling and Software*, vol. 72, pp. 173–183, 2015.

[119] J.-P. Baudry, "Model selection for clustering. choosing the number of classes," Ph.D. dissertation, Univ. Paris-Sud., 2009, https://tel.archives-ouvertes.fr/tel-00461550/document (Accessed April 19, 2020).

[120] ——, "Estimation and model selection for model-based clustering with the conditional classification likelihood," *Electronic journal of statistics*, vol. 9, no. 1, pp. 1041–1077, 2015.

[121] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[122] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated completed likelihood," *IEEE transactions on pattern analysis and machine intelligence*, vol. 22, no. 7, pp. 719–725, 2000.

[123] L. Scrucca, M. Fop, T. B. Murphy, and A. E. Raftery, "mclust 5: clustering, classification and density estimation using gaussian finite mixture models," *The R journal*, vol. 8, no. 1, p. 289, 2016.

[124] C. F. Dormann, J. M. Calabrese, G. Guillera-Arroita, E. Matechou *et al.*, "Model averaging in ecology: a review of bayesian, information-theoretic, and tactical approaches for predictive inference," *Ecological Monographs*, vol. 88, no. 4, pp. 485–504, 2018.

[125] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[126] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*. Springer New York, 2009.

[127] D. Fletcher, *Model averaging*. Springer, 2018.

[128] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, "Bayesian model averaging: a tutorial," *Statistical science*, vol. 14, no. 4, pp. 382–417, 1999.

[129] M. H. Faber, *Statistics and Probability Theory: In Pursuit of Engineering Decision Support*. Springer, 2012.

[130] ——, "On the governance of global and catastrophic risks," *International Journal of Risk Assessment and Management*, vol. 15, no. 5-6, pp. 400–416, 2011.

[131] European Committee for Standardization (CEN), "Petroleum and natural gas industries - Specific requirements for offshore structures - Part 1: Metocean design and operating considerations," *EN ISO 19901-1:2015*, 2015.

[132] ——, "Petroleum and natural gas industries - Fixed steel offshore structures," *EN ISO 19902:2008*, 2008.

[133] Y. Goda, *Random seas and design of maritime structures*. University of Tokyo press, 1985.

[134] G. Forristall, "Wave crest distributions: Observations and second-order theory," *Journal of Physical Oceanography*, vol. 30, no. 8, pp. 1931–1943, 2000.

[135] E. M. Bitner-Gregersen, K. C. Ewans, and M. C. Johnson, "Some uncertainties associated with wind and wave description and their importance for engineering applications," *Ocean Engineering*, vol. 86, pp. 11–25, 2014.

[136] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[137] T. P. Minka, "A family of algorithms for approximate bayesian inference," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[138] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning : with applications in R*. Springer, 2013.

[139] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794. [Online]. Available: http://dx.doi.org/10.1145/2939672.2939785

PART II

PAPERS

PAPER A

A FRAMEWORK FOR OFFSHORE LOAD ENVIRONMENT MODELING

Sebastian T. Glavind, and Michael H. Faber

The paper has been published in the
Proceedings of the ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering (OMAE2018), OMAE2018-77674, 2018.

The layout has been revised.

ABSTRACT

In the present paper, we propose a novel decision analytical framework for systems modeling in the context of risk-informed integrity management of offshore facilities. Our focus concerns the development of system models representing environmental loads associated with storm events. Appreciating that system models in general serve to facilitate the optimal ranking of decision alternatives, we formulate the problem of systems modeling as an optimization problem to be solved jointly with the ranking of decision alternatives. Taking offset in recent developments in structure learning and Bayesian regression techniques, a generic approach for the modeling of environmental loads is established, which accommodates for a joint utilization of phenomenological understanding and knowledge contained in databases of observations. In this manner, we provide a framework and corresponding techniques supporting the combination of bottom-up and top-down modeling. Moreover, since phenomenological understanding as well as analysis of databases may lead to the identification of several competing system models, we include these in the formulation of the optimization problem. The proposed framework and utilized techniques are illustrated on a principle example. The example considers systems modeling and decision optimization in the context of possible evacuation of an offshore facility in the face of an emerging storm event.

***Keywords**: ocean waves and associated statistics, structural safety and risk analysis, system integrity assessment.*

1 INTRODUCTION

In the context of the newly established Danish Hydrocarbon Research and Technology Centre (DHRTC), major initiatives have been launched to identify new, safe, and more efficient frameworks and approaches to facilitate optimization of assets integrity management decisions. The present study shall be seen as an early report on one of these activities, where focus is directed on how the rationale for the development of knowledge concerning the offshore load environment may be improved. In particular, we asses two avenues for improving probabilistic engineering modeling in support of decision-making, namely modeling basis and model representation.

1.1 MODELING BASIS

Traditional models are most often based on a phenomenological understanding, e.g., probabilistic physics model formulations with parameters estimated based on statistical evidence achieved through observations and experiments ("bottom-up" approaches). It is evident that such approaches rely strongly on the adequacy of a-priori available knowledge and information, which is

not always granted. As a result, it is generally the case that all focus of the modeling is directed on what is understood to be the most likely physical formulation of the phenomena of interest, and other possible explanations are implicitly excluded. Moreover, possible variables not realized to affect the phenomena of interest are systematically omitted in the modeling, which in turn increases the uncertainty associated with the derived models. In recent years robust so-called data-driven modeling approaches ("top-down" approaches) have been formulated and increasingly applied with success in a wide range of applications. Data-driven approaches facilitate that models are derived directly from data contained in e.g., databases and do not necessitate a prior understanding of the phenomena generating the data. Data-driven approaches generally identify the most likely relationship between covariates and observations and facilitates a quantification of this likelihood. The downside of data-driven approaches is, however, that they may indeed result in models contradicting the available knowledge.

One objective of the present research is thus to assess whether a combination of bottom-up and top-down modeling may be formulated, which facilitates a consistent utilization of prior phenomenological knowledge, and knowledge extracted from information contained in databases. Moreover, this formulation should accommodate that in principle all possible and relevant likely models may be identified and quantified with respect to their likelihood.

1.2 MODEL REPRESENTATION

Engineering modeling, e.g., in the context of assets integrity management, is traditionally undertaken by interfacing domain specific models, established individually by subject-matter experts. The domain specific models (e.g., models of the wave environment, water particle kinematics, hydraulic forces, structural responses, and failure criteria) are generally developed in accordance with the best available knowledge within the relevant domains of expertise, and they are optimized individually to provide the highest degree of precision with available and achievable information. The decisions regarding how to optimize precision are generally based on the prior understanding of the domain experts and often assessed without specific consideration of the context in which the models are applied. One example of this approach is development of a so-called "digital twin" model of a structure, where a numerical structural model is adopted to information collected using techniques of structural health monitoring. Such approaches surely provide a basis for supporting decisions; however, they neither facilitate for a context-driven optimization of the individual models nor a joint optimization of the interfaced models. As a result, the models may be unnecessarily precise in domains, which are not important for the decision context and not adequately precise

in domains of special importance for the decision context.

Another objective of the present research is thus, under consideration of the findings related to the first objective, to establish a theoretical and methodical basis for engineering modeling, which facilitates for integrating the optimization of model representation directly into the decision context, through an assessment of how the precision of the modeling affects the ranking of the considered decision alternatives.

2 THE NORTH SEA SYSTEM

The North Sea is a shallow shelf sea of the Atlantic Ocean on the European continental shelf. It is bounded by the northern and central European mainland to the east and south, including Norway, Denmark, Germany, the Netherlands, Belgium, and France. The East coast of Great Britain, and the Orkney and Shetland islands constitute the western border, and above the Shetland Islands, in the north, the North Sea connects with the Norwegian Sea. Moreover, in the east, the North Sea connects with the Baltic Sea via Skagerrak and Kattegat, and in the southwest, it connects with the Atlantic Ocean through the English Channel, cf. Fig. A.1. This geographical area is located in the mid-latitude cell, i.e., between 30- and 60-degrees northern latitude, which is characterized by temperate climate and prevailing westerly winds, the so-called westerlies.

In the North Sea region, extremes of winds and waves usually occur during the passage of a depression. A depression is an area of low atmospheric pressure and cyclonic airflow, which has a counter-clockwise rotation in the Northern hemisphere due to the Coriolis effect. Depressions vary from stormy and intense, which is characterized by a large area of strong winds; to nebulous, which is cloudy or hazy areas with light winds. The most common type of depression in the North Sea area is the frontal depression, i.e., depressions and associated frontal systems crossing an area normally from west to east. Another type of depression experienced in this region is the polar low, also termed cold air depression. Such depressions do not have fronts and are generally less intense than frontal depressions [1].

The sea bottom topology varies from a mean water depth of 200 m between the Shetland islands and Norway to 50 m between the Dogger Banks and northern Denmark, and 20 m off the Dutch-German coast, with an overall mean water depth of 80 m. This topology influences the system of eigenoscillations, and thus the resonance to tidal forcing, as well as surface level rise during storm surges [2]. The tides are semi-diurnal, with two high and two low tides per day. The larges tidal range occur on the east coast of Great Britain and in the English Channel. The highest storm surges are experienced in the shallow southern part of the North Sea, usually subject to northerly

Figure A.1: The North Sea.[1]

winds, but storm surges in general have a significant impact on areas with a large tidal range [1, 2].

The current velocity at a given location is composed of tidal currents and residual currents. Tidal current result from astronomical forcing, while the components of the residual current include circulation, storm-generated currents, as well as short- and long-period currents generated by various phenomena, such as density gradients, wind stress, and internal waves. Especially in the southern part of the North Sea with shallow waters and narrow passages, tides and surges may be associated with strong currents [1, 2].

The nature of wind-driven waves varies according to the generating winds, the water depth, and the fetch over which they were generated. Where fetch is restricted, which is the case for shelf seas, storm waves tend to be shorter, steeper, and lower than ocean waves. Oceanic areas are also subject to swell waves, which are waves that have moved out of the area in which they were generated. Swell waves can penetrate to semi-enclosed areas like the North Sea, where significant waves may be experienced without strong winds [1].

[1]Source: `https://commons.wikimedia.org/wiki/File:North_Sea_map-en.png`.

3 DESIGN OF OFFSHORE STRUCTURES

Offshore environmental loads are loads caused by the environmental phenomena at sea, i.e., wind, waves, currents, tides, earthquakes, temperature, ice, seabed movement, and marine growth. The response of an offshore structure to environmental loads depends, among others, on its dynamic response characteristics. Structures with significant dynamic response, e.g., with natural periods around the wave frequencies or their second order components, require an analysis of wave energy spectra or time series of the surface elevation, whereas it may be sufficient to use individual periodic waves for structures that only respond in a quasi-static manner [1, 3].

Dependent on the design checks needed for a given offshore structure, different types of metocean information are required, including:

- *Extreme and abnormal sea state parameters*. These parameters are used to define joint environmental actions for design checks in relation to the ultimate limit state (ULS) and accidental limit state (ALS).
- *Long-term probability distribution of sea state parameters*. This probability distribution is used to define environmental actions for design checks in relation to the fatigue limit state (FLS).
- *Long-term time series of sea state parameters*. These time series are required for response-based assessments of offshore structures.
- *Short-term description of environmental conditions*. This description is needed to perform checks in relation to the serviceability limit state (SLS), as well as short-term offshore operations [1].

Table A.1 summarizes some design considerations related to the different design checks.

4 METOCEAN INFORMATION

The metocean parameters and their (joint) probability distributions for a given location are usually specified by use of a metocean database. A metocean database may be established by monitoring sea state variables, such as significant wave heights, zero crossing periods, wave directions etc., over a period of years and/or by hindcasting of historical events. If the database is established based on numerical models, it is important that the simulated results are calibrated against appropriate measurements. Moreover, the metocean database should be sufficiently long to encompass all physical processes, which may be encountered during the service life of the structure [1].

Table A.1: Design considerations.

Design considerations related to ULS/ALS. ISO specifies three approaches for defining extreme actions and action effects: (i) Specify return period wave height, wind speed and current velocity, all determined by extrapolation of the individual variables considered individually. (ii) Specify return period of primary variable, i.e., the variable driving the response, and associated wave, wind, and current characteristics. (iii) Perform a response-based analysis based on any reasonable combination of wind, wave and current that result in; a significant global response of the structure, i.e., base share or overturning moment, with a specified return period; or a global extreme environmental action on the structure with a specified return period [1].
Design considerations related to FLS. Fatigue is an accumulation of damage caused by the repeated application of time-varying stresses, which in offshore structures are due to time-varying actions caused by waves, currents, gust winds, or a combination of these. In design assessments of fixed steel jacket structures, wave loading is usually regarded as the primary load effect. A FLS assessment of a structure requires that we specify all environmental conditions that are expected to occur during its period of exposure, i.e., its construction phase, including transport, and its design service lift [1, 3].
Design considerations related to SLS and short-term operations. Most offshore operations are sensitive to the displacement and vibration level of certain structural elements, thus these levels most be verified against acceptable limits for operations on a structure. An approach, which is typically employed in e.g., planning of maintenance operations, is the persistence analysis, or weather-window analysis, where upper bounds are specified for a set of environmental parameters, while the maintenance operations are conducted [1, 4].

Offshore metocean monitoring systems can vary from simple weather stations to complete data acquisition systems, including a range of sensors as well as signal processing, display, storage, and transmission features. These systems play an important role in ensuring safe offshore operations and structural integrity by providing real-time information for operational use and long-term records for engineering purposes [1].

Apart from measurements and simulations, we may gain valuable information from experimental results. Especially, wave experiments are an important source of information due to the fact that we do not fully understand wave phenomena, such as formation, transformation and breaking, as well as wave-wave and wave-structure interactions [5]. Thus, model tests also play an important role in design assessments of offshore structures.

4.1 METOCEAN DATABASE

The wave environment is usually sampled as sea states represented by a given significant wave height, period, direction, and directional spreading. These average quantities are calculated at a defined averaging interval, which varies from 1 h to 3 h. Wave data are recorded by in-situ instruments (buoys, wave staffs, radars, lasers, LASAR (array of lasers) and step gauges) and remote sensing technique (satellites and aircrafts). Note that buoy measurements can be used to estimate integral properties of a wave field, but crests heights measured by buoys are generally smaller than those measured by other sensors [1, 6, 7].

Current speed and direction are measured at a number of depths throughout the water column. Ocean currents should be measured at minimum three depth (or bins), i.e., near-surface, mid-depth and near-bottom, but in deep waters more measurements are usually needed to capture the current profile [1]. The mean speed and direction should be recorded at least once per hour. Current data are recorded by local instruments (drifters and current meters) and remote sensing techniques (satellites) [8].

Natural winds are defined by two components: mean or sustained wind speed, and gust wind speed. The gust component is generated by the turbulence of the flow field, whereby it has components in all three spatial directions. Mean wind speed and direction are specified at a defined averaging interval, typically 10 min to 3 h, and reference elevation, typically 10 m to 20 m above mean sea level (MSL) [1, 3]. In-situ instruments (buoys, ships, and platforms) and remote sensing technique (satellites and aircrafts) collect wind data [7].

The water level at a given site consists of a more or less stationary component, which is referenced to as chart datum, usually MSL, and variations with time relative to this level, i.e., residual-water-level. The residual variations are due to astronomical tides, and wind and atmospheric pressure, which can lead to storm surges. Variations in MSL are due to long-term climate effects and sea floor subsidence [1]. Local water depth measurements may be conducted in a variety of ways, one approach is to use an echo-based technique (e.g., ultrasonic instruments) [9]. Remote sensing techniques (satellites) also produce bathymetric and residual-water-level data [10].

Other relevant parameters that are usually contained in a metocean database include sea ice and icing, water and air temperature, air pressure, water salinity, etc..

4.2 METOCEAN EXPERIMENTS

In some cases, design assessments of offshore structures need to be addressed by mean of hydrodynamic model tests. As previously emphasized, this is the

case when buoy measurements of the wave environment are available for a specific site, and the crest height distribution is needed.

The behavior of surface waves is studied experimentally in wave tanks. If the length and width of the tank is of comparable size, the tank is called a wave basin; and if the length of the tank is significantly larger than its width, the tank is called a wave flume. Wave basin tests are usually conducted to study three-dimensional effects in relation to undisturbed wave fields as well as loading on and response of scaled structure models, when exposed to specific wave environments. The same phenomena are studied in a wave flume tests but typically in a two-dimensional setting.

The fundamental premise of model testing is that the real phenomenon being studied is emulated to a satisfactory degree by the model test. When structural responses are considered, three general categories of similarity can be defined: Geometric shape, kinematics of various motions, and dynamic forcing. Geometric similarity reflects that all physical dimensions are scaled down in the model by a certain ratio or ratios compared to the prototype. Kinematic similarity requires that the model velocity and acceleration of the various bodies and the fluid are proportional to those of the prototype. Dynamic similarity reflects that the dynamic forces in the model reflects the dynamic forces in the prototype with the same scale ratio [11].

Other relevant categories of model tests in relation to the offshore environment are current and tide experiments, wind tunnel tests, and different hybrid tests, where multiple phenomena are studied simultaneously in one experimental setup. For instance, air-water interaction effects may be studied in a wind-wave flume. Additionally, physical model tests may be supplemented by numerical simulations, which then lead to considerations regarding the transformation from the computational environment to the model or prototype environment.

5 BASIC CONSIDERATIONS ON MODEL BUILDING

Scientific models are established on the premise that they serve decision-making in the generic context of systems performance management, noting that systems may comprise any combination of interactions between applied technology, humans, organizations, and the natural environment. In this context, the objective of model building is to represent the available and relevant knowledge about the performances and/or characteristics of systems in consistency with scientific knowledge and evidence obtained from e.g., experiments and observations. In the following, to represent knowledge and rank decision alternatives, Bayesian probability theory and Bayesian decision anal-

ysis are applied, see e.g., [12]. Moreover, the framework of Bayesian networks is utilized to represent joint dependence relations in the problem domain, cf. next section.

5.1 SYSTEMS AND DECISION-MAKING

At the simplest level, a model $\mathcal{M}(a)$ provides a relationship between input and output, measured in terms of utility, conditional on a decision represented by a. Figure A.2a illustrates how a system provides this relationship between decision alternatives $\boldsymbol{a}$ and the associated utilities $\mathbb{U}(\boldsymbol{a})$. The system performance is generally associated with uncertainty, thus the performance (output or utility) is random. In accordance with Bayesian decision theory [12] and the axioms of utility theory [13], the optimal decision alternative is selected from $\boldsymbol{a}$ by optimizing the expected utility, i.e., $a^* = \arg\max_a(\mathbb{E}[\mathbb{U}(a)])$.

In the general case [14, 15], the system under consideration is unknown in itself, and it is unknown which is the most relevant representation of the system. In Fig. A.2b, the variable s represents one choice of system representation out of a set of system representations $\boldsymbol{s}$, and σ represents a realization of the real system. The optimization of decision alternatives is further complicated by the fact that some of the decision alternatives within $\boldsymbol{a}$ are only relevant for one or some of the competing system representations. The optimization of decision alternatives must thus be undertaken jointly with a choice of system representation.

To account for the competing system representations, we introduce the system model $\boldsymbol{\mathcal{M}}(a)$:

$$\boldsymbol{\mathcal{M}}(a) = (\Sigma(a), C(a), \boldsymbol{X}(a))^T, \tag{A.1}$$

where $\Sigma(a)$ is a probabilistic system representation with realizations $\{\sigma_j\}_{j=1}^{n_s}$ corresponding to the set of system choices. Each system representation is comprised of an ensemble of n_{c_j} constituents interacting jointly to provide the functionalities of the system, i.e., mapping input to output. For a given choice of system s, the performances of the constituents are modeled by a set of constituent models C, and a prior probabilistic representation $P'(\boldsymbol{X}|s)$ of all variables entering the model. For the sake of generality, we highlight that in principle all the models defining the system have temporal and spatial references; these are omitted here to simplify notation.

The optimization of decision alternatives, including system choice, may

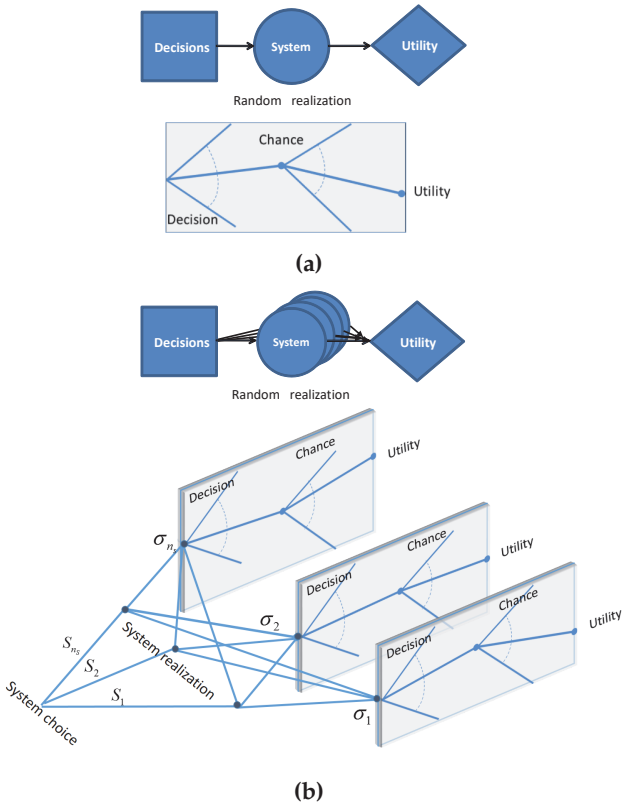


Figure A.2: Systems role in decision analysis: (a) one possible system, (b) several possible systems.

now be written as

$$(s^*, a^*) = \arg\max_{s,a} \mathbb{U}(s,a) = \arg\max_{s} \left(P(\Sigma = s) \arg\max_{a} \left(\mathbb{E}'_{\boldsymbol{X}|s}[\mathbb{U}(a,\boldsymbol{X})] \right) + \mathbb{E}'_{\Sigma\setminus s} \left[\mathbb{E}'_{\boldsymbol{X}|\{\Sigma\setminus s\}}[\mathbb{U}(a^*,\boldsymbol{X})] \right] \right), \tag{A.2}$$

where $a^* = \arg\max_a \mathbb{E}'_{\boldsymbol{X}|s}[\mathbb{U}(a,\boldsymbol{X})]$, see also [15]. In Eq. A.2, the robustness of the decision with regard to the choice of system may be assessed as the ratio of the first term to the sum of the two terms. This ratio, which will take values between 0 and 1 (1=robust), indicates how sensitive the decision is with regard to the possibility that the optimization is undertaken under an erroneous system assumption.

Furthermore, as indicated earlier, we note that model building should be seen as an integrated part of the decision optimization. There is no need

for a model to be accurate in the domains of reality which are irrelevant for the decisions subject to optimization. On the contrary, by embedding the model building operation inside the optimization of decision alternatives, the available knowledge may be fully utilized to optimize the utility associated with the system under consideration, and thus consistently rank decision alternatives.

5.2 INTERPRETATION OF AND REQUIREMENTS TO SYSTEM MODELS

The available approaches for modeling the performance of systems may be categorized as classical engineering understanding based bottom-up models and data-driven top-down models. However, in either case, evidence can and must be accounted for in the modeling process. As outlined in the foregoing, a model is a representation of reality in a context of decision-making, meaning that a good model facilitates consistent ranking of the considered decision alternatives.

In recent developments on data-driven modeling and data-driven learning, the perspective is often taken that such approaches are superior to bottom-up modeling approaches, since they simply reflect the information contained in the evidence. However, it must be appreciated that reality is fundamentally subjective and should be understood as a proxy for truth, to the extent that this (the truth) is objectively understood. Reality is thus associated with uncertainty but may be framed through experience and information (knowledge), i.e., a combination of philosophical and scientific insights, and observations. Framing of reality is thus fundamentally subjective, since it is based on a choice of which experience, which information (data), and which class of models are used as the modeling basis.

The implication of this is that whether bottom-up or top-down approaches, or combinations hereof, are utilized as basis for modeling of systems performances, the models will always be subjective and thus influenced by epistemic uncertainties. A framework for systems modeling from [16] in the context of assets integrity management is illustrated in Fig. A.3.

In Fig. A.3, the concept of indicators is introduced as a means to account for evidence, which is indirectly, and generally more weakly, related to the performances of the system. As an example, an indicator of a short-term maximum crest height could be the significant wave height. Observations of indicators thus provide information; however, they are in general subject to additional uncertainty. The concept of indicators provides a strong means for including evidence in systems modeling, and they may further be used to facilitate multi-scale systems representations. This principle of introducing evidence achieved through observation of indicators is illustrated in Fig. A.4.

We emphasize that probabilistic system models must consistently account

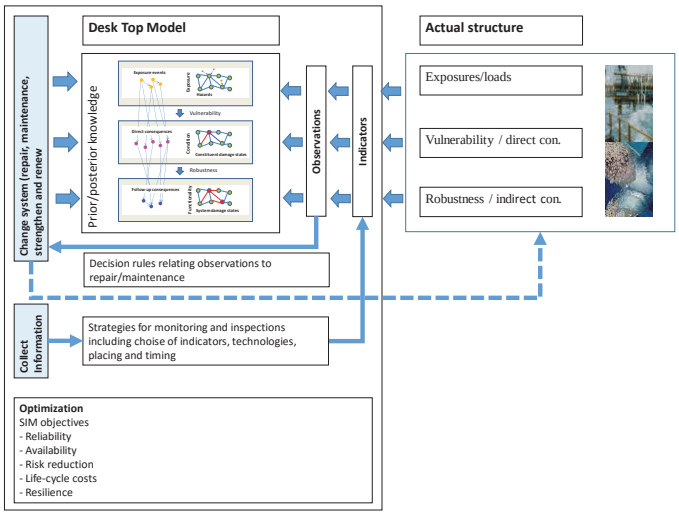


Figure A.3: Systems modeling framework in the context of offshore asset integrity management.

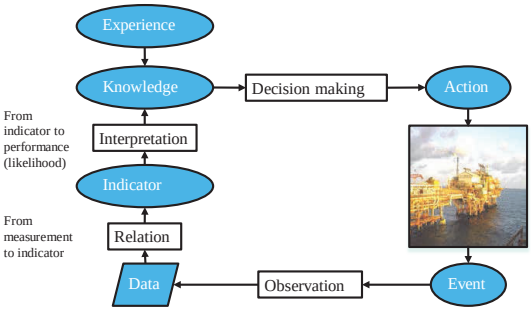


Figure A.4: Illustration of the concept of indicators as applied in Fig. A.3.

for and distinguish between uncertainty associated with sparsity of evidence, and possible model uncertainty and associated lack of fit. This is of crucial importance in the context of model optimization, where an optimal trade-off between complexity (in terms of representation, constituent, and parameter models), and the associated statistical uncertainties must be identified.

In summary, with reference to Eqs. A.1 and A.2, probabilistic system models should facilitate:

- Representation of multiple possible competing systems and associated likelihoods.
- Inclusion of probabilistic constituent models (including aleatory and epis-

temic uncertainty).

- Probabilistic descriptions of the parameters of the constituent models (including epistemic and aleatory uncertainty).
- Inclusion of evidence obtained from experiments on and observations (including indicators) of the system.
- Consistent representation of statistical uncertainties due to sparsity of evidence.

6 BAYESIAN NETWORKS

Bayesian networks (BNs), which constitute a branch of probabilistic graphical models (PGMs), encode a joint distribution over a set of random variables $\boldsymbol{X}$ by decomposing it into a product of local, conditional probability distributions according to a directed acyclic graph (DAG) $\mathcal{G}$.

In the graph structure $\mathcal{G}$, each vertex $v_i \in \boldsymbol{V}$ corresponds to a random variable X_i, and the edges $\boldsymbol{E}$ between the vertices represent a set of conditional dependence relations implied by $\mathcal{G}$. Moreover, by studying the missing edges in $\mathcal{G}$, we can directly read off a set of conditional independence relations between the random variables. For each random variable X_i in $\mathcal{G}$, we specify a conditional probability distribution $P(X_i|\mathbf{Pa}_i)$, which defines the dependence of X_i on the random variables, which X_i is conditional dependent on in $\mathcal{G}$, termed the parent set $\mathbf{Pa}_i$ of variable X_i. The joint distribution encoded by a BN is shown in Eq. A.3.

$$P(\boldsymbol{X}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}) = \prod_i P(X_i|\mathbf{Pa}_i), \tag{A.3}$$

where $\boldsymbol{\Theta}_{\mathcal{G}}$ denotes the set of model parameters. For discrete variables, the set of parameters correspond to the probability masses of each combination of states: $\boldsymbol{\Theta}_{\mathcal{G}} = \bigcup\{P(x_i|\mathbf{pa}_i) = \Theta_{x_i|\mathbf{pa}_i}\}$. For continuous variables, the parameter set correspond to the parameters needed to specify the probability density functions of the random variables.

A classic example of a BN from [17] is shown in Fig. A.5. This network shows how Mr. Holmes reasons about his burglary alarm A going off. If the alarm goes off, his neighbour Dr. Watson W may call him, and the triggering of the alarm will have one of two causes: (i) there is a burglar B in his house, or (ii) there is an earthquake E in the area. Moreover, he may gain additional information on the earthquake scenario by listening to the radio news R. The joint distribution, which factorizes according to Fig. A.5, is written:

$$P(B, E, A, R, W) = P(B)P(E)P(A|B, E)P(R|E)P(W|A).$$

Now, imagine that Holmes gets a call from Watson about his alarm going off. Holmes rushes to his car believing that a burglar has triggered the alarm. On his way home, the radio news reports an earthquake in the area. This additional piece of information makes him change his belief in the burglary scenario, as the reported earthquake "explain away" the triggered alarm.

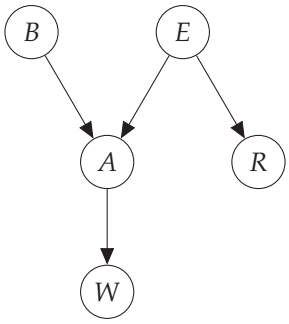


Figure A.5: Burglary or earthquake Bayesian network.

6.1 LEARNING BNs

As apparent from Eq. A.3, a BN is fully specified by its graph $\mathcal{G}$ and its parameters $\boldsymbol{\Theta}_{\mathcal{G}}$. The process of specifying the pair $\{\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}\}$ is termed learning, and it is usually performed in two steps: structure learning and parameter learning. Structure learning refers to the construction of the graph structure $\mathcal{G}$, and parameter learning refers to the specification of the model parameters $\boldsymbol{\Theta}_{\mathcal{G}}$.

Both learning tasks may be undertaken by use of a bottom-up or top-down approach, or by a combining hereof. In a top-down approach, the graph structure and parameters are established using information provided in a database. In a bottom-up approach, domain experts are interviewed to identify the graph structure and parameters [18]. As mentioned, BNs are defined in terms of conditional dependence relations and probabilistic properties, without any implication that edges should point from causes to effects. However, it is argued by Pearl that causal BNs pose a more reliable and natural way of expressing our knowledge about the domain we are modeling [19]. That is, we should strive to use a combined learning approach whenever possible, as it makes the best use of the available and relevant knowledge about a given system.

In this paper, we only present a concise description of BNs. The interested reader may refer to the seminal work by Pearl [17, 19], as well as recent prominent textbooks [20–22].

7 A PRINCIPLE EXAMPLE

In order to illustrate the principles and methods introduced in this paper, a principle example is given in this section. We will consider the following decision problem: The facility manager of an offshore facility is informed that a storm is approaching from a westerly direction. She knows that the facility will fail if the storm peak significant wave height H_{m0} exceeds 6 m, and she is now faced with a decision of whether or not to evacuate the facility. She does not know which westerly direction the storm is approaching from, but if it approaches from SW or W, she can choose to either evacuate by helicopter or boat. If the storm approaches from NW, she can only choose to evacuate by helicopter. Furthermore, dependent on the direction from which the storm is approaching, failure will have different consequences.

The decision problem is outlined in Fig. A.6. In the figure, $\{s_j\}_{j=1}^3$ corresponds to the choice of storm model, i.e., SW, W, and NW; and $\{a_k\}_{k=1}^3$ corresponds to the decision alternatives, i.e., evacuate by helicopter, no evacuation, and evacuate by boat. Please note that the combination s_3 and a_3 is not possible, as she cannot evacuate by boat, if the storm is approaching from NW. Moreover, the system dependent probability of failure is defined as $p_f = P(H_{m0} > 6\,\mathrm{m}|s)$, the maximum cost $U_{max} = -1$ monetary units, and $P(\sigma_j)$ represents the probability of storm direction j.

Now, we want to help the facility manager to make an optimal decision,

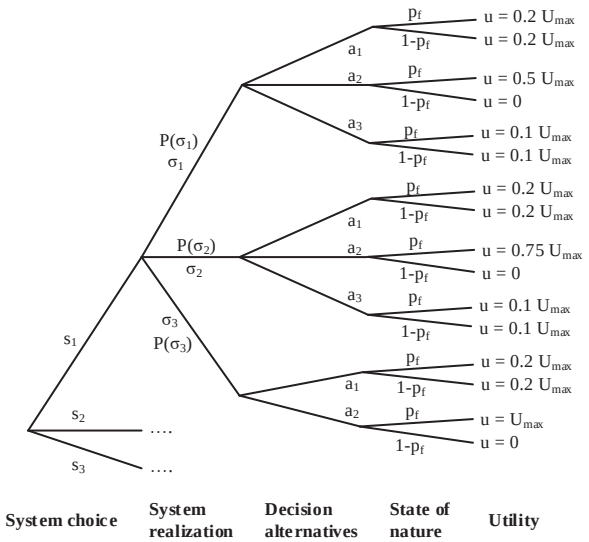


Figure A.6: Illustration of the decision problem.

thus, based on a metocean database (see below), we build a probabilistic model (BN) for the joint distribution of the set of environmental variables in the database. This model may then be used to estimate the probabilities in Fig. A.6.

7.1 DATABASE

In this study, we consider simulated storm data for several locations (platforms) in the North Sea. The data are produced by the Danish Hydraulic Institute (DHI) by use of their spectral wave simulator. The database contains observations of environmental variables at 23 platforms over 2187 storms, corresponding to about 30 years of data measurements. The variables included in the database appear in Tab. A.2.

Table A.2: Metocean database.

Variable	Description	Unit
Lng	Longitude	°
Lat	Latitude	°
Dpt	Water depth	m
WiS	Maximum storm wind speed	m/s
CS	Current speed	m/s
RWL	Residual water level (surge + tide)	m
TY	Time of storm peak	°
WaD	Peak wave direction	°
WiD	Direction of max. storm wind speed	°
CD	Current direction	°
*Hm*0	Storm peak significant wave height	m

7.2 DISCRETIZATION

An important prepossessing consideration, when applying BNs to a real-world domain, is how to handle continuous variables. To avoid distributional assumptions, continuous variables are discretized in this study. In this regard, the number of intervals and their boundaries have to be chosen carefully, as valuable information about the distribution of the variables and their dependency may be lost otherwise. For some of the variables, we predefine the discretization boundaries. This is the case for all directional variables, as well as for the time of storm peak and for the wave height variable, as we need a specific granularity of these variables in regard to the decision problem.

The remaining variables are discretized by use of a multivariate discretization procedure, embedded in the structure learning procedure, which takes

the interactions in the graph structure into account. The method we use is based on [23], where it is assumed that a data set is generated in two steps: First, an interval of a variable is selected from the distribution of the discrete variable. Second, the corresponding continuous value is drawn from a uniform distribution over the interval. We then seek an optimal discrete representation $\mathcal{D}$ of the original continuous data set $\mathcal{D}^c$, which maximizes the objective function: $P(\mathcal{D}|\mathcal{G})P(\mathcal{D}^c|\mathcal{D})$.

As the graph structure changes throughout the structure learning phase, the discretization is adjusted dynamically to maximize the objective function in a manner similar to that proposed in [24]. That is; first, the data are discretized without taking any interaction between the variables into account (this corresponds to an initial empty graph); second, this discretization is used to learn a BN. These two steps are repeated until the objective function converges to a local optimum. A similar scheme for combined structure learning and discretization is used in [25].

In our implementation, we use functionalities from the publicly available R package *bnlearn* [26] to learn the structure of the graph.

7.3 RESULTS AND CONCLUSIONS

Initially, the graph structure and optimal discretization are learned using the database. In this regard, the structure learning is constrained by our causal understanding of the domain, e.g., we force an edge going from *TY* to *WiS*, *WiD*, *Hm*0, and *RWL*, as well as edges meeting at *Hm*0 from *WiD*, and *WiS*; see Tab. A.2 for a reference on the variable names. The learned graph structure and corresponding discretization appear in Fig. A.7 and Tab. A.3, respectively.

Apart from the predefined connections, we observe that the BN encodes an additional set of statistical dependencies. For instance, as we would expect, there is a statistical dependence between the water depth at a location and the current speed in a storm, and between the maximum wind speed in a storm and the wind direction. Furthermore, if we consider the discretization, we see that the optimal discretization of the location variables *Lng* and *Lat*, and the depth variable *Dpt* is binary.

By used of the BN model, the probabilities in Fig. A.6 are estimated, and subsequently the decision problem is solved by use of Eq. A.2. The solution is shown in Fig. A.8, where the optimal representation and decision alternative, given representation, are indicated by bold-faced boxes. It appears that system representations s_1 and s_3 both optimize the expected utility. They receive the same expected value of utility, because they agree on the optimal action a^* being not to evacuate.

Based on the example it may be concluded that the formulated approach to systems modeling, which combine phenomenological knowledge with

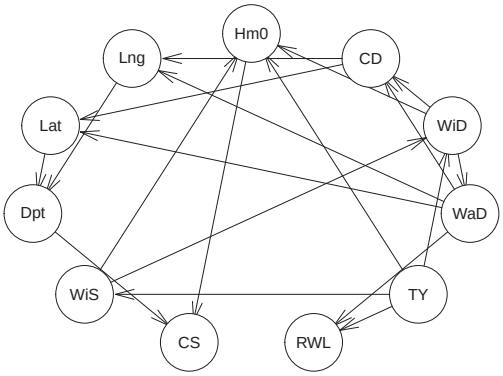


Figure A.7: BN model.

Table A.3: Discretization.

Variable	Levels	Comments
Lng	2	learned
Lat	2	learned
Dpt	2	learned
WiS	9	learned
CS	6	learned
RWL	9	learned
TY	4	predefined ($Spring$, $Summer$, $Fall$, $Winter$)
WaD	8	predefined (NW, N, NE, E, SE, S, SW, W)
WiD	8	predefined (NW, N, NE, E, SE, S, SW, W)
CD	8	predefined (NW, N, NE, E, SE, S, SW, W)
$Hm0$	16	predefined $((0,2], (2,2.5], (2.5,3], ..., (9, Inf])$

knowledge retrieved from databases of relevant information and embed the model building within a decision analytic framework, appears feasible and robust. Presently more studies are undertaken to understand in more detail the sensitivities of the derived models and decisions to various algorithmic choices involved in the modeling.

8 CONCLUSIONS

The present paper presents early developments on the formulation and implementation of a novel decision analytic framework for systems modeling in the context of risk-informed integrity management of offshore facilities, with a focus on the development of system models representing environmental loads associated with storm events. To account for the fact that system mod-

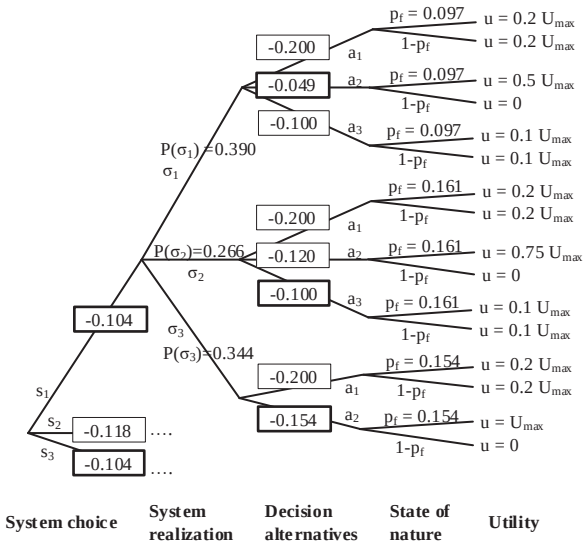


Figure A.8: Decision tree for the optimal system choice decision problem.

els in general serve to facilitate the optimal ranking of decision alternatives, we formulate the problem of systems modeling as an optimization problem to be solved jointly with the ranking of decision alternatives. Moreover, based on recent developments in structure learning and Bayesian regression techniques, a generic approach for the modeling of environmental loads is established, which accommodates for a joint utilization of phenomenological understanding and knowledge contained in databases of observations. The developed framework and corresponding techniques greatly support the combination of bottom-up and top-down modeling and facilitates for consistently addressing the existence of possible competing systems in the context of assets integrity management. The proposed framework and utilized techniques are illustrated on a principle example, where we consider systems modeling and decision optimization in the context of possible evacuation of an offshore facility in the face of emerging storm events. The example shows that the formulated modeling framework is indeed feasible, and future research will be directed on further algorithmic optimization as well as broader and more involved applications.

ACKNOWLEDGMENT

The authors acknowledge the funding received from Centre for Oil and Gas – DTU / Danish Hydrocarbon Research and Technology Centre (DHRTC). We

would also like to thank Maersk Oil and DHI for providing the data and support needed to conduct this research.

NOMENCLATURE

a **or** $\boldsymbol{a}$	Decision alternative(s).
s **of** $\boldsymbol{s}$	System representation or set of system representations.
x_i **or** $\boldsymbol{x}$	Realization of random variable(s).
$\mathbb{E}(\cdot)$	Expectation operator.
$\mathcal{M}(\cdot)$	System model.
$P(\cdot)$	Probability or (conditional) probability distribution.
$\mathbf{Pa}_i$	Parent set of variable X_i.
$\mathbb{U}(\cdot)$	Utility function.
X_i **or** $\boldsymbol{X}$	Random variable or set of random variables.
σ	System realization.
Σ	Probabilistic system representation.
Θ **or** $\boldsymbol{\Theta}$	Parameter or parameter vector.

REFERENCES

[1] European Committee for Standardization (CEN), "Petroleum and natural gas industries - Specific requirements for offshore structures - Part 1: Metocean design and operating considerations," *EN ISO 19901-1:2015*, 2015.

[2] J. Suendermann and T. Pohlmann, "A brief analysis of north sea physics," *Oceanologia*, vol. 53, no. 3, pp. 663–689, 2011.

[3] S. Chandrasekaran, *Dynamic Analysis and Design of Offshore Structures*. Springer India, 2015.

[4] European Committee for Standardization (CEN), "Petroleum and natural gas industries - Fixed steel offshore structures," *EN ISO 19902:2008*, 2008.

[5] Z. Liu and P. Frigaard, *Generation and Analysis of Random Waves*. Aalborg University, 1999.

[6] G. Forristall, "Wave crest distributions: Observations and second-order theory," *Journal of Physical Oceanography*, vol. 30, no. 8, pp. 1931–1943, 2000.

[7] E. M. Bitner-Gregersen, K. C. Ewans, and M. C. Johnson, "Some uncertainties associated with wind and wave description and their importance for engineering applications," *Ocean Engineering*, vol. 86, pp. 11–25, 2014.

[8] V. Klemas, "Remote sensing of coastal and ocean currents: An overview," *Journal of Coastal Research*, vol. 28, no. 3, pp. 576–586, 2012.

[9] S. E. Borujeni, "Ultrasonic underwater depth measurement," in *2002 International Symposium on Underwater Technology*, 2002, pp. 33–38.

[10] W. H. Smith and D. T. Sandwell, "Conventional bathymetry, bathymetry from space, and geodetic altimetry," *Oceanography*, vol. 17, no. 1, pp. 8–23, 2004.

[11] Y. Goda, *Random seas and design of maritime structures*. University of Tokyo press, 1985.

[12] H. Raiffa and R. Schlaifer, *Applied statistical decision theory*. MIT Press, 1961.

[13] J. von Neumann and O. Morgenstern, *Theory of games and economic behavior*. Princeton University Press, 1953.

[14] M. H. Faber and M. A. Maes, "Epistemic uncertainties and system choice in decision making," in *Ninth International Conference on Structural Safety and Reliability, ICOSSAR*, 2005, pp. 3519–3526.

[15] M. H. Faber, "On the governance of global and catastrophic risks," *International Journal of Risk Assessment and Management*, vol. 15, no. 5-6, pp. 400–416, 2011.

[16] ——, "Risk Informed Structural Systems Integrity Management: A Decision Analytical Perspective," in *36th International Conference on Ocean, Offshore and Arctic Engineering*, Trondheim, 2017, p. 62715.

[17] J. Pearl, *Probabilistic Reasoning in Intelligent Systems. Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[18] M. Scutari and J.-B. Denis, *Bayesian networks: with examples in R*. CRC press, 2014.

[19] J. Pearl, *Causality*. Cambridge university press, 2009.

[20] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.

[21] S. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Pearson Education Limited, 2014.

[22] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[23] S. Monti and G. F. Cooper, "A multivariate discretization method for learning bayesian networks from mixed data," in *Fourteenth International Conference on Uncertainty in Artificial Intelligence*, 1998, pp. 404–413.

[24] N. Friedman and M. Goldszmidt, "Discretizing continuous attributes while learning bayesian networks," in *Thirteenth International Conference on Machine Learning*, 1996, pp. 157–165.

[25] K. Vogel, "Applications of bayesian networks in natural hazard assessments," Ph.D. dissertation, University of Potsdam, 2013.

[26] M. Scutari, "Learning bayesian networks with the bnlearn R package," *Journal of Statistical Software*, vol. 35, no. 3, pp. 1–22, 2010.

PAPER B

SYSTEMS MODELING USING BIG DATA ANALYSIS TECHNIQUES AND EVIDENCE

Sebastian T. Glavind, Juan G. Sepulveda, Jianjun Qin,
and Michael H. Faber

The paper has been published in the
Proceedings of the IEEE 2019 4th International Conference on System Reliability and Safety (ICSRS2019), ICSRS2019-R0119, 2019.

The layout has been revised.

ABSTRACT

In the present contribution, the potentials of utilizing techniques of big data analysis as a means to improve the understanding of complex probabilistic system representations are investigated. It is assumed that a probabilistic model is available for the representation of the system performances and that an adequate Monte Carlo simulation technique is available and applied for the probabilistic analysis of these. Model-based cluster analysis is then applied to establish a visual representation of the Monte Carlo simulated scenarios of events leading to different performances of the considered system. Various conditioning events on the simulated scenarios, such as specific failure events, are readily introduced by sorting. Assuming that the Monte Carlo simulated scenarios of events are utilized to establish a surrogate representation of the considered system, variance-based sensitivities are derived for both the case of independent and dependent random variables. To this end, so-called ANOVA and the very recently formulated ANCOVA decompositions are applied. The proposed scheme is illustrated on a simple example in which the probabilistic characteristics of non-linear structural performances of a moment resisting frame structure are considered. It is seen from the example that big data techniques may readily be applied to provide significant insights on which scenarios of events govern the probabilistic characteristics of the performances of the system, and with respect to how uncertainties associated with the random variables used to model the system propagate in the system and affect its responses. The latter is especially useful when aiming to reduce model complexity, but also in the context of structural health monitoring where response characteristics that contain significant information about the state of the system must be identified.

***Keywords**: model-based clustering, sensitivity analysis, system performance assessment, Monte Carlo simulation.*

1 INTRODUCTION

The probabilistic representation of systems performances is a challenging but crucially important task in the context of engineering decision-making. Across different engineering application areas a large variety of different probabilistic approaches for the representation of systems performances have been developed, see e.g., [1–5], all aiming to provide information-consistent models of the systems performances, which govern the ranking of decisions for their design and management.

In many cases, the considered systems are rather complex and may e.g., comprise interconnected systems subject to uncertainties, which are represented by high dimensional vectors of causally and stochastically dependent random variables. Moreover, the considered systems may in general exhibit significant non-linear characteristics at different scales between demands act-

ing on and within the systems and their performances.

The output of analyses, based on the available probabilistic models, typically center around a relatively few key characteristics, such as the annual probability of complete or partial system failure and various types of systems damage events. This type of output may be seen to comprise the key information with respect to what ultimately drives the expected values of consequences associated with the performances of the systems. However, such relatively sparse extracts of information offer very little, if any, information and/or knowledge with respect to the characteristics of the event scenarios of the physical processes, which lead to different states of failures and damages.

The immediate results of probabilistic systems analyses therefore do not provide much insight on whether the developed and analyzed models behave physically meaningful, how uncertainties associated with the probabilistic modeling affect the probabilistic characteristics of systems performances, and how the systems performances may efficiently be improved by changing the physical characteristics of the system or by improving the knowledge about the system. The latter is especially relevant in a context where it is possible to observe, e.g., by means of monitoring, the performances of a system over time and to utilize the collected information as a means for improving the system model. Thus, there is a strong need to improve presently available techniques for sensitivity analysis of probabilistic representations of systems.

This challenge is taken up in the present contribution, where the potentials of utilizing techniques of big data analysis as a means to improve the understanding of complex probabilistic system representations are investigated. Our starting point is that a particular system is addressed for which a probabilistic model is available for the representation of the system performances. Furthermore, for the sake of simplicity, it is assumed that an adequate Monte Carlo simulation based technique is applied for the probabilistic analysis of these (see e.g., [6]). The overall scheme is illustrated in Fig. B.1, where it is indicated that all available information from the Monte Carlo simulations are gathered and stored in a database.

The basic approach followed in the present contribution is addressing the middle part of Fig. B.1, highlighted in Fig. B.2. Going from left to right in Fig. B.2, it is seen how a computer model of a real system is established in order to generate a database of response characteristics by means of Monte Carlo simulation, i.e., scenarios of events describing the responses of the considered system. The system responses addressed in the present study are associated with failure scenarios, but in principle any system response characteristic may be generated and analyzed in the same manner.

Next, as outlined in Sec. 2, the scenarios of events stored in the database are exposed to modern data mining tools, using model-based clustering based on multidimensional Gaussian mixtures. This facilitates derivation of joint parametric representations of the probabilistic characteristics of the sys-

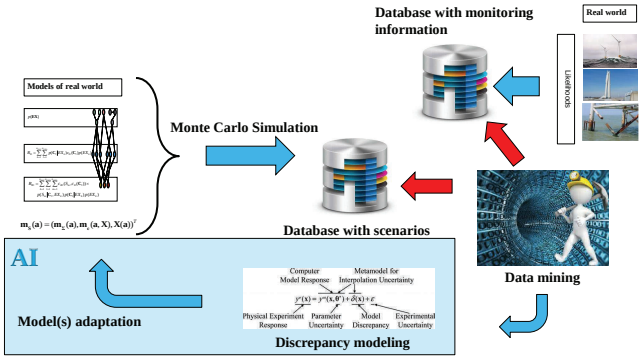


Figure B.1: Framework for system modeling, analysis, and updating.

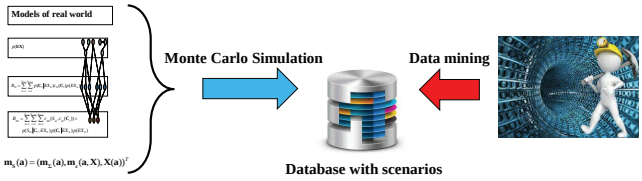


Figure B.2: Workflow of the present paper.

tem for given failure scenarios, i.e., $p(\boldsymbol{X})$. The cluster analyses provide significant information in themselves, as they reveal patterns in the simulation results, which are not otherwise observable, e.g., regions of jointly occurring realizations of random variables that dominate the contributions to probabilities of particular realizations of scenarios of system failures; commonly referred to as most likely failure points or design points in structural reliability theory. Moreover, cluster analysis may also efficiently reveal which physical and/or organizational characteristics of the system contribute to the robustness of the considered system.

Sensitivity analyses are then introduced to enhance the understanding of how uncertainties associated with the probabilistic modeling of the system affect the uncertainties associated with the performances of the system. Variance-based sensitivity analysis, as described in Sec. 3, may be conducted by application of the so-called ANOVA and ANCOVA decompositions for the case of independent and dependent input variables, respectively.

The proposed approaches are finally illustrated on an example in Sec. 4., where the probabilistic characteristics of non-linear structural performances of a simple moment resisting frame structure are considered. In the example, as a means to establish a representation of the probabilistic characteristics of the system responses, i.e., $y = f(\boldsymbol{X})$, a surrogate model for the considered response function is introduced using polynomial chaos expansions.

2 MODEL-BASED CLUSTERING

In cluster analysis, it is assumed that the considered data set $\mathcal{D}$ is sampled from a set of distinct base models, and the target of the analysis is to infer the most likely generating base model for each realization, i.e., the latent cluster assignment. In this regard, it is assumed that the data set $\mathcal{D} = \{\boldsymbol{x}[n]\}_{n=1}^{N}$ consists of N i.i.d. observations of a random vector $\boldsymbol{X}$ in $\mathbb{R}^M$. Model-based clustering is commonly used as a basis for cluster analysis, as it provides a framework for choosing the relevant number of clusters in the data as well as assessing the resulting partitioning of the data.

In a model-based setting, if it is further assumed that the base model for each of the clusters is Gaussian, the joint distribution can be represented as a Gaussian mixture model (GMM) of the form:

$$p(\boldsymbol{x}|\boldsymbol{\Theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{B.1}$$

where $\boldsymbol{\Theta}$ represent the collection of all model parameters, $\boldsymbol{\mu}_k$ is the mean vector of cluster k, $\boldsymbol{\Sigma}_k$ is the covariance matrix of cluster k, and π_k is the mixing weight or probability of cluster k, such that $\sum_k \pi_k = 1$ with $0 \leq \pi_k \leq 1$. The generative model for the data is shown in Fig. B.3, where $\boldsymbol{z}[n] \in \{0, 1\}$ is a binary random variable with a 1-of-K encoding in which a particular element z_k is equal to 1 and all other elements are equal to zero. This variable represent the latent cluster assignment for data item n with a marginal distribution specified by the mixing weights, such that $p(z_k = 1) = \pi_k$ and $p(\boldsymbol{z}) = \prod_k \pi_k^{z_k}$ [7].

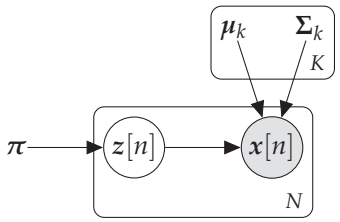


Figure B.3: Meta-network of a Gaussian mixture model.

The log-likelihood of the data under this model is:

$$\log p(\mathcal{D}|\boldsymbol{\Theta}) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}[n]|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right). \tag{B.2}$$

No closed form solution can be derived for the maximization of this expression with respect to the parameters, due to the summation over k that appears

inside the logarithm, and thus it is necessary to resort to an iterative scheme, like expectation maximization (EM), to estimate the parameters of the distribution, see e.g., [7].

The EM algorithm solves the problem of parameters estimation, but one additional problem persists, namely how to choose the number of mixture components K, i.e., the number of clusters represented in the data. One approach to address this issue is to define a likelihood-based score metric that penalizes model complexity, as the likelihood in itself will simply increase as more mixture components are considered, which eventually will lead to overfitting. Two such metrics are the Bayesian information criterion (BIC) [8] and the integrated complete-data likelihood (ICL) [9]:

$$\mathrm{BIC}(\mathcal{M}) = 2\log p(\mathcal{D}|\hat{\boldsymbol{\theta}}) - \nu \log N \tag{B.3}$$

$$\mathrm{ICL}(\mathcal{M}) = \mathrm{BIC}(\mathcal{M}) + 2\sum_{n=1}^{N}\sum_{k=1}^{K} \hat{z}_k[n] \log \tau_k(\boldsymbol{x}[n]), \tag{B.4}$$

where $\mathcal{M}$ reflects the model choice, i.e., number of mixture components and covariance structure, $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimate for the parameter vector under the model, ν is the number of free parameters in the model, $\tau_k(\boldsymbol{x}[n])$ is the probability that $\boldsymbol{x}[n]$ belongs to the kth mixture component, and $\hat{z}_k[n]$ is the cluster assignment of $\boldsymbol{x}[n]$ based on $\boldsymbol{\tau}(\boldsymbol{x}[n])$. Thus,

$$\tau_k(\boldsymbol{x}[n]) = \frac{\hat{\pi}_k \mathcal{N}(\boldsymbol{x}[n]|\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)}{\sum_{k'=1}^{K} \hat{\pi}_{k'} \mathcal{N}(\boldsymbol{x}[n]|\hat{\boldsymbol{\mu}}_{k'}, \hat{\boldsymbol{\Sigma}}_{k'})} \tag{B.5}$$

$$\hat{z}_k[n] = \begin{cases} 1 & \text{if } \arg\max_{k'} \ \tau_{k'}(\boldsymbol{x}[n]) = k \\ 0 & \text{otherwise.} \end{cases} \tag{B.6}$$

As the BIC score tends to select the number of mixture components needed to reasonably approximate the density rather than the number of clusters, the ICL score is used for model selection in the present study. It appears from Eq. B.4 that the ICL score is a penalized version of the BIC score, which adds further penalization through an additional entropy term that reflects cluster overlap. In a Bayesian setting, the ICL approach is equivalent to maximizing an approximation to the integrated complete-data likelihood with non-informative priors, whereas the BIC approach is equivalent to maximizing an approximation to the integrated likelihood with non-informative priors [10, 11].

3 VARIANCE-BASED SENSITIVITY ANALYSIS

In this section, it is outlined how variance-based sensitivity analysis can be performed via the functional ANOVA decomposition for the case of indepen-

dent inputs and via the functional ANCOVA decomposition for the case of dependent inputs.

3.1 ANOVA DECOMPOSITION

In variance-based sensitivity analysis, it is assessed how the variance of the output depends on the uncertain input variables by considering how the output variance can be decomposed. The basis is Sobol's decomposition [12] of response function $Y = f(X_1, X_2, ..., X_M)$ into a set of functions of increasing dimensionality, see Eq. B.7.

$$f = f_0 + \sum_i f_i + \sum_i \sum_{j>i} f_{ij} + ... + f_{12...M}, \tag{B.7}$$

where the individual terms are only functions of the factors in their index, thus $f_i = f(X_i)$, $f_{ij} = f(X_i, X_j)$ and so on.[1] Given that each term in Eq. B.7 is defined to have zero mean, i.e., $\int f_i(x_i)dx_i = 0$ and $\int f(x_i)f(x_j)dx_i dx_j = 0$, the individual terms can be uniquely calculated by use of the conditional expectation of the model output. Thus,

$$f_0 = \mathbb{E}[Y] \tag{B.8}$$

$$f_i = \mathbb{E}[Y|X_i] - f_0 \tag{B.9}$$

$$f_{ij} = \mathbb{E}[Y|X_i, X_j] - f_i - f_j - f_0. \tag{B.10}$$

The first-order sensitivity index corresponds to the variance of the univariate terms $\mathbb{V}_i = \mathbb{V}[f_i] = \mathbb{V}[\mathbb{E}[Y|X_i]]$ scaled by the unconditional output variance $\mathbb{V}[Y]$:

$$S_i = \frac{\mathbb{V}_{X_i}[\mathbb{E}_{\boldsymbol{X}_{\sim i}}[Y|X_i]]}{\mathbb{V}[Y]}, \tag{B.11}$$

where $\boldsymbol{X}_{\sim i}$ denotes all variables except X_i. The index Eq. B.11 represents the main effect contribution from factor i to the output variance [13].

Two factors are said to interact when their effect on Y cannot be expressed as a sum of single effects. For independent input factors, the output variance decomposes as:

$$\mathbb{V}[Y] = \sum_i \mathbb{V}_i + \sum_i \sum_{j>i} \mathbb{V}_{ij} + ... + \mathbb{V}_{12...M}, \tag{B.12}$$

where the terms $\mathbb{V}_{ij}$, $\mathbb{V}_{ijk}$ et cetera correspond to interaction terms. Dividing both sides of Eq. B.12 by the output variance, the following relationship appear:

$$1 = \sum_i S_i + \sum_i \sum_{j>i} S_{ij} + ... + S_{12...M}. \tag{B.13}$$

[1] Note that the decomposition in Eq. B.7, also referred to as high-dimensional model representation (HDMR), is not a series expansion, as it has a finite number of terms.

Based on Eq. B.13, a set of properties can be derived for the first-order sensitivity indices, see Tab. B.1 [13].

Table B.1: Properties of first-order sensitivity indices.

$\sum_i S_i \leq 1$	Always
$\sum_i S_i = 1$	Additive models
$1 - \sum_i S_i$	Indicates presence of interactions

The total effect index represents the joint effect of all contributions related to a factor. That is, the first-order effect of a factor and its higher-order effects due to interactions. Hence, for a three-factor model, the total effect of factor 1 is:

$$S_{T_1} = S_1 + S_{12} + S_{13} + S_{123}. \tag{B.14}$$

The terms in Eq. B.13 could in principle be used to construct the total effect indices but in order to do this, $2^k - 1$ terms must be calculated. That is, this procedure suffers under the curse of dimensionality. Instead, the law of total variance is explored:

$$\mathbb{V}[Y] = \mathbb{V}_{X_i}[\mathbb{E}_{\boldsymbol{X}_{\sim i}}[Y|X_i]] + \mathbb{E}_{X_i}[\mathbb{V}_{\boldsymbol{X}_{\sim i}}[Y|X_i]], \tag{B.15}$$

or equivalently

$$\mathbb{V}[Y] = \mathbb{V}_{\boldsymbol{X}_{\sim i}}[\mathbb{E}_{X_i}[Y|\mathbf{X}_{\sim i}]] + \mathbb{E}_{\boldsymbol{X}_{\sim i}}[\mathbb{V}_{X_i}[Y|\mathbf{X}_{\sim i}]]. \tag{B.16}$$

In both factorizations, the first term represents the variance due the conditioning set, and the second term represents the residual variance, i.e., the variance due to variables not in the conditioning set. In Eq. B.16, the total effect index of variable i is represented by the residual variance divided by the output variance:

$$S_{T_i} = \frac{\mathbb{E}_{\boldsymbol{X}_{\sim i}}[\mathbb{V}_{X_i}[Y|\mathbf{X}_{\sim i}]]}{\mathbb{V}[Y]} = 1 - \frac{\mathbb{V}_{\boldsymbol{X}_{\sim i}}[\mathbb{E}_{X_i}[Y|\mathbf{X}_{\sim i}]]}{\mathbb{V}[Y]}. \tag{B.17}$$

Note that the first-order sensitivity index corresponds to the first term in Eq. B.15 divided by the output variance, cf. Eq. B.11. Equation B.17 provides a more efficient way of calculating the total effect index than the brute force formulation Eq. B.14.

In a variance-based sensitivity assessment, the set of all S_i and S_{T_i} indices provides a reasonable good description of the model sensitivity at a computationally cost that is tractable for most models. Thus, variance-based main effects are suitable in a factor prioritization setting, while total effects address a factor fixing setting [13].

3.2 ANCOVA DECOMPOSITION

The basis for variance decomposition in the case of correlated inputs is again Eq. B.7. For this case, the unconditional variance of the model output may be written:

$$\mathbb{V}[Y] = \mathbb{E}\left[(Y - \mathbb{E}[Y])^2\right] \tag{B.18}$$

$$= \mathbb{E}\left[(Y - f_0)\left(\sum_{\boldsymbol{u}\subseteq\{1,..,k\}} f_{\boldsymbol{u}}\right)\right] \tag{B.19}$$

$$= \mathbb{C}\left[Y, \sum_{\boldsymbol{u}\subseteq\{1,..,k\}} f_{\boldsymbol{u}}\right] \tag{B.20}$$

$$= \sum_{\boldsymbol{u}\subseteq\{1,..,k\}} \mathbb{C}\left[Y, f_{\boldsymbol{u}}\right] \tag{B.21}$$

$$= \sum_{\boldsymbol{u}\subseteq\{1,..,k\}} \left[\mathbb{V}[f_{\boldsymbol{u}}] + \mathbb{C}\left[f_{\boldsymbol{u}}, \sum_{\boldsymbol{v}\subseteq\{1,..,k\}, \boldsymbol{v}\cap\boldsymbol{u}=\emptyset} f_{\boldsymbol{v}}\right]\right] \tag{B.22}$$

where each function $\{f_{\boldsymbol{u}}|\boldsymbol{u} \subseteq \{1,..,k\}\}$ represents the combined contribution of the variables $\boldsymbol{X}_{\boldsymbol{u}}$ to Y. Moreover, Eq. B.21 holds because the variance of Y can be written as the covariance of Y and its functional decomposition minus the zero-order term f_0, and Eq. B.22 holds because Y also contains the functions $f_{\boldsymbol{u}}$ [14, 15].

The variance-covariance decomposition in Eq. B.22 facilitates a separation of the uncorrelated and correlated effects in the following sensitivity indices:

$$S_{\boldsymbol{u}} = \frac{\mathbb{C}[Y, f_{\boldsymbol{u}}]}{\mathbb{V}[Y]} \tag{B.23}$$

$$S_{\boldsymbol{u}}^{U} = \frac{\mathbb{V}[f_{\boldsymbol{u}}]}{\mathbb{V}[Y]} \tag{B.24}$$

$$S_{\boldsymbol{u}}^{C} = \frac{\mathbb{C}\left[f_{\boldsymbol{u}}, \sum_{\boldsymbol{v}\subseteq\{1,..,k\}, \boldsymbol{v}\cap\boldsymbol{u}=\emptyset} f_{\boldsymbol{v}}\right]}{\mathbb{V}[Y]}. \tag{B.25}$$

From this definition of indices, it is seen that $S_{\boldsymbol{u}}$ represents the total contribution to output variance due to $\boldsymbol{X}_{\boldsymbol{u}}$, $S_{\boldsymbol{u}}^{U}$ represents the uncorrelated share of output variance due to $\boldsymbol{X}_{\boldsymbol{u}}$, and $S_{\boldsymbol{u}}^{C}$ represents the correlated share of output variance due to $\boldsymbol{X}_{\boldsymbol{u}}$, i.e., the contribution due to correlations between $\boldsymbol{X}_{\boldsymbol{u}}$ and the other input variables. Moreover, the relationship between the indices is:

$$S_{\boldsymbol{u}} = S_{\boldsymbol{u}}^{U} + S_{\boldsymbol{u}}^{C}. \tag{B.26}$$

As a result of this definition, $S_{\boldsymbol{u}}^{U}$ is always positive, the sign of $S_{\boldsymbol{u}}^{C}$ depends on the nature of the correlation between $\boldsymbol{X}_{\boldsymbol{u}}$ and the other input variables,

and thus the sign of $S_{\boldsymbol{u}}$ depends on which of the structural contribution $S_{\boldsymbol{u}}^{U}$ and correlative contribution $S_{\boldsymbol{u}}^{C}$ is largest. In this context, $S_{\boldsymbol{u}}^{C}$ should be understood as a corrective term that indicates whether the total contribution is overestimated or underestimated because of the correlation between inputs. If $|S_{\boldsymbol{u}}^{C}|$ is small the correlation has a weak influence of the contribution of $\boldsymbol{X}_{\boldsymbol{u}}$, and if it is large the correlation has a strong influence of the contribution of $\boldsymbol{X}_{\boldsymbol{u}}$ [14, 15].

Finally, Saltelli ét al. [16] argue that the condition $\mathbb{E}[\mathbb{V}[Y|\mathbf{X}_{\sim i}]] = 0$ is a necessary and sufficient condition to deem X_i non-influential, under any model or correlation/dependency structure among the inputs. Note that in case of correlation among the inputs, the total effect terms can be smaller than the first-order terms, see [16] for further details.

4 CASE STUDY

4.1 INTRODUCTION

The example considers a portal frame structure [17, 18] subjected to a horizontal and a vertical concentrated load, i.e., P_1 and P_2, respectively. The model has five nodes and four elements, and failure is defined to be an event where any of the structural elements exceed the corresponding moment capacity M_j. Since a hinge can form at either side of an element, a total of 8 hinge locations are considered, which are denoted by $1, 2, \ldots, 8$ in Fig. B.4. The loads and moment capacities are modeled as independent random variables according to Tab. B.2.

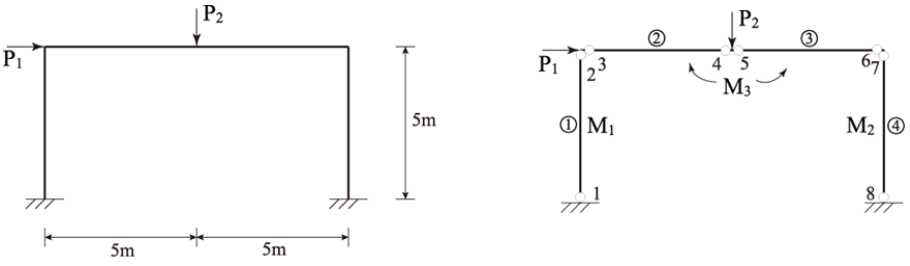


Figure B.4: Left: Portal frame structure. Right: Structural model properties.

Table B.2: Probabilistic model.

Input	Distribution	Mean	Unit	CoV
P_1	Weibull	23.75×10^3	[N]	0.30
P_2	Weibull	43.50×10^3	[N]	0.09
M_1	Lognormal	100×10^3	[N·m]	0.05
M_2	Lognormal	100×10^3	[N·m]	0.05
M_3	Lognormal	93.5×10^3	[N·m]	0.05

4.2 RELIABILITY MEASURES

The probability of failure P_{F_j} associated with the failure event F_j is expressed in terms of the probability integral:

$$P_{F_j} = \int_{g_j(\boldsymbol{x})} q(\boldsymbol{x}) d\boldsymbol{x}, \quad j = 1, \ldots, 8, \tag{B.27}$$

where the random variables $\boldsymbol{X}$ are fully characterized by the multidimensional probability density function $q(\cdot)$. Moreover, the failure domain g_j corresponding to the failure event F_j is defined as:

$$g_j = 1 - D_j \leq 0, \quad j = 1, \ldots, 8. \tag{B.28}$$

Failure occurs when the internal moments $\widehat{M}_j$ of any hinge node exceed the capacity M_j of the corresponding element. Thus, the failure events F_j associated with each hinge node is given by:

$$F_j = D_j > 1, \quad j = 1, \ldots, 8, \tag{B.29}$$

where the normalized demand D_j is defined as:

$$D_j = \frac{\widehat{M}_j}{M_j}. \tag{B.30}$$

The properties of the structural elements are selected so that the reliability of the structural system satisfies:

$$P_{F_j} \leq 10^{-4}, \quad j = 1, \ldots, 8. \tag{B.31}$$

4.3 ROBUSTNESS INDEX

An incremental non-linear analysis is performed, where the loads are gradually increased while solving successive states of equilibrium. This allows to calculate the corresponding robustness indices, see e.g., [19]:

$$I_R = \frac{c_{D,I}}{c_{D,I} + c_{D,P}}, \tag{B.32}$$

where $c_{D,I}$ and $c_{D,P}$ represent the direct consequences associated with the initiation phase and the propagation phase of the failure scenario of the system, respectively. In this particular case we have:

$$I_R = \frac{1}{N_H}, \tag{B.33}$$

where N_H represents the number of failures or hinge locations when the incremental non-linear analysis is finished. Table B.3 and Fig. B.5 show the 12 different scenarios, where it is indicated with 1 if the node immediately becomes plastic, with 2 if it becomes plastic after the first redistribution of internal forces, and with 3 if it becomes plastic after the second redistribution of internal forces. Table B.3 also shows the number of realizations N_F that leads to the corresponding failure scenario, the probability of occurrence P_F, and the robustness index I_R for each failure scenario from a total of 10^8 simulations.

Table B.3: Failure scenarios.

	1	2	3	4	5	6	7	8	N_F	P_F	I_R
SC 1	0	0	0	0	0	0	0	1	1,934	1.90×10^{-5}	1.00
SC 2	0	0	0	0	0	0	1	0	18	1.80×10^{-7}	1.00
SC 3	0	0	0	0	0	0	1	2	17	1.70×10^{-7}	0.50
SC 4	0	0	0	0	0	0	2	1	249	2.50×10^{-6}	0.50
SC 5	0	0	0	0	0	1	0	0	9,263	9.30×10^{-5}	1.00
SC 6	0	0	0	0	0	1	0	2	181	1.80×10^{-6}	0.50
SC 7	0	0	0	0	0	2	0	1	122	1.20×10^{-6}	0.50
SC 8	0	0	0	1	1	0	0	0	19	1.90×10^{-7}	1.00
SC 9	0	0	0	1	1	2	0	0	7	7.00×10^{-8}	0.50
SC 10	0	0	0	2	2	1	0	0	47	4.70×10^{-7}	0.50
SC 11	0	0	0	2	2	1	0	3	3	3.00×10^{-8}	0.33
SC 12	0	0	0	3	3	1	0	2	1	1.00×10^{-8}	0.33
$\sum$									11,861		

4.4 CLUSTER ANALYSIS

In order to enhance the understanding of the probabilistic characteristics of the performances of the considered structural system, a cluster analysis is undertaken on the data related to the different failure scenarios (SCs) for which $P_F \geq 5 \cdot 10^{-7}$, according to Tab. B.3, using Gaussian mixture models (GMMs), as discussed in Sec. 2. Furthermore, also the optimal clustering of a pooled database is computed, considering these SC realizations as one data set. In our GMM implementation functionalities from the publicly available R toolbox mclust [11] are utilized.

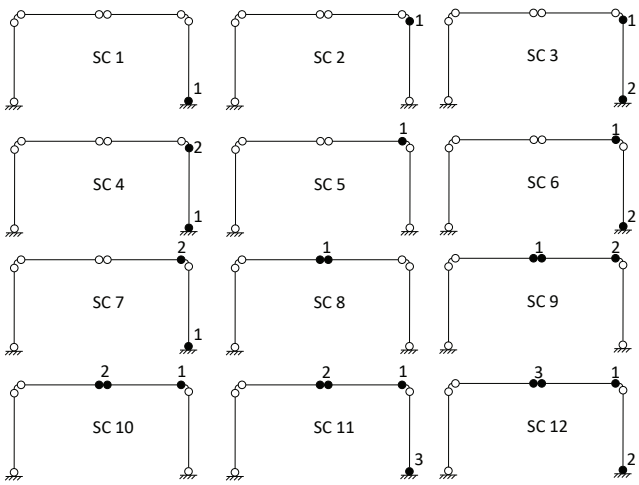


Figure B.5: Failure scenarios.

Figure B.6 shows the mean vectors of all the considered SCs (all scenarios are represented by either one or two clusters). It appears that the SCs are generally governed by the larger positive realizations of P_1 and to a lesser degree P_2. Moreover, it is seen that the M_1 component of all mean vectors is ≈ 0, which indicates that this variable is unimportant for this failure assessment. The two last variables M_2 and M_3 seems to alternate between two bounding patterns, i.e., (i) the M_2 component is zero, and the M_3 component take on a large negative value; and (ii) the M_2 component take on a large negative value, and the M_3 is zero. Bounding pattern (i) reflects failures in the beam, and (ii) reflects failures in the right column, see Fig. B.5 and Tab. B.3.

If the optimal clustering of the pooled database is computed, it is seen that the two governing patterns emerge; and adding additional mixture components simply adds traces around these governing patterns. Figures B.7 and B.8 show the cluster means when two and three clusters are considered, respectively. In Fig. B.8, for example, it is observed that by considering an additional mixture component compared to Fig. B.7, both clusters in SC5 now appear in the mixture distribution of the pooled database, but, for clustering purposes, it may be so that these can be adequately represented by the red cluster in Fig. B.7, depending on the application.

Finally, it should be noted that the cluster analysis significantly enhances the understanding of which scenarios of realizations of loads and resistances contribute to the structural robustness performances, and how much. The identified scenarios are seen to either result in robustness indexes I_R equal to 1 or 0.5. However, it is also seen from Tab. B.3 that the probabilities of scenarios leading to $I_R = 0.5$ are one or two orders of magnitude lower than

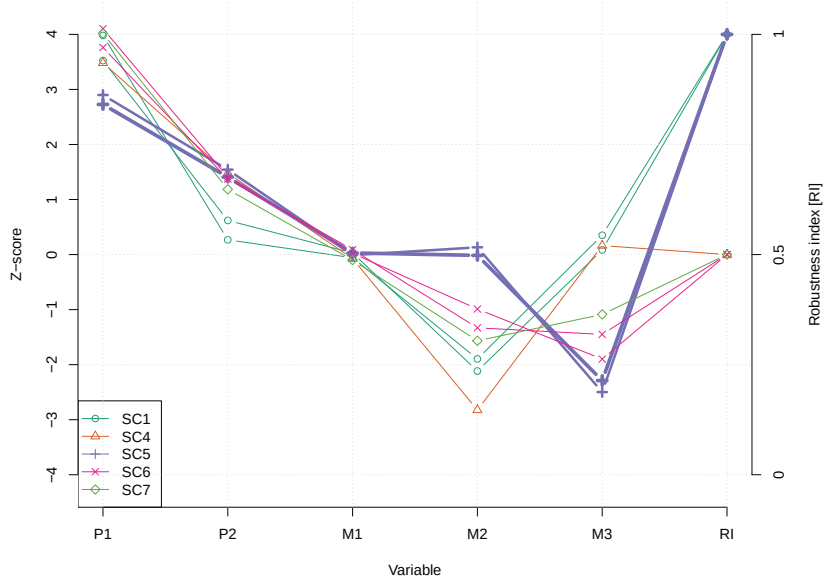


Figure B.6: Mean vectors of selected failure scenarios (standard normal space). The line width indicates the number of data points assigned to each cluster.

the probabilities of scenarios leading to $I_R = 1$. The structure thus performs rather robust in the sense that initial failures do not tend (in probability) to propagate into further failures. Moreover, from Fig. B.6, it is seen that the scenarios leading to $I_R = 0.5$ (e.g., SC6) are dominated by realizations of high values of the load P_1 and realizations of low values of the resistance M_2 and M_3. This information sheds light on in which manner the robustness performance of the structure might be improved by increasing the yield capacities of M_2 and M_3.

4.5 SENSITIVITY ANALYSIS

As a means to investigate how uncertainties associated with the probabilistic representation of the considered structural system affects structural performances, which might be of interest in the context of structural health monitoring for damage detection, a variance-based sensitivity analysis is conducted in the following. To this end, the sample points related to failure scenario 5 (SC5) are utilized together with the corresponding samples of the horizontal displacement at node 4 (hinge 6–7). The displacement acts as an indicator for damage and is regarded as the response variable in this assessment.

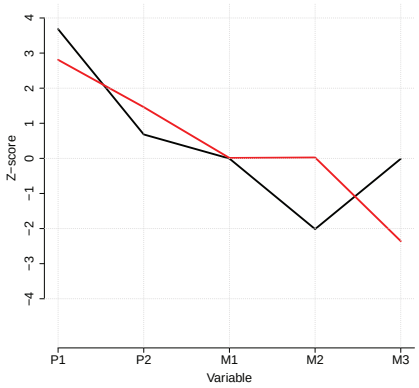


Figure B.7: Clustering of pooled database (2 clusters; standard normal space).

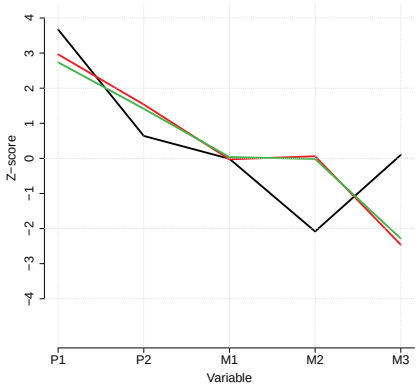


Figure B.8: Clustering of pooled database (3 clusters; standard normal space).

A surrogate model of the response variable is established utilizing a polynomial chaos expansion (PCE) [20]. This greatly enhances the efficient evaluation of the sensitivities, without having to run the underlying computational model. This approach also has the very important merit that it scales well to domains where the underlying computational model is computationally expensive to evaluate, as well as situations where the analyses are based on an experimental data. Furthermore, an approximation to the sensitivity indices defined in Sec. 3 can be derived directly from the parameters of the PCE model, see e.g [15, 21]. The present PCE implementation uses functionalities from the publicly available Python toolbox openTURNS [22].

Figure B.9 shows the marginal distributions and pairwise correlations in the realizations of SC5. It is apparent from the figure that by conditioning on this failure scenario, a conditional sample of correlated input variables is produced from the original sample of uncorrelated input variables, which is in agreement with the GMM representation of this failure scenario. Especially, a positive correlation is observed between P_1 and M_3, and between P_2 and M_3, and a negative correlation is observed between P_1 and P_2. Moreover, it is observed that there is a strong correlation between the response and P_1, a correlation between the response and $\{P_2, M_3\}$, and a weak correlation between the response and $\{M_1, M_2\}$.

To account for the correlation between the inputs, the ANCOVA decomposition, as explained in Sec. 3, is applied for the sensitivity assessment. The results of the sensitivity analysis is illustrated in Tab. B.4. The importance ranking of the contributions is mostly influenced by model structure and to a lesser degree by the correlation between inputs. The highest total contribution is associated with P_1, which has both a significantly higher uncorrel-

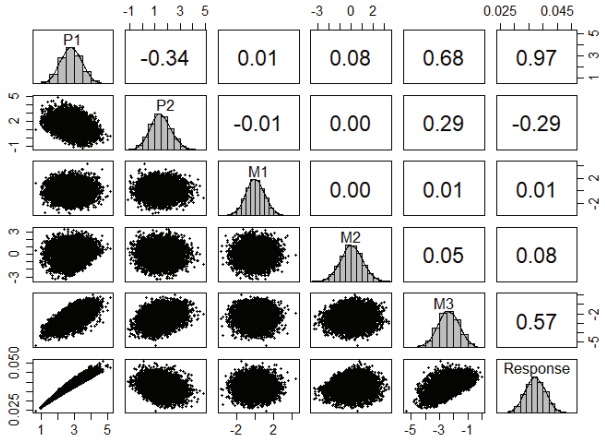


Figure B.9: Marginal distributions and pairwise correlations amongst the realizations of failure scenario 5 (standard normal space).

ative and correlative contribution than the remaining inputs. The inputs P_2 and M_3 both have relevant contributions, whereas insignificant contributions are observed for M_1 and M_2. This is expected, as the response represents a horizontal displacement of the beam, see Fig. B.5. Moreover, as the ANCOVA decomposition allows to distinguish between which part of a contribution is due to the variable itself, and which part is due to its correlation with the other inputs, it is seen, for instance, that the structural contribution of P_2 is three times the total contribution, due to a negative contribution from its correlation with other inputs.

For comparison, the corresponding Sobol indices (ANOVA decomposition) are given in Tab. B.5. As the correlative contributions in Tab. B.4 are generally small in magnitude, the correlations have only a weak effect on the response sensitivity. The first order and total sensitivity indices in Tab. B.5 are equal, and the ranking of the parameters follows the total contribution of the ANCOVA indices. The ANCOVA indices related to P_1 and P_2 (0.933 and 0.001) are slightly smoothed compared to the corresponding Sobol indices (0.935 and 0.003) because of the mutual correlation between the two variables. The same observation holds true for M_3, which has a positive mutual correlated with both P_1 and P_2, see Fig. B.9. Finally, the total effect indices of Tab. B.5 show that M_1 and M_2 provide non-significant contributions to the output variance and may be fixed to an arbitrary value in their range.

Table B.4: Ancova first-order sensitivity indices.

Input	S_i	S_i^U	S_i^C
P_1	0.933	0.932	0.001
P_2	0.001	0.003	-0.002
M_1	≈ 0	≈ 0	≈ 0
M_2	≈ 0	≈ 0	≈ 0
M_3	0.067	0.064	0.003
$\sum$	1.000	0.999	0.001

Table B.5: Sobol sensitivity indices.

Input	S_i	S_{T_i}
P_1	0.935	0.935
P_2	0.003	0.003
M_1	≈ 0	≈ 0
M_2	≈ 0	≈ 0
M_3	0.062	0.062
$\sum$	1.000	1.000

5 CONCLUSION

The challenge of understanding the responses of complex probabilistic representations of systems is taken up from the perspective of utilizing the potentials of modern techniques of data mining. Assuming that a probabilistic model is available for the representation of the relevant performances of a system, the suggestion in the present contribution is to (i) utilize model-based cluster analysis as a means to achieve understanding of which scenarios of realizations, represented by probabilistic system representations, govern system performances of particular interest; and (ii) map how the uncertainties associated with the probabilistic modeling of the system propagate and influence the uncertainties associated with the considered system responses. Regarding (i), the approach taken in the present contribution is to perform model-based clustering with a multidimensional Gaussian mixture model, and with respect to (ii), it is shown how the so-called ANOVA and the very recently formulated ANCOVA decompositions may be applied for variance based sensitivity analysis.

The proposed scheme is illustrated on a simple example in which the probabilistic characteristics of non-linear structural performances of a moment resisting frame structure are considered. The example clearly highlights the significant potentials associated with the application of big data techniques to improve the understanding of complex system models. The cluster analysis provides not only a strong means for checking the relevance and physical adequacy of complex system models but also a significant insights on how complex models may be designed, modified, and/or maintained to achieve adequate and cost-efficient performance characteristics with respect to e.g., robustness and also resilience. The sensitivity analysis is especially useful when aiming to reduce model complexity, but also, and very importantly, in the context of structural health monitoring where response characteristics that contain significant information about the state of the system must be identified.

In this contribution, the cluster and sensitivity analysis are conducted as separate tasks in an exploratory data analysis, but in general, these may be used in combination. An example could be a scenario event with a sparse representation in the database. In this case, the joint input model $p(\boldsymbol{X})$ from the cluster analysis could be used to simulated realizations from the considered failure domain, and subsequently these realizations may to propagate through the surrogate model in order to produce a larger sample of realizations $\{\boldsymbol{x}'_n, y'_n\}_{n=1}^{N'}$ of the considered event scenario. Furthermore, both analysis convey essential information characteristics of the inputs leading to a specific failure event, and they may thus be used in combination to better understand the underlying mechanisms driving the considered event scenario or in sequence, where e.g., Morris screening [23] is used as a proxy for the total sensitivity index in order to screen out non-influential variables from the domain of interest, before we proceed with a thorough analysis of the domain through cluster analysis and the main effect index for factor prioritization.

ACKNOWLEDGMENT

The authors gratefully acknowledge the funding received from Centre for Oil and Gas – DTU / Danish Hydrocarbon Research and Technology Centre (DHRTC).

REFERENCES

[1] D. G. Vlachos, "A review of multiscale analysis: examples from systems biology, materials engineering, and other fluid–surface interacting systems," *Advances in Chemical Engineering*, vol. 30, pp. 1–61, 2005.

[2] G. Stefanou, "The stochastic finite element method: Past, present and future," *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 9-12, pp. 1031–1051, 2009.

[3] M. H. Faber, M. A. Maes, J. W. Baker, T. Vrouwenvelder, and T. Takada, "Principles of risk assessment of engineered systems," in *10th International Conference on Applications of Statistics and Probability in Civil Engineering*, 2007.

[4] Joint Committee on Structural Safety (JCSS), *Risk assessment in engineering: principles, system representation & risk criteria*. Ed. M. H. Faber, 2008.

[5] J. Qin and M. H. Faber, "Risk management of large rc structures within spatial information system," *Computer-Aided Civil and Infrastructure Engineering*, vol. 27, no. 6, pp. 385–405, 2012.

[6] J. G. Sepúlveda and M. H. Faber, "Benchmark of emerging structural reliability methods," in *13th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP)*, 2019, pp. 1–8.

[7] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[8] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[9] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated completed likelihood," *IEEE transactions on pattern analysis and machine intelligence*, vol. 22, no. 7, pp. 719–725, 2000.

[10] J.-P. Baudry, "Estimation and model selection for model-based clustering with the conditional classification likelihood," *Electronic journal of statistics*, vol. 9, no. 1, pp. 1041–1077, 2015.

[11] L. Scrucca, M. Fop, T. B. Murphy, and A. E. Raftery, "mclust 5: clustering, classification and density estimation using gaussian finite mixture models," *The R journal*, vol. 8, no. 1, p. 289, 2016.

[12] I. M. Sobol, "On sensitivity estimation for nonlinear mathematical models," *Mathematical Models and Computer Simulations*, vol. 1, no. 4, pp. 407–414, 1993.

[13] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

[14] G. Li, H. Rabitz, P. E. Yelvington, O. O. Oluwole, F. Bacon, C. E. Kolb, and J. Schoendorf, "Global sensitivity analysis for systems with independent and/or correlated inputs," *Journal of Physical Chemistry*, vol. 114, no. 19, pp. 6022–6032, 2010.

[15] Y. Caniou, "Global sensitivity analysis for nested and multiscale modelling," Ph.D. dissertation, Blaise Pascal University – Clermont II, 2012.

[16] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto, "Sensitivity analysis in practice: a guide to assessing scientific models," *Chichester, England*, 2004.

[17] P. Thoft-Christensen and Y. Murotsu, *Application of structural systems reliability theory*. Springer-Verlag, 1986.

[18] D.-S. Kim, S.-Y. Ok, J. Song, and H.-M. Koh, "System reliability analysis using dominant failure modes identified by selective searching technique," *Reliability Engineering & System Safety*, vol. 119, pp. 316–331, 2013.

[19] M. H. Faber, J. Qin, S. Miraglia, and S. Thoens, "On the probabilistic characterization of robustness and resilience," *Procedia Engineering*, vol. 198, pp. 1070–1083, 2017.

[20] R. G. Ghanem and P. D. Spanos, *Stochastic finite elements: a spectral approach*. Dover Publications, 2003.

[21] L. L. Gratiet, S. Marelli, and B. Sudret, "Metamodel-based sensitivity analysis: polynomial chaos expansions and gaussian processes," in *Handbook of Uncertainty Quantification*, R. Ghanem, H. Owhadi, and D. Higdon, Eds. Springer, 2017, ch. 38, pp. 1289–1325.

[22] M. Baudin, A. Dutfoy, B. Iooss, and A.-L. Popelin, "Openturns: An industrial software for uncertainty quantification in simulation," in *Handbook of Uncertainty Quantification*, R. Ghanem, H. Owhadi, and D. Higdon, Eds. Springer, 2017, pp. 2001–2038.

[23] M. D. Morris, "Factorial sampling plans for preliminary computational experiments," *Technometrics*, vol. 33, no. 2, pp. 161–174, 1991.

PAPER C

A FRAMEWORK FOR OFFSHORE LOAD ENVIRONMENT MODELING

Sebastian T. Glavind, and Michael H. Faber

The paper has been published in the
Journal of Offshore Mechanics and Arctic Engineering, vol. 142, no. 2,
pp. 021702, OMAE-19-1059, 2020.[0]

[0]This paper was presented at *ASME 2018 37th International Conference on Ocean, Offshore, and Arctic Engineering*, Paper No. OMAE2018-77674 (Paper A). Contributed by the Ocean, Offshore, and Arctic Engineering Division of ASME for publication in the *Journal of Offshore Mechanics and Arctic Engineering*.

The layout has been revised.

ABSTRACT

This paper presents a novel decision analytical framework for systems modeling in the context of risk informed integrity management of offshore facilities. Our focus concerns the development of system models representing environmental loads associated with storm events. Appreciating that system models in general serve to facilitate the optimal ranking of decision alternatives, we formulate the problem of systems modeling as an optimization problem to be solved jointly with the ranking of integrity management decision alternatives. Taking offset in recent developments in structure learning and Bayesian regression techniques, a generic approach for the modeling of environmental loads is established, which accommodates for a joint utilization of phenomenological understanding and knowledge contained in databases of observations. In this manner, we provide a framework and corresponding techniques supporting the combination of bottom-up and top-down modeling. Moreover, since phenomenological understanding and analysis of databases may lead to the identification of several competing system models, we include these in the formulation of the optimization problem. The proposed framework and utilized techniques are illustrated in an example. The example considers systems modeling and decision optimization in the context of a possible evacuation of an offshore facility in the face of an emerging storm event.

***Keywords**: ocean waves and associated statistics, structural safety and risk analysis, system integrity assessment.*

1 INTRODUCTION

In the context of the newly established Danish Hydrocarbon Research and Technology Centre (DHRTC), major initiatives have been launched to identify new, safe, and more efficient frameworks and approaches to facilitate the optimization of assets integrity management decisions. This study is an early report on one of these activities, where focus is directed on how the rationale for the development of knowledge concerning the offshore load environment may be improved. In particular, we asses two avenues for improving probabilistic engineering modeling in support of decision-making, i.e., through the modeling basis and the model representation.

1.1 ON INFORMATION AND KNOWLEDGE

Knowledge and information form the basis for representing the systems, which are subject to decision optimization. Thus, before proceeding on the topic of development of models, we will start out with a brief outline on how we account for this basis. Following the guideline for system representations

proposed by the Joint Committee on Structural Safety (JCSS) [1], Fig. C.1 provides a system representation in terms of the flow of consequences generated as a result of exposure events.

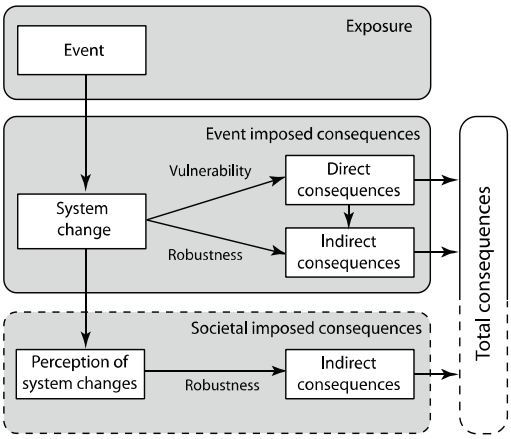


Figure C.1: The JCSS systems representation.

The flow of consequences and their magnitude are generally subject to uncertainty of both aleatory and epistemic character. In accordance with JCSS [1], this uncertainty may be adequately represented by means of the Bayesian probability theory. Following Bayesian decision analysis [2] and the axioms of von Neumann and Morgenstern [3], decision optimization on the management of the system, as illustrated in Fig. C.1, may be supported by the available knowledge and information about how decisions change the generation of the expected value of consequences. Nielsen et al. [4] points out that the knowledge and information, which is relevant to consider, when establishing a probabilistic systems representation (a model), is the knowledge and information that affects the identification of optimal decisions, i.e., the ranking of decision alternatives. With this insight, it becomes obvious that the process of model building and systems management, i.e., the context of the model building, should not be separated.

1.2 MODELING BASIS

Traditional models are most often based on a phenomenological understanding, e.g., probabilistic physics model formulations with parameters estimated based on statistical evidence achieved through observations and experiments ("bottom-up" approaches). It is evident that such approaches rely strongly on the adequacy of a-priori available knowledge and information, which is not always granted. As a result, it is generally the case that all focus of the model-

ing is directed on what is understood to be the most likely physical formulation of the phenomena of interest, and other possible explanations are implicitly excluded. Moreover, possible variables not subjectively realized to affect the phenomena of interest are systematically omitted in the modeling, which in turn increases the uncertainty associated with the derived models. In recent years, robust so-called data-driven modeling approaches ("top-down" approaches) have been formulated and increasingly applied with success in a wide range of applications, see e.g., [5–9]. Data-driven approaches facilitate that models are derived directly from data contained in e.g., databases and do not necessitate an understanding of the phenomena generating the data by the analyst. Data-driven approaches generally identify the most likely relationship between covariates and observations and facilitate a quantification of this likelihood. However, the downside to data-driven approaches is that they may indeed result in models contradicting established knowledge, e.g., generally accepted causal relationships.

One objective of the present research is thus to assess whether a combination of bottom-up and top-down modeling may be formulated, which facilitates a consistent utilization of prior phenomenological knowledge and knowledge extracted from information contained in databases. Moreover, this formulation should facilitate that in principle all possible and relevant likely models may be identified and quantified with respect to their likelihood.

1.3 MODEL REPRESENTATION

Engineering modeling, e.g., in the context of assets integrity management, is traditionally undertaken by interfacing domain-specific models, established individually by subject matter experts. The domain-specific models (e.g., models of the wave environment, water particle kinematics, hydraulic forces, structural responses, and failure criteria) are generally developed in accordance with the best available knowledge within the relevant domains of expertise, and they are optimized individually to provide the highest degree of precision with the available and achievable information, see e.g., [10, 11]. The decisions regarding how to optimize precision are generally based on the prior understanding of the domain experts and often assessed without specific consideration of the context in which the models are applied. One example of this approach is development of a so-called digital twin model of a structure, where a numerical structural model is adopted to information collected using techniques of structural health monitoring, see e.g., [12–14]. Such approaches surely provide a basis for supporting decisions; however, they neither facilitate for a context-driven optimization of the individual models nor a joint optimization of the interfaced models. As a result, models may be unnecessarily precise in domains, which are not important for the decision

context and not adequately precise in domains of special importance for the decision context.

Another objective of the present research is thus, under consideration of the findings related to the first objective, to establish a theoretical and methodical basis for engineering modeling, which facilitates for integrating the optimization of model representation directly into the decision context, through an assessment of how the precision of the modeling affects the ranking of the considered decision alternatives.

The remainder of this paper is organized as follows: Secs. 2 and 3 present a novel decision analytical framework for systems modeling in the context of the risk-informed integrity management of offshore structures, and, in addition, we provide modeling techniques supporting the combination of the bottom-up and top-down modeling. Section 4 presents the suggested approach considering systems modeling and decision optimization in the context of a possible evacuation of an offshore facility in the face of an emerging storm event. Finally, the proposed approach is discussed in relation to the present best practice and concluded in Secs. 5 and 6.

2 BASIC CONSIDERATIONS ON MODEL BUILDING

Scientific models are established on the premise that they serve decision-making in the generic context of systems performance management, noting that systems may comprise any combination of interactions between applied technology, humans, organizations, and the natural environment. In this context, the objective of model building is to represent the available and relevant knowledge about the performances and/or characteristics of systems in consistency with scientific knowledge and evidence obtained from e.g., experiments and observations. In the following, to represent knowledge and rank decision alternatives, Bayesian probability theory and Bayesian decision analysis are applied, see e.g., [2]. Moreover, the framework of Bayesian networks (BNs) is utilized to represent joint dependency relations in the problem domain, compare with Sec. 3.

2.1 SYSTEMS AND DECISION-MAKING

At the simplest level, a model $\mathcal{M}(a)$ provides a relationship between input and output, measured in terms of utility, conditional on a decision represented by a. Figure C.2a illustrates how a system provides this relationship between decision alternatives $\boldsymbol{a}$ and the associated utilities $\mathbb{U}(\boldsymbol{a})$. The system performance is generally associated with uncertainty, and thus, the

performance (output or utility) is random. In accordance with Bayesian decision theory [2] and the axioms of utility theory [3], the optimal decision alternative is selected from $\boldsymbol{a}$ by optimizing the expected utility, i.e., $a^* = \arg\max_a(\mathbb{E}[\mathbb{U}(a)])$.

In the general case [15, 16], the system under consideration is unknown in itself, and it is unknown which is the most relevant representation of the system. In Fig. C.2b, the variable s represents one choice of system representation out of a set of system representations $\boldsymbol{s}$, and σ represents a realization of the real system. The optimization of decision alternatives is further complicated by the fact that some of the decision alternatives within $\boldsymbol{a}$ are only relevant for one or some of the competing system representations. The optimization of decision alternatives must thus be undertaken jointly with a choice of system representation.

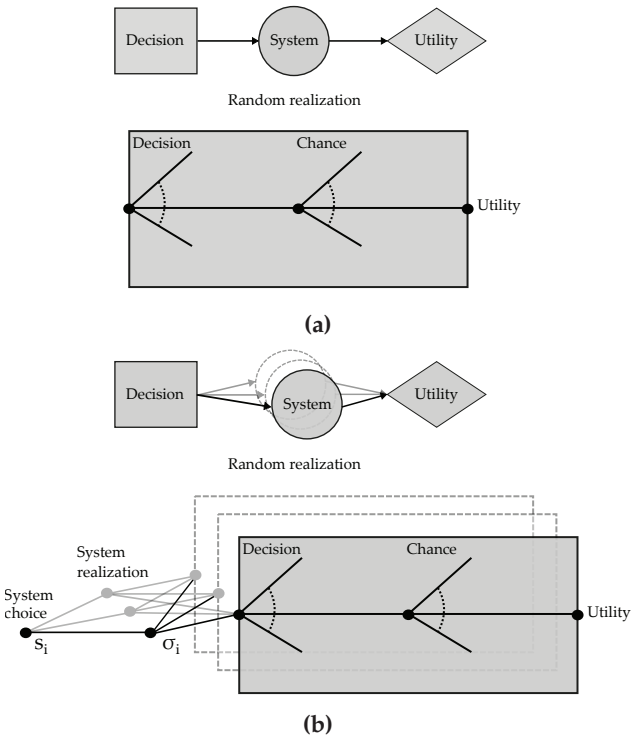


Figure C.2: Systems role in decision analysis: (a) one possible system, (b) several possible systems.

To account for the competing system representations, we introduce the system model $\boldsymbol{\mathcal{M}}(a)$:

$$\boldsymbol{\mathcal{M}}(a) = (\Sigma(a), C(a), \boldsymbol{X}(a))^T, \tag{C.1}$$

where $\Sigma(a)$ is a probabilistic system representation with realizations $\{\sigma_j\}_{j=1}^{n_s}$ corresponding to the set of system choices. Each graph model is comprised of an ensemble of n_{c_j} constituents interacting jointly to provide the functionalities of the system, i.e., mapping input to output. For a given choice of system s, the performances of the constituents are modeled by a set of constituent models C and a prior probabilistic representation $P'(\boldsymbol{X}|s)$ of all variables entering the model. For the sake of generality, we highlight that in principle all the models defining the system have temporal and spatial references; these are omitted here for ease of notation.

The optimization of decision alternatives, including system choice, may now be written as:

$$(s^*, a^*) = \arg\max_{s,a} \mathbb{U}(s,a) = \arg\max_{s} \left(P(\Sigma = s) \arg\max_{a} \left(\mathbb{E}'_{\boldsymbol{X}|s}[\mathbb{U}(a,\boldsymbol{X})] \right) + \mathbb{E}'_{\Sigma\setminus s}\left[\mathbb{E}'_{\boldsymbol{X}|\{\Sigma\setminus s\}}[\mathbb{U}(a^*,\boldsymbol{X})] \right] \right), \tag{C.2}$$

where $a^* = \arg\max_a \mathbb{E}'_{\boldsymbol{X}|s}[\mathbb{U}(a,\boldsymbol{X})]$, see also [16]. In Eq. C.2, the robustness of the decision with regard to the choice of system may be assessed as the ratio of the first term to the sum of the two terms. Thus,

$$\text{Robustness}(s,a^*) = \frac{P(\Sigma = s)\mathbb{E}'_{\boldsymbol{X}|s}[\mathbb{U}(a^*,\boldsymbol{X})]}{\mathbb{E}'_{\Sigma}\left[\mathbb{E}'_{\boldsymbol{X}|\Sigma}[\mathbb{U}(a^*,\boldsymbol{X})]\right]} \tag{C.3}$$

This ratio, which attain values between 0 and 1 (1 = robust), indicates to which degree the expected value of benefits associated with a chosen decision alternative depends – in expected value – on the correctness of the underlying system assumption. That is, how sensitive the chosen decision is with regard to the possibility that the optimization, based on which the decision is identified, is undertaken under an erroneous system assumption. As mentioned by Nielsen et al. [4], an example of decision optimization from offshore engineering in the face of competing systems concerns inspection and maintenance of fatigue crack growth in welded details of steel jacket structures. For a long period, inspection and maintenance were undertaken based on the assumption that findings from inspections originated from the fatigue crack growth. It was later realized, however, that a large proportion of inspection findings originated from welding defects such as slag inclusions and had nothing to do with growing cracks. Robustness of decisions has not attained much systematic attention in the engineering literature, where the tradition for strong model assumptions is widespread and rarely questioned. In the context of global climate change, there is, however, an increased focus on the uncertainties associated with the possible different societal developments that drive the climate changes, see e.g., [17], and the effects of these uncertainties on the adequacy of decision alternatives for mitigation and adaptation.

Furthermore, as indicated earlier, we note that the model building should be seen as an integrated part of the decision optimization. There is no need for a model to be accurate in the domains of "reality", which are irrelevant for the decisions subject to optimization. On the contrary, by embedding the model building operation inside the optimization of decision alternatives, the available knowledge may be fully utilized to optimize the expected value of utility associated with the system under consideration and thus consistently rank decision alternatives.

2.2 INTERPRETATION OF AND REQUIREMENTS TO SYSTEM MODELS

The available approaches for modeling the performance of systems may be categorized as classical engineering understanding-based bottom-up models and data-driven top-down models. However, in either case, evidence can and must be accounted for in the modeling process. As outlined in the foregoing, a model is a representation of reality in the context of decision-making, meaning that a good model facilitates consistent ranking of the considered decision alternatives.

In recent developments on data-driven modeling and data-driven learning, the perspective is often taken that such approaches are superior to bottom-up modeling approaches, since they simply reflect the information contained in the evidence. However, it must be appreciated that "reality" is fundamentally subjective and should be understood as a proxy for "truth" to the extent that this (the truth) is objectively understood. Thus, "reality" is associated with uncertainty, but may be framed through experience and information (knowledge), i.e., a combination of philosophical and scientific insights and observations. Framing of "reality" is thus fundamentally subjective, since it is based on a choice of which experience, which information (data), and which class of models are used as the modeling basis.

The implication of this is that whether bottom-up or top-down approaches, or combinations hereof, are utilized as the basis for modeling of systems performances, the models will always be subjective and thus subject to epistemic uncertainties. A framework for systems modeling from [18] in the context of assets integrity management is illustrated in Fig. C.3.

In Fig. C.3, the concept of indicators is introduced as a means to account for evidence, which is indirectly, and generally more weakly, related to the performances of the system. As an example, an indicator of a short-term maximum crest height could be the significant wave height. Observations of indicators thus provide information; however, they are in general subject to additional uncertainty. The concept of indicators provides a strong means for including evidence in systems modeling, and they may be further used to facilitate multi-scale system representations. This principle of introducing

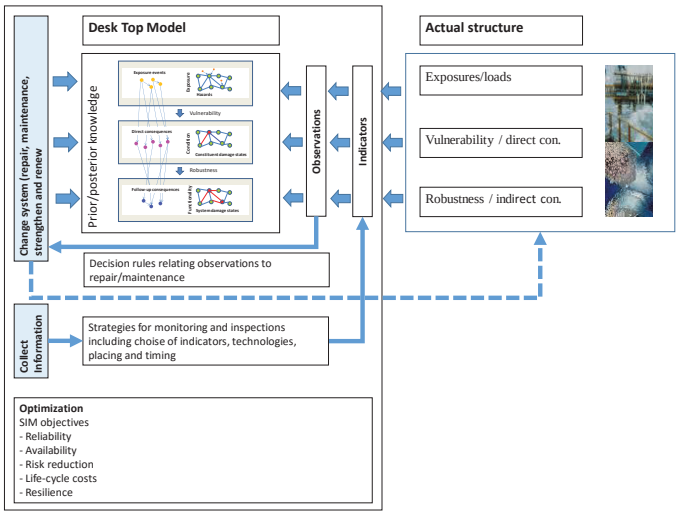


Figure C.3: Systems modeling framework in the context of offshore asset integrity management.

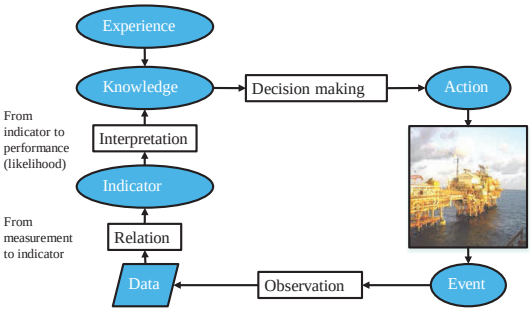


Figure C.4: Illustration of the concept of indicators as applied in Fig. C.3.

evidence achieved through observation of indicators is illustrated in Fig. C.4.

We emphasize that probabilistic system models must consistently account for and distinguish between uncertainty associated with sparsity of evidence, and possible model uncertainty and associated lack of fit. This is of crucial importance in the context of model optimization, where an optimal trade-off between complexity (in terms of graph, constituent, and parameter models) and the associated statistical uncertainties must be identified.

In summary, with reference to Eqs. C.1 and C.2, probabilistic system models should facilitate:

- Representation of multiple possible competing systems (graph models) and associated likelihoods.

- Inclusion of probabilistic constituent models (including aleatory and epistemic uncertainty).
- Probabilistic descriptions of the parameters of the constituent models (including epistemic and aleatory uncertainty).
- Inclusion of evidence obtained from experiments on and observations (including indicators) of the system.
- Consistent representation of statistical uncertainties due to sparsity of evidence.

3 BAYESIAN NETWORK

As mentioned in Sec. 2, we use the framework of Bayesian networks (BNs) to represent probabilistic systems. BNs, which constitute a branch of probabilistic graphical models, encode a joint probability distribution over a set of random variables $\boldsymbol{X}$ by decomposing it into a product of local, conditional probability distributions according to a directed acyclic graph $\mathcal{G}$.

In the graph structure $\mathcal{G}$, each vertex $v_i \in \boldsymbol{V}$ corresponds to a random variable X_i, and the edges $\boldsymbol{E}$ between the vertices represent a set of conditional dependence relations implied by $\mathcal{G}$. Moreover, by studying the missing edges in $\mathcal{G}$, we can directly read off a set of conditional independence relations between the random variables. For each random variable X_i in $\mathcal{G}$, we specify a conditional probability distribution $P(X_i|\mathbf{Pa}_i)$, which defines the dependence of X_i on the random variables that X_i is conditional dependent on in $\mathcal{G}$, termed the parent set $\mathbf{Pa}_i$ of variable X_i. The joint distribution encoded by a BN is given in Eq. C.4:

$$P(\boldsymbol{X}|G, \boldsymbol{\Theta}_{\mathcal{G}}) = \prod_i P(X_i|\mathbf{Pa}_i), \tag{C.4}$$

where $\boldsymbol{\Theta}_{\mathcal{G}}$ denotes the set of model parameters related to graph $\mathcal{G}$. For discrete variables, the set of parameters corresponds to the probability masses of each combination of states: $\boldsymbol{\Theta}_{\mathcal{G}} = \bigcup\{P(x_i|\mathbf{pa}_i) = \Theta_{x_i|\mathbf{pa}_i}\}$. For continuous variables, the parameter set corresponds to the parameters needed to specify the probability density functions of the random variables, see e.g., [19].

3.1 INFERENCE IN BNs

One of the most common inferences made in BNs is the so-called conditional probability query. In a conditional probability query, we compute the posterior distribution $P(\boldsymbol{Y}|\boldsymbol{E}_v = \boldsymbol{e}_v)$ of a subset $\boldsymbol{Y}$ of the variables in the BN, given a (possibly empty) evidence set $\boldsymbol{E}_v = \boldsymbol{e}_v$ on some of the other variables in

the network. By the definition of conditional probability, we may write this probability distribution as follows:

$$P(\boldsymbol{Y}|\boldsymbol{E}_v = \boldsymbol{e}_v) = \frac{P(\boldsymbol{Y}, \boldsymbol{e}_v)}{P(\boldsymbol{e}_v)}. \tag{C.5}$$

In Eq. C.5, the numerator is computed from the factorization of the joint distribution $P(\boldsymbol{X})$, defined by the BN, by marginalizing out the variables $\boldsymbol{W} = \boldsymbol{X} - \boldsymbol{Y} - \boldsymbol{E}_v$, which are neither query nor evidence variables:

$$P(\boldsymbol{Y}, \boldsymbol{e}_v) = \sum_{\boldsymbol{W}} P(\boldsymbol{Y}, \boldsymbol{W}, \boldsymbol{e}_v). \tag{C.6}$$

Here, we assume that the variables are discrete, but the considerations in this section applies equally well to continuous variables or to a combination of discrete and continuous variables, in which case, the summations are replaced, where appropriate, by integrations.

Because $\boldsymbol{Y}$, $\boldsymbol{W}$ and $\boldsymbol{E}_v$ are all the variables in the BN, each term in the summation $P(\boldsymbol{y}, \boldsymbol{w}, \boldsymbol{e}_v)$ is simply one entry in the joint distribution. The denominator in Eq. C.5 may now be computed as follows:

$$P(\boldsymbol{e}_v) = \sum_{\boldsymbol{Y}} P(\boldsymbol{Y}, \boldsymbol{e}_v), \tag{C.7}$$

which allows us to reuse the result of Eq. C.6, instead of having to marginalize out both $\boldsymbol{Y}$ and $\boldsymbol{W}$ from the joint distribution $P(\boldsymbol{X})$ of all variables in the BN [20, 21].

Equation C.6 represents a brute force procedure for computing $P(\boldsymbol{Y}, \boldsymbol{E}_v = \boldsymbol{e}_v)$ called sum-product, where we first compute the product of factors in the summation and then marginalize out the variables $\boldsymbol{W}$ that are not of immediate interest, but there exists a variety of more efficient inference algorithms, both exact, like the sum-product algorithm, and approximate. Some of the more common algorithms are variable elimination, belief propagation, particle-based methods, and variational inference (see e.g., [20] for further details).

3.2 LEARNING BNs

As apparent from Eq. C.4, a BN is fully specified by its graph $\mathcal{G}$ and its parameters $\boldsymbol{\Theta}_{\mathcal{G}}$. The process of specifying the pair $(\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}})$ is termed learning, and it is usually performed in two steps: structure learning and parameter learning. Structure learning refers to the construction of the graph structure $\mathcal{G}$, and parameter learning refers to the specification of the model parameters $\boldsymbol{\Theta}_{\mathcal{G}}$.

Both learning tasks may be undertaken by use of a bottom-up or top-down approach or by a combination hereof. In a bottom-up approach, domain experts are interviewed to identify the graph structure and parameters. In a top-down approach, the graph structure and parameters are established using information provided in a database. The graph structure may be learned either by performing conditional independence test on the database, called constraint-based structure learning, or by optimizing a fit-to-data score metric, called score-based structure learning. Moreover, both frequentistic and Bayesian approaches may be employed for parameter learning [22]. As indicated, BNs are defined in terms of conditional dependence relations and probabilistic properties, without any implication that edges should point from causes to effects. However, it is argued by Pearl [23] that causal BNs pose a more reliable and natural way of expressing our knowledge about the domain we are modeling. That is, we should strive to use a combined learning approach whenever possible, as it makes the best use of the available and relevant knowledge about a given system.

In this paper, we only present a concise description of BNs. The interested reader may refer to the seminal work by Pearl [19, 23], as well as recent prominent textbooks [20, 24, 25].

4 ILLUSTRATION OF THE PROPOSED APPROACH

To illustrate the ideas and methods introduced in this paper, an example is introduced. We consider the following decision problem: The facility manager of an offshore facility is informed that a storm is approaching from a westerly direction. She knows that the facility will fail, if the storm peak significant wave height H_{m0} exceeds 6 m, and she is now faced with a decision of whether to evacuate the facility. She does not know which westerly direction the storm is approaching from, but if it approaches from SW or W, she can choose to either evacuate by helicopter or boat. If the storm approaches from NW, she can only choose to evacuate by helicopter. Furthermore, dependent on the direction from which the storm is approaching, failure will have different consequences.

The decision problem is outlined in Fig. C.5. In the figure, $\{s_j\}_{j=1}^{3}$ corresponds to the choice of storm model, i.e., SW, W, and NW; and $\{a_k\}_{k=1}^{3}$ corresponds to the decision alternatives, i.e., evacuate by helicopter, no evacuation, and evacuate by boat. Please note that the combination s_3 and a_3 is not possible, as evacuation by boat is not possible, if the storm is approaching from NW. Moreover, the system-dependent probability of failure is defined as $p_f = P(H_{m0} > 6\,\mathrm{m}|s)$, the maximum cost $U_{max} = -1$ monetary units,

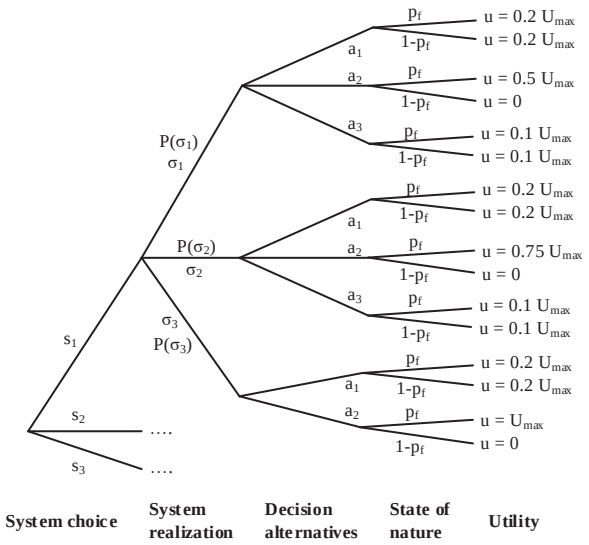


Figure C.5: Illustration of the optimal system choice decision problem.

and $P(\sigma_j)$ represent the probability of storm direction j.

Now, we want to support the facility manager in identifying the optimal decision. Therefore, we build a probabilistic model (BN) for the joint probability distribution of the set of environmental variables in a metocean database (Sec. 4.1). This model is then used to estimate the system-dependent probabilities of failure in Fig. C.5, and thereafter, the decision problem may be solved. The individual steps are described in Secs. 4.1–4.3.

4.1 DATABASE

In this study, we consider the ocean environment of an area located approximately 220 km offshore the west coast of Jutland, Denmark in the North Sea at a water depth of approximately 40 m. The related metocean database is composed of wind fields and corresponding wave hindcast simulator outputs for a period of 37 years from 10th January 1979 to 30th December 2015. Within the considered area, the simulator outputs are sampled at 23 locations.

The hindcast data are produced by use of a MIKE21 spectral wave simulator model [26], with climate forecast system reanalysis wind fields [27] as input. This model is run for a number of different combinations of hindcast tuning and set-up parameters, specified using a Latin hypercube design, to generate multiple sets of wave hindcast outputs, and a single central setting of the parameters is chosen in consultancy with hindcast experts after

verifying that the effect on the outputs is small, when varying the parameter setting.

The simulator outputs for this central parameter setting are then filtered to identify storm peak wind and wave characteristics for storm events using a procedure similar to that used in [28]. Thus, a total of 2187 storm events are defined by exceedances of a threshold, which is non-stationary with respect to season and direction and therefore may not necessarily comply with the meteorological definition of storms. The hindcast storm events are characterized by the variables in Tab. C.1. Note that wind and wave direction are defined as the direction from which they emanate, measured clockwise from north in degrees. More details on this metocean database may be found in [29].

Table C.1: Metocean database.

Variable	Description	Unit
Lng	Longitude	$^\circ$
Lat	Latitude	$^\circ$
Dpt	Water depth	m
WiS	Maximum storm wind speed	m/s
CS	Current speed	m/s
RWL	Residual water level (surge + tide)	m
TY	Time of storm peak	$^\circ$
WaD	Peak wave direction	$^\circ$
WiD	Direction of max. storm wind speed	$^\circ$
CD	Current direction	$^\circ$
$Hm0$	Storm peak significant wave height	m

4.2 LEARNING BNs AND DYNAMIC DISCRETIZATION OF CONTINUOUS DATA

An important prepossessing consideration, when applying BNs to a real-world domain, is how to handle continuous variables. In this study, we discretize random variables with continuous sample spaces. By casting the discretization process as part of the learning problem, we hereby strive to make as few assumptions as possible regarding the distribution family of the domain variables, when learning a BN representation of a given system. In this regard, the number of intervals and their boundaries have to be chosen carefully, as valuable information about the probability distribution of the variables and their dependencies may be lost otherwise. For some of the variables, we predefine the discretization boundaries. This is the case for all directional variables, as well as for the time of storm peak and for the wave

height variable, as we need a specific granularity of these variables in regard to the decision problem.

The remaining variables are discretized by use of a multivariate discretization procedure, embedded in the structure learning procedure, which takes the interactions in the graph structure into account. The method we use is based on [30], where it is assumed that a data set is generated in two steps: First, an interval of a variable is selected from the distribution of the discrete variable. Second, the corresponding continuous value is drawn from a distribution over the interval. We then seek an optimal discrete representation $\mathcal{D}$ of the original continuous data set $\mathcal{D}^c$, which maximizes the objective function: $P(\mathcal{D}|\mathcal{G})P(\mathcal{D}^c|\mathcal{D})$.

As the graph structure changes throughout the structure learning phase, the discretization is adjusted dynamically to maximize the objective function in a manner similar to that proposed in [31]. Thus, we start by learning the optimal discretization of an initial graph structure, which in turn is used to learn a new graph structure. These two steps are repeated until the score function converges to a local optimum. A similar scheme for the combined structure learning and discretization is used in [32].

In our implementation, we use functionalities from the publicly available R package bnlearn [33] to learn the structure of the graph, using the Bayesian Dirichlet equivalent score metric within a tabu search algorithm, which is a score-based learning algorithm.

4.3 RESULTS AND CONCLUSIONS

Initially, the graph structure and optimal discretization are learned using the database. In this regard, the structure learning is constrained by our causal understanding of the domain, e.g., we force an edge going from TY to WiS, WiD, $Hm0$, and RWL, as well as edges meeting at $Hm0$ from WiD and WiS; see Tab. C.1 for a reference on the variable names. The learned graph structure and the corresponding discretization appear in Fig. C.6 and Tab. C.2, respectively.

Apart from the predefined connections, we observe that the BN encodes additional statistical dependencies. For instance, as we would expect, there is a statistical dependence between the water depth at a location and the current speed in a storm, and between the maximum wind speed in a storm and the wind direction. Furthermore, if we consider the discretization, we see that the optimal discretization of the location variables Lng and Lat and the depth variable Dpt is binary. This discretization policy is supported by the scatter plot in Fig. C.7, where we see that the samples of the three variables are clustered in two regions (black and gray).

Having defined the graph structure and discretization, we quantify the model parameters using Bayesian statistics with Dirichlet equivalent uni-

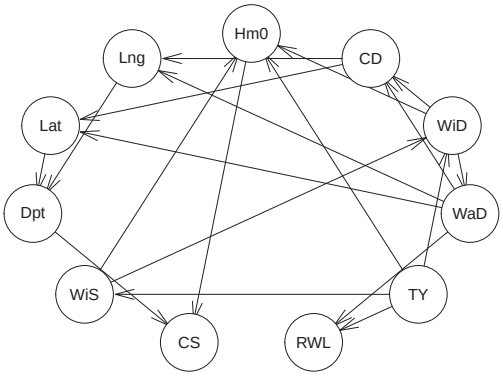


Figure C.6: BN model.

Table C.2: Discretization.

Variable	Levels	Comments
Lng	2	learned
Lat	2	learned
Dpt	2	learned
WiS	9	learned
CS	6	learned
RWL	9	learned
TY	4	predefined ($Spring$, $Summer$, $Fall$, $Winter$)
WaD	8	predefined (NW, N, NE, E, SE, S, SW, W)
WiD	8	predefined (NW, N, NE, E, SE, S, SW, W)
CD	8	predefined (NW, N, NE, E, SE, S, SW, W)
$Hm0$	16	predefined ($(0,2]$, $(2,2.5]$, $(2.5,3]$, ..., $(9,Inf]$)

form priors. By use of this BN model $(\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}})$, the failure probabilities in Fig. C.5 are evaluated, and subsequently, the decision problem is solved by use of Eq. C.2.

In Fig. C.8, the solution is shown for the case of maximum a posteriori (MAP) inference in the BN model. The optimal representation(s) and decision alternative, given representation, are indicated by bold-faced boxes. It appears that both system representations s_1 and s_3 optimize the expected utility, because they agree on the optimal action a^* being not to evacuate.

In case we include the statistical uncertainties in the model evaluation, instead of using the MAP parameters, we obtain probability distributions for $p_f|s$ instead of just fixed quantities. Propagating these through the decision analysis, we obtain probability distributions for the expected utilities $\mathbb{E}[U|s]$ and robustness's as well.

In Fig. C.9, the probability distribution for failure probability, expected

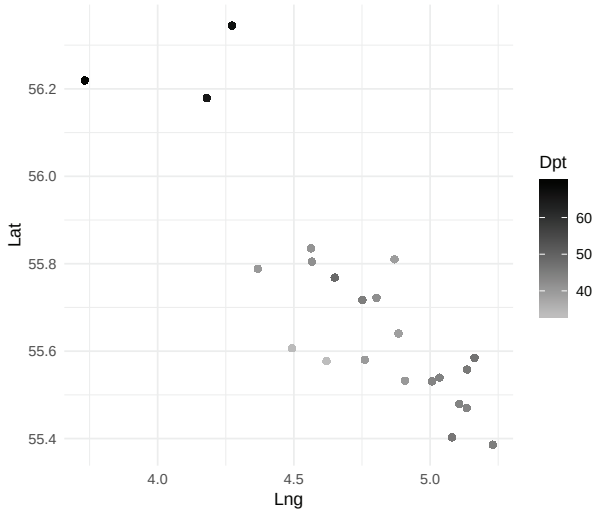


Figure C.7: Scatter plot of sample points for the longitude, latitude, and depth variable.

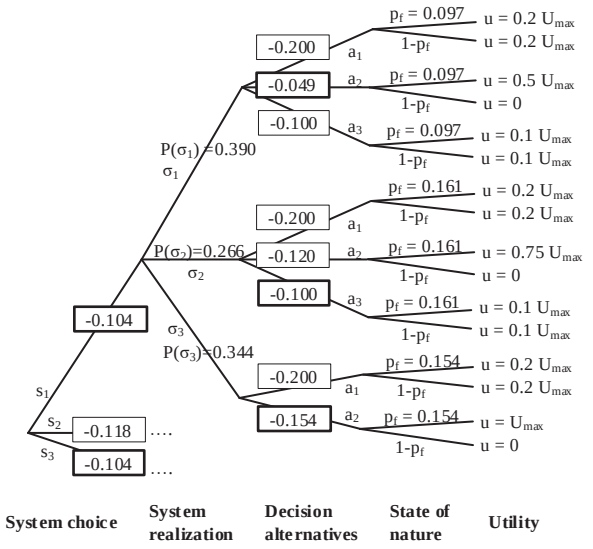


Figure C.8: Solution to the optimal system choice decision problem.

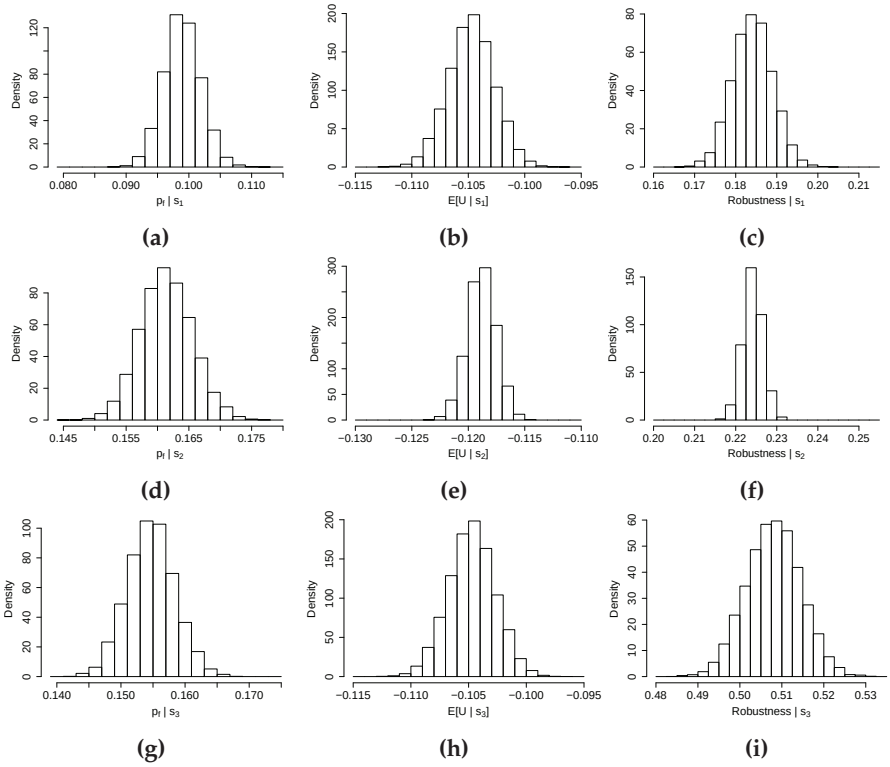


Figure C.9: Optimal system choice decision problem including statistical uncertainties: (a)–(c) show the distribution of failure probability, expected value of utility, and robustness corr. system choice s_1. (d)–(f) and (g)–(i) show the results corr. system choices s_2 and s_3, respectively.

utility, and robustness are shown for the three different system representations. The first column of the figure shows the system-dependent failure probabilities, the second column shows the system-dependent expected values of utilities, and the third column shows the system-dependent robustness. From the first two columns, we see that the quantities spread around the MAP assignments of Fig. C.8 as expected.

As mentioned, the robustness in the third column indicates how sensitive the decisions are with regard to the possibility that the optimization is undertaken under an erroneous system assumption. It appears that the spread around the MAP assignment is largest for system s_3 and smallest for system s_2. Furthermore, we see a significant difference in the robustness for the different system representations. Thus, based solely on the system-dependent expected utilities, we are not able to choose between system representation s_1 and s_3, but from the assessment of the robustness, we see that

the decision based on s_3 is significantly more robust than the decision based on s_1.

In general, the decision analysis may result in different rankings of the systems, and corresponding optimal decision alternatives, for different instantiations of the parameters, whereby we must resort to, for instance, majority voting in the sample to choose the pair (s^*, a^*). In the simple decision problem, we consider in this study, all instantiations of the parameters resulted in the same ranking as of Fig. C.8.

5 DISCUSSION

The approach to integral modeling and decision optimization proposed, outlined, and illustrated in the foregoing provides several advantages compared with traditional, present best practice modeling approaches. The main advantage being that the model building is "value-driven" in the context of its application, whereas traditional modeling approaches at best implicitly or indirectly account for the application of the model when this is formulated and its parameters estimated. A second advantage is that the proposed modeling approach explicitly accounts for possible competing systems, which may influence the adequacy of chosen decision alternatives. Moreover, the presented formulations provide information on the degree to which possible competing systems may influence the optimality of decision ranking. A third advantage of the presented approach is that its technical implementation through the use of Bayesian networks greatly accommodates for the joint utilization of both the prior knowledge and the knowledge that may be extracted from the analysis of the data. This concerns both the causal relationships between model variables and the probabilistic characteristics of these. Finally, it should be highlighted that the proposed modeling approach accommodates for a consistent representation of both aleatory and epistemic uncertainties in the same manner as is normally pursued in traditional modeling schemes. Presently, the proposed modeling approach is being further applied, investigated, and refined in the context of the ongoing DHRTC projects, not only focusing on the probabilistic modeling of the offshore loading environment but also addressing other modeling challenges, such as the stochastic representation of non-linear structural responses and fatigue crack growth in welded tubular joints. These ongoing studies will serve to assess the broader significance of the proposed approach compared to traditional approaches and to improve understanding of its added value.

6 CONCLUSION

The present paper presents early developments on the formulation and implementation of a novel decision analytic framework for systems modeling in the context of risk-informed integrity management of offshore facilities, with a focus on the development of system models representing environmental loads associated with storm events. To account for the fact that system models in general serve to facilitate the optimal ranking of decision alternatives, we formulate the problem of systems modeling as an optimization problem to be solved jointly with the ranking of decision alternatives. Moreover, based on recent developments in structure learning and Bayesian regression techniques, a generic approach for the modeling of environmental loads is established, which accommodates for a joint utilization of phenomenological understanding and knowledge contained in databases of observations. The developed framework and corresponding techniques greatly support the combination of bottom-up and top-down modeling and facilitates for consistently addressing the existence of possible competing systems in the context of assets integrity management. The proposed framework and the utilized techniques are illustrated in an example, where we consider systems modeling and decision optimization in the context of possible evacuation of an offshore facility in the face of emerging storm events. The example shows how a probabilistic model for storm events may be formulated using the Bayesian networks modeling paradigm and how the corresponding system representations may be used in a decision optimization; first, based on MAP inference in the BN model and second, including statistical uncertainties in the model formulation. On the basis of this assessment, we conclude that this decision framework is indeed feasible, and future research will focus on broader and more involved applications of the framework, as well as further algorithmic developments and optimization.

ACKNOWLEDGMENT

The authors gratefully acknowledge the funding received from Centre for Oil and Gas – DTU / Danish Hydrocarbon Research and Technology Centre (DHRTC). We would also like to thank Total E&P, the Danish Hydraulic Institute, and Shell Research Ltd. for providing the data and support needed to conduct this research.

NOMENCLATURE

a **or** $\boldsymbol{a}$	Decision alternative(s).

s **of** $\boldsymbol{s}$	System representation or set of system representations.
x_i **or** $\boldsymbol{x}$	Realization of random variable(s).
$\mathbb{E}(\cdot)$	Expectation operator.
$\mathcal{M}(\cdot)$	System model.
$P(\cdot)$	Probability or (conditional) probability distribution.
$\mathbf{Pa}_i$	Parent set of variable X_i.
$\mathbb{U}(\cdot)$	Utility function.
X_i **or** $\boldsymbol{X}$	Random variable or set of random variables.
σ	System realization.
Σ	Probabilistic system representation.
Θ **or** $\boldsymbol{\Theta}$	Parameter or parameter vector.

REFERENCES

[1] Joint Committee on Structural Safety (JCSS), *Risk assessment in engineering: principles, system representation & risk criteria*. Ed. M. H. Faber, 2008.

[2] H. Raiffa and R. Schlaifer, *Applied statistical decision theory*. MIT Press, 1961.

[3] J. von Neumann and O. Morgenstern, *Theory of games and economic behavior*. Princeton University Press, 1953.

[4] L. Nielsen, S. T. Glavind, J. Qin, and M. H. Faber, "Faith and fakes–dealing with critical information in decision analysis," *Civ Eng Environ Syst*, vol. 36, no. 1, pp. 32–54, 2019.

[5] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, and Y. Si, "A data-driven design for fault detection of wind turbines using random forests and xgboost," *IEEE Access*, vol. 6, pp. 21 020–21 031, 2018.

[6] K. Amasyali and N. M. El-Gohary, "A review of data-driven building energy consumption prediction studies," *Renew Sust Energ Rev*, vol. 81, pp. 1192–1205, 2018.

[7] X. Li, Q. Ding, and J. Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering and System Safety*, vol. 172, pp. 1–11, 2018.

[8] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, "Image reconstruction by domain-transform manifold learning," *Nature*, vol. 555, no. 7697, pp. 487–492, 2018.

[9] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mech Syst Signal Pr*, vol. 115, pp. 213–237, 2019.

[10] European Committee for Standardization (CEN), "Petroleum and natural gas industries - Specific requirements for offshore structures - Part 1: Metocean design and operating considerations," *EN ISO 19901-1:2015*, 2015.

[11] ——, "Petroleum and natural gas industries - Fixed steel offshore structures," *EN ISO 19902:2008*, 2008.

[12] Y. Bazilevs, X. Deng, A. Korobenko, F. L. Di Scalea, M. D. Todd, and S. G. Taylor, "Isogeometric fatigue damage prediction in large-scale composite structures driven by dynamic sensor data," *J Appl Mech*, vol. 82, no. 9, p. 091008, 2015.

[13] C. Li, S. Mahadevan, Y. Ling, S. Choze, and L. Wang, "Dynamic bayesian network for aircraft wing health monitoring digital twin," *AIAA Journal*, vol. 55, no. 3, pp. 930–941, 2017.

[14] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with big data," *Int J Adv Manuf Tech*, vol. 94, no. 9-12, pp. 3563–3576, 2018.

[15] M. H. Faber and M. A. Maes, "Epistemic uncertainties and system choice in decision making," in *Ninth International Conference on Structural Safety and Reliability, ICOSSAR*, 2005, pp. 3519–3526.

[16] M. H. Faber, "On the governance of global and catastrophic risks," *International Journal of Risk Assessment and Management*, vol. 15, no. 5-6, pp. 400–416, 2011.

[17] O. Edenhofer *et al.*, *AR5 Climate change 2014: mitigation of climate change*. Cambridge University Press, 2015, https://www.ipcc.ch/report/ar5/wg3 (Accessed September 16, 2019).

[18] M. H. Faber, "Risk Informed Structural Systems Integrity Management: A Decision Analytical Perspective," in *36th International Conference on Ocean, Offshore and Arctic Engineering*, Trondheim, 2017, p. 62715.

[19] J. Pearl, *Probabilistic Reasoning in Intelligent Systems. Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[20] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.

[21] U. B. Kjærulff and A. L. Madsen, *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*. Springer New York, 2013.

[22] M. Scutari and J.-B. Denis, *Bayesian networks: with examples in R*. CRC press, 2014.

[23] J. Pearl, *Causality*. Cambridge university press, 2009.

[24] S. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Pearson Education Limited, 2014.

[25] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[26] O. R. Sørensen, H. Kofoed-Hansen, M. Rugbjerg, and L. S. Sørensen, "A third-generation spectral wave model using an unstructured finite volume technique," in *29th International Conference on Coastal Engineering*, 2004, pp. 894–906.

[27] S. Saha, S. Moorthi, X. Wu, J. Wang, S. Nadiga, P. Tripp, D. Behringer, Y. T. Hou, H. Y. Chuang, M. Iredell, M. Ek, J. Meng, R. Yang, M. P. Mendez, H. Van Den Dool, Q. Zhang, W. Wang, M. Chen, and E. Becker, "The NCEP climate forecast system version 2," *J Climate*, vol. 27, no. 6, pp. 2185–2208, 2014.

[28] K. Ewans and P. Jonathan, "The effect of directionality on northern north sea extreme wave design criteria," *J Offshore Mech Arct*, vol. 130, no. 4, p. 041604, 2008.

[29] M. Jones, H. F. Hansen, A. R. Zeeberg, D. Randell, and P. Jonathan, "Uncertainty quantification in estimation of extreme environments," *Coastal Eng*, vol. 141, pp. 36–51, 2018.

[30] S. Monti and G. F. Cooper, "A multivariate discretization method for learning bayesian networks from mixed data," in *Fourteenth International Conference on Uncertainty in Artificial Intelligence*, 1998, pp. 404–413.

[31] N. Friedman and M. Goldszmidt, "Discretizing continuous attributes while learning bayesian networks," in *Thirteenth International Conference on Machine Learning*, 1996, pp. 157–165.

[32] K. Vogel, "Applications of bayesian networks in natural hazard assessments," Ph.D. dissertation, University of Potsdam, 2013.

[33] M. Scutari, "Learning bayesian networks with the bnlearn R package," *Journal of Statistical Software*, vol. 35, no. 3, pp. 1–22, 2010.

PAPER D

ON NORMALIZED FATIGUE CRACK GROWTH MODELING

Sebastian T. Glavind, Henning Brüske, and Michael H. Faber

The paper has been published in the
Proceedings of the ASME 2020 39th International Conference on Ocean, Offshore and Arctic Engineering (OMAE2020), OMAE2020-18613, 2020.

The layout has been revised.

ABSTRACT

Modeling of fatigue crack growth plays a key role in risk-informed inspection and maintenance planning for fatigue sensitive structural details. Probabilistic models must be available for observable fatigue performances such as crack length and depth, as a function of time. To this end, probabilistic fracture mechanical models are generally formulated and calibrated to provide the same probabilistic characteristics of the fatigue life as the relevant SN fatigue life model. Despite this calibration, it is recognized that the rather complex fracture mechanical models suffer from the fact that several of their parameters are assessed experimentally on an individual basis. Thus, the probabilistic models derived for these parameters in general omit possible mutual dependencies, and this in turn is likely to increase the uncertainty associated with modeled fatigue lives. Motivated by the possibility to reduce the uncertainty associated with complex multi-parameter probabilistic fracture mechanical models, a so-called normalized fatigue crack growth model was suggested by Tychsen (2017). In this model, the main uncertainty associated with the fatigue crack growth is captured in only one parameter. In the present contribution, we address this new approach for the modeling of fatigue crack growth from the perspective of how to best estimate its parameters based on experimental evidence. To this end, parametric Bayesian hierarchical models are formulated taking basis in modern big data analysis techniques. The proposed probabilistic modeling scheme is presented and discussed through an example considering fatigue crack growth of welds in K-joints. Finally, it is shown how the developed probabilistic crack growth model may be applied as basis for risk-based inspection and maintenance planning.

***Keywords**: structural safety and risk analysis, system integrity assessment.*

1 INTRODUCTION

Fatigue crack growth in welded details is a major contributor to service life costs of offshore facilities. Fatigue induced crack growth poses an important risk to the integrity of offshore structures. Thus, to ensure an adequate level of safety and reliability over the service life of offshore facilities, integrity management measures must be implemented to ensure that possible developments of fatigue cracks are kept within acceptable limits. Due to the substantial uncertainty associated with the offshore fatigue loading environment, fatigue and crack growth degradation processes in general, as well as the quality of inspection and maintenance activities, this poses a rather significant challenge. Over especially the last three decades, significant research and development effort has been invested to identify methods and techniques facilitating optimal integrity management of welded details in offshore oil and gas production systems. Based on risk assessments, so-called

risk-based inspection (RBI) planning methods have been developed and implemented especially for jacket type steel structures, FPSOs and FSOs, see [1] for a comprehensive overview. Using RBI techniques, inspection and maintenance activities may be optimized in such a manner that requirements to the safety of personnel and to the qualities of the environment are fulfilled, and at the same time service life costs are minimized.

One of the key challenges associated with the development of robust and efficient methods for RBI concerns the probabilistic modeling of the crack growth process. In past research and development efforts, the focus to this end has been on the application of fracture mechanics as a means to model the development of cracks from the phase of initiation over propagation to failure in terms of crack through or loss of critical stiffness. In principle this approach works well, and operational RBI techniques have been developed and implemented into practice over the years, see e.g., [2]. However, since the probabilistic characterization of the various parameters entering the fracture mechanical models for practical reasons traditionally is undertaken on a marginal basis, i.e., parameter-wise, the dependencies between these parameters are generally not accounted for, and this might lead to an overestimation of the uncertainty associated with the modeled crack growth. To circumvent this weak point of present best practice in RBI for fatigue sensitive details, Maersk Oil and Gas initiated a research and development project to investigate if it would be possible to develop an alternative formulation of fatigue crack growth models, see [3]. As a result of this initiative, a so-called normalized fatigue crack growth model was formulated by which the crack dimensions, normalized with respect to the critical crack size, may be represented as a function of the fatigue loading (time), normalized by the time until fatigue failure assessed through traditional SN experiments. This type of fracture mechanical representation of fatigue crack growth is thus made possible by normal SN experiments, augmented with observations of crack growth at a selected number of intervals during the experiments.

With this new approach to the modeling of fatigue crack growth, the question arises on how best to estimate the parameters of the models consistently based on experimental evidence. The present paper offers a contribution to this end by considering modern techniques of big data analysis, such as model-based machine learning in conjunction with Bayesian inference. In this paper, we first introduce these techniques. Thereafter, the normalized fatigue crack growth model is outlines, and subsequently the presented techniques and methods are applied to an example addressing probabilistic fatigue crack growth modeling of welded details of K-joints. Finally, it is shown how the developed probabilistic normalized fatigue crack growth model may be applied to derive risk-based inspection plans.

2 MODEL-BASED MACHINE LEARNING

In this section, we describe an emerging methodology for applying machine learning (ML) called model-based machine learning (MBML) [4]. In traditional ML, the practitioner typically selects a suitable ML algorithm from the literature when faced with a new problem. If the algorithm requires modification to comply with the problem at hand, the practitioner must either modify the existing algorithm or combine the algorithm with other ML algorithms from the literature, both of which can be challenging. In contrast, when applying MBML, a custom ML algorithm is formulated for a given problem by decomposing the algorithm construction into two distinct parts, namely (i) probabilistic model representation, and (ii) inference engine. The model representation covers the set of application specific assumptions made about the problem domain, i.e., the process generating the data, where any assumptions regarding uncertainties are expressed using probabilities. The model representation is typically implemented in a compact modeling language from which custom code for learning and inference can be generated automatically based on the chosen inference method. This is referred to as the inference engine. In some cases, of cause, the MBML algorithm might correspond to an existing ML algorithm, while in other cases it may not. The important distinction here is that a MBML algorithm makes the model assumptions explicit through the model representation, while in traditional ML algorithms these are often implicitly defined [4, 5].

As indicated above, the MBML framework offers several advantages when defining a ML algorithm for a given problem, e.g., the ease with which highly tailored models can be created for specific applications; rapid model prototyping and modification for model comparison; compact model representation that permits debugging and collaboration; and the fact that practitioners can focus their attention on understanding a single modeling environment, as many traditional ML algorithms will appear as special cases of the MBML framework [4].

The MBML framework can in principle be implemented using a variety of different approaches [4]; however, here we focus on an approach that leverages Bayesian inference in probabilistic graphical models, e.g., Bayesian networks (BNs), and recent developments in probabilistic programming.

Bayesian networks define a joint probability distribution over a set of random variables $\boldsymbol{X}$ by decomposing it into a product of local, conditional probability distributions according to a directed acyclic graph (DAG) $\mathcal{G}$, i.e., the model structure. In the DAG $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$, each vertex $v_i \in \boldsymbol{V}$ corresponds to a random variable $X_i \in \boldsymbol{X}$, and the edges $\boldsymbol{E}$ between the vertices represent the set of direct dependence relations implied by $\mathcal{G}$. Moreover, by studying the edges and missing edges in $\mathcal{G}$, we can directly read off a set

of (conditional) independence relations between the domain variables. For each random variable X_i in $\mathcal{G}$, we specify a conditional probability distribution $P(X_i|\mathbf{Pa}_i)$, which defines the dependence of X_i on the random variables, which X_i is directly dependent on in $\mathcal{G}$, termed the parent set $\mathbf{Pa}_i$ of variable X_i. The joint distribution defined by a BN is shown in Eq. D.1.

$$P(\boldsymbol{X}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}) = \prod_i P(X_i|\mathbf{Pa}_i), \tag{D.1}$$

where $\boldsymbol{\Theta}_{\mathcal{G}}$ denotes a set of model parameters. For discrete random variables, the set of model parameters correspond to the probability masses of each combination of states: $\boldsymbol{\Theta}_{\mathcal{G}} = \bigcup\{P(x_i|\mathbf{pa}_i)\}$. For continuous random variables, the parameter set corresponds to the parameters needed to specify the probability density functions of the random variables [6].

Probabilistic programming is a programming paradigm in which probabilistic models are specified, and the corresponding inference code generated automatically, based on the chosen inference method. Moreover, it allows probabilistic and conventional, deterministic code to be combined, which provides a modeling flexibility beyond conventional graphical model notation [4]. Programming languages used for probabilistic programming are referred to as probabilistic programming languages. Some examples are BUGS [7], Infer.NET [8], JAGS [9] and Stan [10].

3 BAYESIAN INFERENCE

One of the most common inferences made in probabilistic models is the so-called conditional probability query. In a conditional probability query, we compute the posterior distribution $P(\boldsymbol{Z}|\boldsymbol{E}_v = \boldsymbol{e}_v)$ of a subset $\boldsymbol{Z}$ of the variables in the BN, given a (possibly empty) evidence set $\boldsymbol{E}_v = \boldsymbol{e}_v$ on some of the other variables in the network. By the definition of conditional probability, this probability distribution may be written as

$$P(\boldsymbol{Z}|\boldsymbol{E}_v = \boldsymbol{e}_v) = \frac{P(\boldsymbol{Z}, \boldsymbol{e}_v)}{P(\boldsymbol{e}_v)}. \tag{D.2}$$

In Eq. D.2, the numerator is computed from the factorization of the joint distribution $P(\boldsymbol{X})$, defined by the BN, by marginalizing out the variables $\boldsymbol{W} = \boldsymbol{X} - \boldsymbol{Z} - \boldsymbol{E}_v$, which are neither query nor evidence variables:

$$P(\boldsymbol{Z}, \boldsymbol{e}_v) = \sum_{\boldsymbol{W}} P(\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{e}_v). \tag{D.3}$$

Here, it is assumed that the variables are discrete, but the considerations in this section applies equally well to continuous variables, or to a combination of discrete and continuous variables, in which case, the summations are replaced, where appropriate, by integrals.

Because $\boldsymbol{Z}$, $\boldsymbol{W}$ and $\boldsymbol{E}_v$ are all the variables in the BN, each term in the summation $P(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{e}_v)$ is simply one entry in the joint distribution. The denominator in Eq. D.2 may now be computed as

$$P(\boldsymbol{e}_v) = \sum_{\boldsymbol{Z}} P(\boldsymbol{Z}, \boldsymbol{e}_v), \tag{D.4}$$

which facilitates reuse of the result of Eq. D.3, instead of having to marginalize out both $\boldsymbol{Y}$ and $\boldsymbol{W}$ from the joint distribution $P(\boldsymbol{X})$ of all variables in the BN [11, 12].

Equation D.3 represents a brute force procedure for computing $P(\boldsymbol{Z}, \boldsymbol{E}_v = \boldsymbol{e}_v)$, called sum-product, where; first, the product of factors in the summation is calculated and second, the variables $\boldsymbol{W}$ that are not of immediate interest are marginalized (integrated) out. However, there exists a variety of more efficient inference algorithms, both exact, like the sum-product algorithm, and approximate. Some of the more common algorithms include variable elimination, belief propagation, particle-based methods, and variational inference. See e.g., [11] for further details.

4 NORMALIZED FATIGUE CRACK GROWTH MODELING

In this section, it is shown how the MBML methodology can be used to define a customized ML algorithm for assessing normalized fatigue crack growth.

The first step of our implementation of MBML is to represent the assumptions about the problem domain using the BN formalism. This means specifying the DAG $\mathcal{G}$ and a prior assignment of the parameter vector $\boldsymbol{\Theta}_{\mathcal{G}}$, which holds the parameters of the corresponding (conditional) probability distributions. In this study, the following parametric form, originally proposed by Tychsen [3], is considered for the normalized relationship between load cycles $x = (N/N_c)$ and crack depth $y = (a/a_c)$ as obtained from SN experiments:

$$y = (1+\gamma)x^{\beta} - \gamma \quad s.t. \quad \gamma \geq 0, \tag{D.5}$$

where N is the number of load cycles, N_c is the critical number of load cycles, a is the crack depth, a_c is the critical crack depth, β is a scalar parameter that specifies the growth rate, and γ is a scalar parameter that accounts for crack initiation. Furthermore, it is assumed that the β's and σ's of D experiments can be similar and that the similarity can be inferred based on data, i.e., it is assumed that the β's and σ's have a common population distribution, see Fig. D.1. In this regard, the normalized crack depth y_d is assumed to follow a normal distribution with covariate x_d and standard deviation σ_d, which is assumed to be half-Cauchy distributed with hyper-parameters μ_s and σ_s.

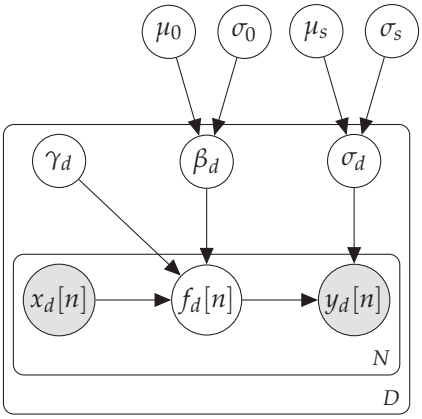


Figure D.1: Meta-network of the hierarchical model.

Moreover, the exponent β_d is assumed to follow a log-normal distribution with hyper-parameters μ_0 and σ_0. Thus,

$$f_d(x) = (1 + \gamma_d)x^{\beta_d} - \gamma_d \tag{D.6a}$$

$$y_d \sim \mathcal{N}(f_d(x), \sigma_d^2) \tag{D.6b}$$

$$\sigma_d \sim \text{half-Cauchy}(\mu_s, \sigma_s) \tag{D.6c}$$

$$\log(\beta_d) \sim \mathcal{N}(\mu_0, \sigma_0^2). \tag{D.6d}$$

The remaining parameters are represented probabilistically through non-informative prior probability assignments. This model is well suited for experiments conducted under different laboratory conditions, where the noise associated with measurements may vary.

The structure of Fig. D.1 is known as a hierarchical or multilevel model, as it expresses the relationship between a set of sub-model (experiments) using hyper-parameters. Note that this is a conditional model for the normalized crack depth y_d given the corresponding normalized load cycles x_d, i.e., $P(Y|X)$, and thus does not comprise a model for $P(X)$.

In the second step of the implementation of MBML, an inference engine must be selected. In the present study, the probabilistic programming language Stan with a Hamiltonian Monte Carlo inference method is selected.

5 BAYESIAN MODEL AVERAGING

So far, the focus has been directed on how to model a set of fatigue experiment outcomes, together with their hyper-parameters, by means of hierarchical modeling. However, in support of inference modeling in the case of

new applications, the framework of Bayesian model averaging (BMA) is introduced in this section.

BMA is an approach to combining queries (predictions or forecasts) from an ensemble of models subject to uncertainty associated with modeling assumptions. The BMA predictive distribution of a quantity of interest provides a weighted average over the posterior predictions for each model, weighted by the model probability, and thus reflect the model's relative contribution to the inference [13].

Consider an ensemble of system representations $\boldsymbol{\mathcal{M}} = \{\mathcal{M}_d\}_{d=1}^{D}$, where each $\mathcal{M}_d$ corresponds to one system representation. Using Bayesian model averaging, inferences are made by averaging over the ensemble models as

$$P(\Delta|\mathcal{D}) = \sum_{d=1}^{D} P(\Delta|\mathcal{M}_d, \mathcal{D}) P(\mathcal{M}_d|\mathcal{D}), \tag{D.7}$$

where Δ is a query assignment, e.g., an inference, $P(\Delta|\mathcal{M}_d, \mathcal{D})$ is its probability distribution given the model representation $\mathcal{M}_d$, and $P(\mathcal{M}_d|\mathcal{D})$ is the probability of model $\mathcal{M}_d$, given the available data. The model probabilities add up to 1, i.e., $\sum_d P(\mathcal{M}_d|\mathcal{D}) = 1$. See e.g., [14] or [15] for further details.

In this study, the radial-basis function kernel is used as weighting function in the BMA formulation, see Eq. D.8.

$$P(\mathcal{M}_d|\mathcal{D}) \propto \prod_{j=1}^{J} \exp(-\lambda_j ||\tilde{u}_j - \tilde{u}_{d,j}||^2), \tag{D.8}$$

where $\tilde{\boldsymbol{u}} = \{\tilde{u}_j\}_{j=1}^{J} = \{\tilde{T}_{nom}, \tilde{N}_c^{\Delta\sigma}\}$ is the normalized vector of nominal thickness and load cycles till fatigue failure for the considered detail, and $\tilde{\boldsymbol{u}}_d = \{\tilde{u}_{d,j}\}_{j=1}^{J} = \{\tilde{T}_{nom,d}, \tilde{N}_{c,d}^{\Delta\sigma}\}$ is the normalized vector of nominal thickness and load cycles till fatigue failure for the d'th experiment in the available data (training set). In this context, the normalizer for nominal thickness is selected as $T_{ref} = \max(\{T_{nom,d}\}_d^D)$, i.e., $\tilde{T}_{nom} = (T_{nom}/T_{ref})$ and $\tilde{T}_{nom,d} = (T_{nom,d}/T_{ref})$; and the normalizer for the number of load cycles till fatigue failure is chosen as $N_{ref}^{\Delta\sigma} = \max(\{N_{c,d}^{\Delta\sigma}\}_d^D)$, i.e., $\tilde{N}_c^{\Delta\sigma} = (N_c/\Delta\sigma^3)/N_{ref}^{\Delta\sigma}$, and $\tilde{N}_{c,d} = (N_{c,d}/\Delta\sigma^3)/N_{ref}^{\Delta\sigma}$. Here, $\Delta\sigma$ is the stress range considered with exponent 3 reflecting steel materials. This normalization is chosen so that the maximal, marginal distance (along the $\tilde{T}_{nom}$ or $\tilde{N}_{c,d}^{\Delta\sigma}$ axis) in the training set is 1, thus $\boldsymbol{\lambda} = \{\lambda_j\}$ controls the strength of the weighting along each axis.

Figures D.2 and D.3 show the relative weighting of the radial-basis function kernel for $\boldsymbol{\lambda} = (4.6, 4.6)^T$. This corresponds to a one-dimensional case where e.g., the experiment with maximum wall thickness in the ensemble is assigned a relative weight of 1%, when evaluating a specimen with a wall thickness corresponding to the minimum in the ensemble. In practice, the $\boldsymbol{\lambda}$ vector is set by cross-validation.

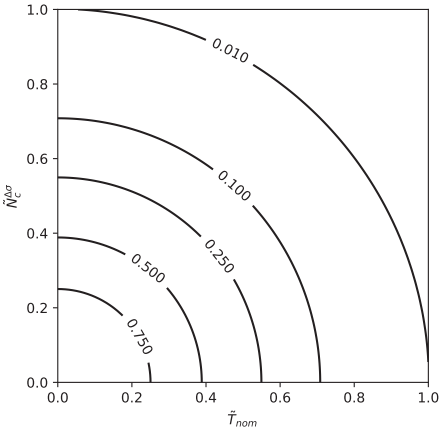


Figure D.2: Bi-variate radial-basis function kernel for $\boldsymbol{\lambda} = (4.6, 4.6)^T$.

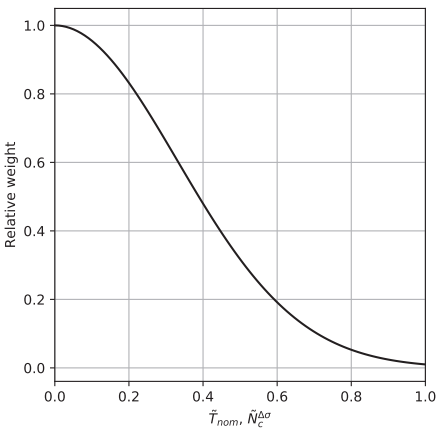


Figure D.3: Marginal radial-basis function kernel for $\lambda_j = 4.6$.

6 CASE STUDY: APPLICATION TO RBI

In this section, a simple example is considered, which illustrates how the normalized fatigue crack growth model corresponding to Fig. D.1 may be applied in the context of RBI for one hot spot. An actual field implementation example is given by [16].

6.1 DATABASE OF FATIGUE EXPERIMENTS

The fatigue experiment results used as basis for the example are taken from [17]. The authors of [17] conducted fatigue tests on six tubular K-joints. For four of the tests (K2, K4, K5 and K6), four crack states have been observed. These states are marked by (1) cracks detected by any available means, (2) visible surface cracks, (3) through thickness cracking, and (4) loss of brace stiffness. For each of the states, the fatigue cycle count, and nominal and maximum stresses are also reported. Alongside these observations, the relative crack depths ($y = a/a_c$) vs relative fatigue lives ($x = N/N_c$) are provided in a graph. a_c (through thickness) and N_c (ultimate fatigue life) refer to state (4) that marks the end of the experiment. Figure 4 from [17] was used to pick x and y values for the development of the model. Two parameter sets influence the weighting of experiments via a radial-basis function. The first parameter set contains the stress ranges, and the second parameter set contains the wall thicknesses, which are 16 mm for all joints.

6.2 ESTIMATION OF THE NORMALIZED FATIGUE CRACK GROWTH MODEL

In this section, the estimation of the normalized fatigue crack growth model is addressed, see Fig. D.1. As mentioned, the model is implemented by use of the probabilistic programming language Stan [10], from which custom code for learning and inference is generated automatically based on the chosen Hamiltonian Monte Carlo inference method.

Figure D.4 shows the hierarchical model with estimated parameters based on the individual data series (K2, K4, K5 and K6). The sub-figures show the data points used for the estimation, the observational models (Eq. D.6b) and the latent functional relationships (Eq. D.6a), along with the corresponding 95% credible bounds. We see that the hierarchical model generally provides a good fit to the data, but the mean residuals have patterns non-conformable with a Gaussian white noise assumption. This is due to the restricted parametric form of the functional relationship, see Eq. D.6a. Moreover, it appears that the individual experiments have distinct noise levels and different crack initiation characteristics.

The modeling of a new fatigue detail is established by averaging the models shown in Fig. D.4 according to their relative relevance defined by the radial-basis function kernel, see Eq. D.8. In this regard, the $\boldsymbol{\lambda}$ vector needs to be fitted. This can be achieved by omitting one experiment at a time in a cross-validation scheme and choosing the $\boldsymbol{\lambda}$ vector that minimizes the average root mean square error of the omitted experiments. The K-joints data set considered in this study only reflect one material thickness, thus the radial-basis function kernel is one dimensional in this case, and $\boldsymbol{\lambda}$ reduces to a scalar value λ. For the K-joints data set, the optimal value of λ is found to be 15.6. The corresponding weighting function is shown in Fig. D.5.

Based on the foregoing, it is now possible to establish the normalized fatigue crack growth model through Eq. D.7. An example is shown in Fig. D.6. The figure shows the observational model and the latent functional relationship, along with the corresponding 95% credible bounds for a fatigue detail with $\tilde{N}_c^{\Delta\sigma} = 0.284$ (a value not represented in the training set), thus $\{\tilde{N}_{c,d}^{\Delta\sigma}\}_{d=1}^{D} = \{0.179, 0.184, 0.307, 1.000\}$. Given that the crack depth is bounded by the interval $[0, 1]$, the function and corresponding credible bounds are truncated at the boundaries. Moreover, the figure shows 10 sample experiments drawn from the latent function.

6.3 RISK-BASED INSPECTION PLANNING

The fatigue reliability problem may be expressed by the fatigue limit state function L:

$$L = R - f, \tag{D.9}$$

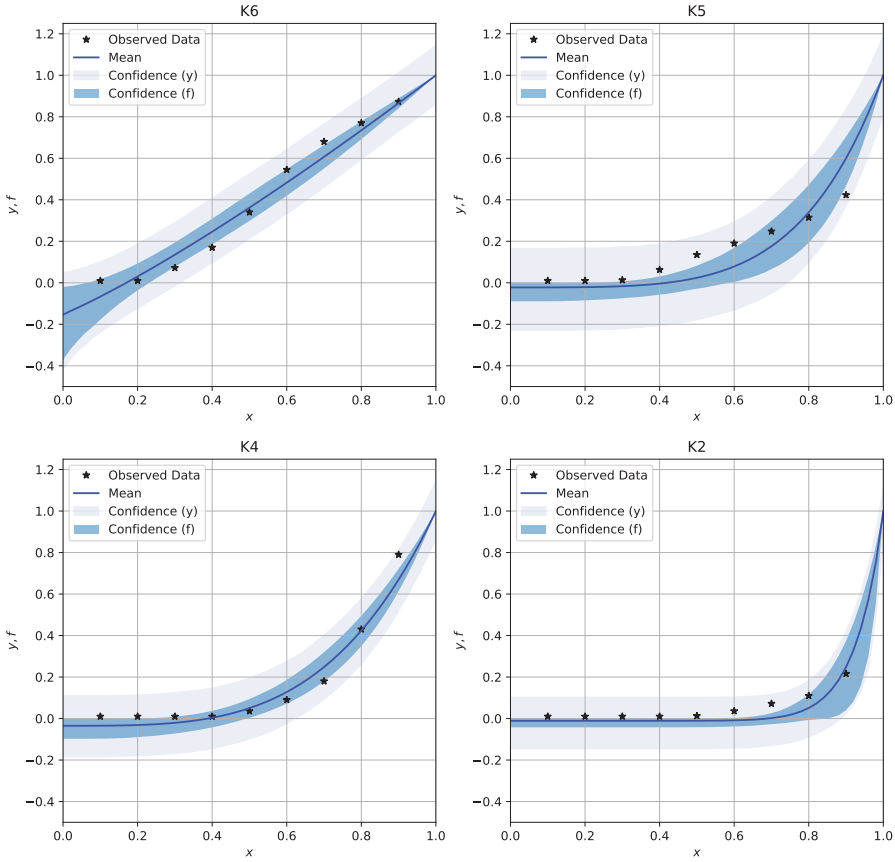


Figure D.4: Hierarchical model fit.

where $\log(R) \sim \mathcal{N}(0, 0.1)$ is the fatigue resistance, and f is the actual (latent), relative fatigue crack depth distribution.

As a first step, the BMA model (Eq. D.7) has to be evaluated at the required x-points using the Monte Carlo samples from the Hamiltonian Monte Carlo inference method. For each x, an approximate continuous density distribution is built from the samples using kernel density estimation (KDE) in order to conduct the reliability estimation. Due to the fixed-point constraints at (0,0) and (1,1), and the presence of a crack initiation phase, some points in the damage evolution cannot be represented by distributions but by deterministic scalars. Furthermore, in order to guarantee that the crack depth cannot regress, each KDE is truncated at an upper bound equal to 1, and a dynamic lower bound equal to the crack depth of the previous time step. A visualization of the dynamically truncated distribution is shown in Fig. D.7 for a fatigue detail with $\tilde{N}_c^{\Delta\sigma} = 0.284$.

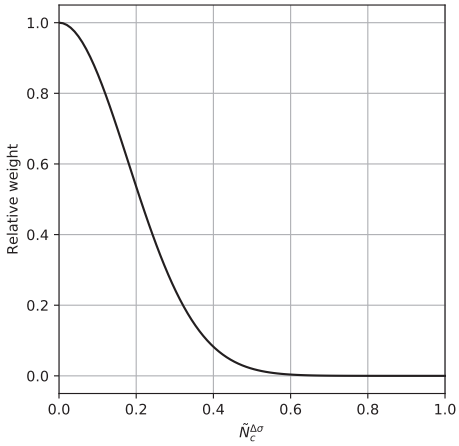


Figure D.5: Marginal radial-basis function kernel for $\lambda = 15.6$.

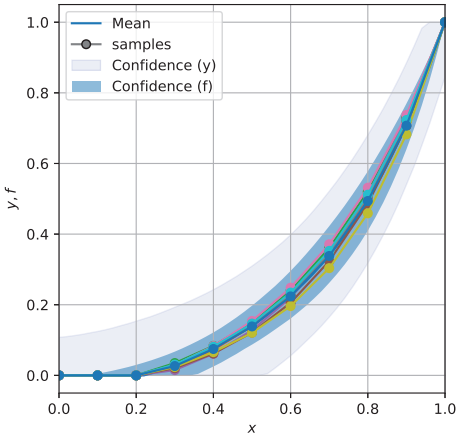


Figure D.6: BMA model for the observational model (y) and the latent functional relationship (f).

Risk-based inspection (RBI) planning requires to update the failure probability $P(a(t) \geq a_c)$ with inspection information, once a pre-defined threshold is reached [1, 16]. In this regard, $a(t)$ is given by the function f as used in Eq. D.9 with a scaling in order to match the dimension of the actual wall thickness and critical crack depth a_c, and a threshold of $P(a(t) \geq a_c) = 10^{-4}$ is applied. Moreover, a basic variant of RBI is considered, where only no damage detection is treated. The reasoning behind this is that if a damage is detected, it will be repaired and in principle the fatigue crack growth starts from initial conditions corresponding to e.g., a new or grind repaired weld.

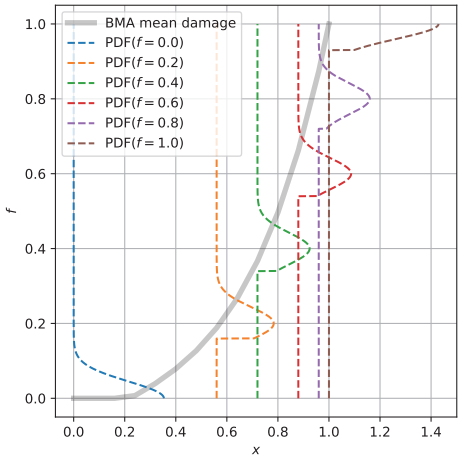


Figure D.7: Examples of dynamically truncated distributions to be drawn from crack size samples.

The updating is performed using Bayesian inference, where the posterior failure probability given no damage detection is calculated, see Eq. D.10.

$$P(a(t) \geq a_c | a(t) < a_d(t)) = \frac{P(a(t) \geq a_c \cap a(t) < a_d(t))}{P(a(t) < a_d(t))}, \tag{D.10}$$

where $a_d(t)$ represents the crack depth detectable in an inspection at time t; t is expressed in terms of years, and the normalized fatigue cycles x must be calibrated accordingly. The commonly used exponential threshold model provides the probability of detection. Equation D.11 is treated as a cumulative distribution function, used to sample a detectable crack depth and to determine by comparison with $a(t)$ if the fatigue damage is detected.

$$\text{CDF}\left(a_d(t)\right) = P_0 \left(1 - \exp\left(-\frac{1}{\lambda_0} a_d(t)\right)\right), \tag{D.11}$$

the parameters for the example case are $P_0 = 1$ and $\lambda_0 = 2.67$.

For the case of $\tilde{N}_c^{\Delta\sigma} = 0.284$, the resulting inspection plan is given in Tab. D.1. The corresponding evolution of the failure probability with time and recurring inspections is depicted in Fig. D.8. Each color represents one stage of inspection information; from no inspection (very left) to last inspection (very right). The unusually short inspection intervals are due to a fatigue design factor of 1, which is chosen in order to show the inspection threshold exceedance several times.

Table D.1: Listing of the inspection plan.

Inspection #	Year	Failure probability	
		Pre-inspection	Post-inspection
1	44.25	1.0133×10^{-4}	0.9990×10^{-5}
2	45.08	1.1527×10^{-4}	1.5106×10^{-5}
3	46.16	1.1329×10^{-4}	1.4801×10^{-5}
4	47.50	1.1027×10^{-4}	0.7243×10^{-5}

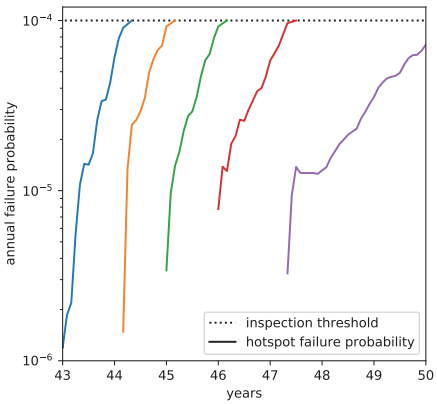


Figure D.8: RBI diagram showing the evolution of failure probability with subsequent inspections and no damage detection.

7 CONCLUSION AND OUTLOOK

The present contribution presents a method for the estimation of a probabilistic normalized fatigue crack growth model and illustrates how this probabilistic model may be applied for the purpose of deriving risk-based inspection plans for fatigue sensitive details in welded offshore structures. The approach presented is based on so-called model-based learning in conjunction with Bayesian estimation. From the present research, we find that the pursued approach is rather robust and very efficient when applied in the context of risk-based inspection planning. Presently, more studies are in process to expose the developed approach to larger data sets covering a broader range of fatigue crack growth experiments collected from the literature. Moreover, investigations are initiated to assess whether non-parametric Bayesian modeling techniques might prove more efficient than the presently applied parametric model-based approach.

ACKNOWLEDGMENT

The authors gratefully acknowledge the funding received from Centre for Oil and Gas – DTU / Danish Hydrocarbon Research and Technology Centre (DHRTC). We would also like to thank Total E&P for providing the support needed to conduct this research.

NOMENCLATURE

a_c	Critical crack depth.
a_d	Crack depth detectable in an inspection.
f	Latent/unobserved functional relationship.
x_i **or** $\boldsymbol{x}$	Realization of random variable(s).
x	Observed normalized load cycles.
y	Observed normalized fatigue crack depth.
$\mathcal{M}_d$	A system/model representation.
X_i **or** $\boldsymbol{X}$	Random variable or set of random variables.
$\mathcal{D}$	Training data set.
$\mathcal{G}$	Directed acyclic graph.
$\boldsymbol{\mathcal{M}}$	Ensemble of system/model representations.
β	Crack growth rate parameter.
γ	Crack initiation parameter.
λ **or** $\boldsymbol{\lambda}$	Parameter(s) of the radial-basis function kernel.
μ	Location parameter.
$\Delta\sigma$	Stress range of fatigue experiment.
σ	Scale parameter.
$\Theta_{\mathcal{G}}$ **or** $\boldsymbol{\Theta}_{\mathcal{G}}$	Parameter or parameter vector related to $\mathcal{G}$.

REFERENCES

[1] M. H. Faber, "Risk Informed Structural Systems Integrity Management: A Decision Analytical Perspective," in *36th International Conference on Ocean, Offshore and Arctic Engineering*, Trondheim, 2017, p. 62715.

[2] D. Straub and M. H. Faber, "Risk Based Acceptance Criteria for Joints Subject to Fatigue," *Journal of offshore mechanics and arctic engineering*, vol. 127, no. May, pp. 150–157, 2005.

[3] J. Tychsen, "Development of normalized stochastic fatigue crack growth model," Maersk Oil, Tech. Rep., 2017.

[4] C. M. Bishop, "Model-based machine learning," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, p. 20120222, 2013.

[5] J. Winn, C. M. Bishop, T. Diethe, J. Guiver, and Y. Zaykov, "Model-based machine learning," (accessed 15 July 2019). [Online]. Available: http://mbmlbook.com/

[6] J. Pearl, *Probabilistic Reasoning in Intelligent Systems. Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[7] D. J. Lunn, A. Thomas, N. Best, and D. Spiegelhalter, "Winbugs - a bayesian modelling framework: Concepts, structure, and extensibility," *Statistics and Computing*, vol. 10, no. 4, pp. 325–337, 2000.

[8] Infer.NET Development Team, "Infer.NET 0.3."

[9] M. Plummer, "JAGS: A program for analysis of bayesian graphical models using gibbs sampling," in *Proceedings of the 3rd international workshop on distributed statistical computing*, vol. 124, no. 125. Vienna, Austria., 2003, pp. 1–10.

[10] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, and A. Riddell, "Stan: A probabilistic programming language," *Journal of Statistical Software*, vol. 76, no. 1, 2017.

[11] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.

[12] U. B. Kjærulff and A. L. Madsen, *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*. Springer New York, 2013.

[13] A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski, "Using bayesian model averaging to calibrate forecast ensembles," *Monthly weather review*, vol. 133, no. 5, pp. 1155–1174, 2005.

[14] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, "Bayesian model averaging: a tutorial," *Statistical science*, vol. 14, no. 4, pp. 382–417, 1999.

[15] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*. Springer New York, 2009.

[16] M. H. Faber, J. D. Sørensen, and D. Straub, "Field Implementation of RBI," *Journal of Offshore Mechanics and Arctic Engineering*, vol. 127, no. August, pp. 220–226, 2005.

[17] D. Stannard, P. Forsyth, and M. Lalani, "New Data On Crack Growth Characteristics Of Fatigue-Loaded Complex Tubular Joints," in *Offshore Technology Conference*, vol. 1988-May. Houston, TX, USA: Offshore Technology Conference, apr 1988, pp. 439–447. [Online]. Available: http://www.onepetro.org/doi/10.4043/5664-MS

PAPER E

ON A SIMPLE SCHEME FOR SYSTEMS MODELING AND IDENTIFICATION USING BIG DATA TECHNIQUES

Sebastian T. Glavind, Juan G. Sepulveda, and Michael H. Faber

The paper has been submitted to
Reliability Engineering & System Safety.

In peer-review
The layout has been revised.

ABSTRACT

In the field of reliability engineering and systems safety it is a common challenge, with basis in a limited set of observations of system performances, to identify the state of the system. Often there are a multitude of different possible system states, including states of damages, which compete in explaining the observations. To account for these in the context of risk-informed management of the systems, the probabilities of the relevant possible different states are needed. In the present contribution, an idea on how this might be supported through big data techniques is presented. Starting point is to establish a knowledge-consistent probabilistic representation of the system, its key performance characteristics, and the observations that may be collected from the system in reality. Monte Carlo simulations are then employed to establish the relevant scenarios of realizations of the random variables describing possible system states, system performance characteristics and observations. Using big data classification on the simulated scenarios, the probabilities of the system being in a given state, given particular outcomes of observations, may then be straightforwardly evaluated. The application of the presented idea is illustrated through two principle examples considering damage identification in structural systems subject to extreme loading.

***Keywords**: systems modeling, observations, big data, system identification, structural damage identification.*

1 INTRODUCTION

As highlighted in [1], a key issue in the governance and management of systems is to ensure that the representation of available knowledge concerning systems consistently accounts for uncertainties, accommodates for the possibility that there might be different (competing) system models, and not least facilitates for utilization of any observation of system performances as a means for updating their representation, see also [2]. Such situations are important to consider in a wide variety of societal decision contexts ranging from structural integrity management, over mitigation of and adaptation to climate change, to hybrid warfare.

Generally, for relevant practical societal decision problems, the systems to be represented are rather complex and may comprise interconnected subsystems with multiple domains of stability and susceptibility to cascading failure event scenarios. The available knowledge of the performances of the systems is subject to significant uncertainties, the representation of which involves high dimensional vectors of random variables subject to causal and stochastic dependencies. In addition, the systems generally exhibit significant non-linear relationships between demands acting on and within the systems

and the systems performances.

In support of establishing adequate system representations, it is possible to take benefit of observations of any observable system performance, e.g., functions and/or services provided by the system. In structural health monitoring, such observations include static and dynamic structural response characteristics. As outlined in [1, 2], a Bayesian approach lends itself to this end. However, given the aforementioned complexities, this comprises a rather challenging task and so far, most achievements in this direction have been targeted on specific applications with respect to both general framing, modeling, and analysis.

In engineering, a variety of different probabilistic approaches for the representation of systems have been developed across different application areas, see e.g., [3–7]; all with the objective to inform the ranking of decision alternatives based on information-consistent models of the systems performances. In the general context of systems identification, and specifically when employing structural health monitoring technologies, one of two avenues is commonly taken when defining a database of realizations of the possible competing systems to consider in the analysis; either extensive numerical or laboratory experiments are performed for a-priori defined systems, see e.g., [8–12].

These approaches generally provide the means for identifying the considered systems or system states, but they do not provide the means for assessing the probabilities of their realizations, and thus the relevance of the systems. Along the same line, they do not account for systems of different origins with response characteristics similar to the studied systems, and they often do not naturally handle the issue of propagating, global system changes, e.g., cascading failure event scenarios, as local system changes, e.g., changes enforced on individual system components, are often the basis for constructing a database used to classify response characteristics.

In the context of structural health monitoring, important contributions to account for possible competing systems have been proposed first in [13] and more recently in [14]. The formulations provided in these works address the core issues of the problem and provide solutions targeted for the considered applications. However, the proposed approaches are not generic and involve extensive numerical efforts. In [15] the representation of uncertain systems is taken up from the novel perspective of applying techniques of big data analysis as a means of enhancing the understanding of the probabilistic characteristics of the system performances. However, in this contribution the observations addressed concern information, which is already accounted for in a given probabilistic model of the system and does as such not contribute to an improved understanding of whether the system model is representative or not.

Following similar ideas, Kurian and Liyanapathirana [16] apply big data

analysis techniques in the context of structural health monitoring as a means of damage identification, e.g., damage detection, localization, and severity assessment. To this end, they take benefit of structured learning as a means to classify observations to different scenarios of possible damages states (competing systems). In their contribution, however, basis is taken in information which only relates to observations from conducted experiments for which the actual damage states are known. In practical applications, the actual system or the actual state of change or damage in a system is not known and thus cannot be utilized as a means for modeling.

2 APPROACH AND OUTLINE

In the present contribution, basis is taken in recent ideas regarding the utilization of big data techniques as a means for modeling and understanding systems as well as for identifying, in probabilistic terms, possible candidates of system representations. To this end, the general idea of modeling from [1] is combined with the approaches in [15, 16] to achieve fully information-consistent probabilistic representations of systems, which account for the possibility of competing system models as well as prior knowledge and information collected through observations of in principle any observable characteristics of the systems performances.

With reference to Fig. E.1, a system is addressed for which a probabilistic model of its performances is available. It might be assumed that such a model is established initially by applying classical bottom-up phenomenological engineering modeling. Furthermore, for the sake of simplicity, it is assumed that an adequate Monte Carlo simulation based technique is applied for the probabilistic analysis of the system, see e.g., [17]. As indicated in Fig. E.1, all available information from the Monte Carlo simulations is gathered and stored in a database.

In addition to the phenomenological engineering model, observations of system performances from "reality" may be utilized – using techniques of big data analysis as a means to improve or calibrate the model – ultimately achieving what is also referred to as digital twins. For common systems and for common system performances, the amount of observable information might be very considerable. For more unusual systems and for rarely occurring systems performances, the opposite holds, and substantial model uncertainties persist.

As highlighted previously, there may be several competing system models both at a given time and for any future time, and these must in principle all be accounted for. The scenarios of the systems performances, which are stored in the model database, provide information about this. Using cluster analysis these scenarios may be analyzed with the objective to identify

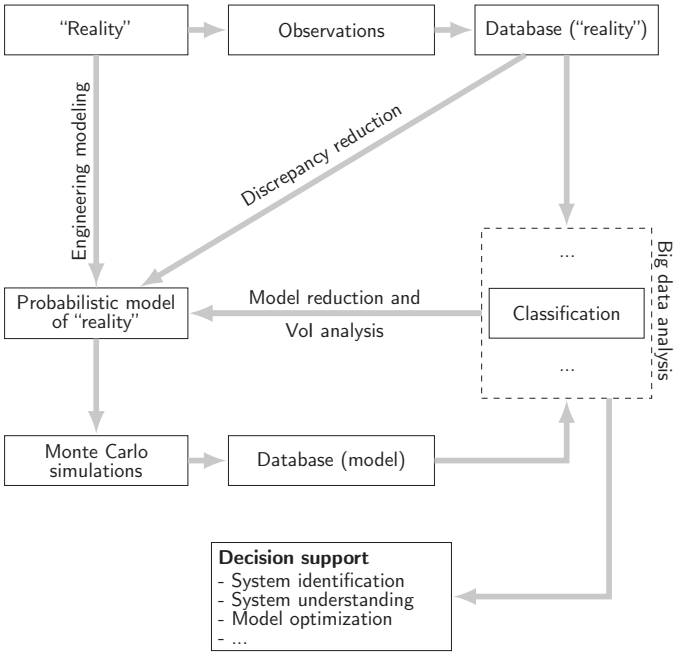


Figure E.1: Framework for system modeling, data mining and evidence based system identification.

and understand possible regularities in the realizations of random variables related to particular system performances of interest, see [15]. The cluster analysis provides information on which uncertainties affect the system performances of interest and which domains of realizations of these contribute to the probabilities of their occurrences.

Finally, and of core importance for the present contribution, supervised classification of the (big) data, generated through the MC simulations and stored in the model database, may be utilized to establish probabilistic relationships between system responses and system states of particular interest. Based on such classifications, it is possible to relate observations of reality to the states of the system in probabilistic terms.

Whereas the approach presented is fully generic and may be applied for the representation of in principle any system, the illustration of the approach in the present contribution is directed on applications in the context of structural integrity management. For simplicity, it is assumed that only one probabilistic model is relevant for the representation of a structure in its intact state. This model might have been optimized based on extensive utilization of observations of the structural performances, using ideas of digital twin technology. The possible competing candidates for the system model come

into the picture in cases where the considered system might have undergone development of damages, e.g., due to extreme load events. It is to this end that the probabilistic classification of system states, which might explain the observations of system performances, is utilized.

It should be highlighted that it is not the intention of this contribution to provide an overview of available research on big data techniques or on approaches for utilization of structural health monitoring in the context of structural integrity management. In all its simplicity, the present contribution might be summarized as providing a rather simple, generic, and robust approach for assessing the probabilities of possible states of a system of interest, given a (possibly limited) set of observations of the performances of the system. The core steps of the suggested approach are:

1. Establish a knowledge-consistent probabilistic model of the considered system, including probabilistic models of the observations of system performances, which may be collected from the systems in reality.

2. Employ Monte Carlo simulation to establish scenarios of realizations of the random variables describing the system, the realizations of system performances, and the corresponding realizations of the observations, which might be collected (in reality) in regard to these.

3. Assess the probabilities of the system performances identified under step 2 and disregard those scenarios for which the probabilities are insignificant in the context of the systems management problem. As an example, if the context at hand regard structural integrity management and the system performances relate to the development of structural damages, then scenarios to be disregarded could be those for which the probabilities are below the highest acceptable probability of failure.

4. Utilize big data classification techniques on the Monte Carlo simulated scenarios as a means for establishing probabilistic relationships between outcomes of observations and system performances of particular interest. As an example, in the context of structural integrity management, such states could be different possible damage states.

As indicated in the foregoing, the objective of the present contribution is not to explore different technologies of big data analysis but rather to show how state-of-the-art big data techniques might be applied in support of systems modeling and analysis. The interested reader is referred to e.g., [18] for a recent review on the application of big data technologies in structural health monitoring. For ease of reference though one such generally well performing technique for classification, namely tree-based classification, is outlined in Sec. 3, along with recommendations on model selection for such techniques. In Sec. 4, the proposed approach is illustrated on two principle examples

from structural engineering. Again, it should be underlined that practical relevance of the two examples is not an objective of the present paper; the examples are principle examples of systems exhibiting the main characteristics of complex systems in general, including multiple possible relevant performances and non-linear relationships between system demands and system performances.

The first example is devised such as to present the general idea in a tractable and easily reproducible manner. In this example, a simple moment resisting frame subject to extreme loads is considered. Based on observations of deflections of the frame during the extreme loading, the task is to identify whether the structure is subject to damage and to assess the probabilities of relevant damage states.

The second example considers a similar case but for a more complex 3-story, 3-bay frame structure for which the number of interesting system performance states (damage states) is significantly increased compared to the first example. Again, due to the idealizations introduced on the mechanical modeling side, to ensure tractability, this example is also not of direct practical relevance but rather intended to illustrate some of the robustness properties of the proposed methodology, as well as to indicate how the proposed methodology might also support value of information analysis in the context of monitoring supported structural integrity management. Finally, in Sec. 5, the findings and conclusions are summarized and suggestions for further research are given.

3 TREE-BASED CLASSIFICATION AND MODEL SELECTION

In supervised learning, a mapping from system inputs to system outputs is defined based on a database of observations of input-output pairs. If the outputs are real numbers, i.e., variables with a continuous sample space, then the task is called regression, and if the outputs are integer valued, i.e., variables with a discrete sample space, then the task is called classification.

In the following, tree-based supervised learning is introduced in Sec. 3.1. Subsequently, the focus is directed on how to reduce the uncertainty in model predictions by forming an ensemble of tree-based models in Sec. 3.2. Most learning algorithms come with a set of hyper-parameters that needs to be specified in order to get the best possible predictive performance when mapping new inputs to outputs. In Sec. 3.3, it is discussed how the hyper-parameters of general machine learning algorithms may be specified using Bayesian optimization with a Gaussian process prior.

3.1 CLASSIFICATION AND REGRESSION TREES

Classification and regression trees (CART) [19, 20], also referred to as decision trees in the machine learning literature, partition the input space into a set of disjoint regions (hyper-cubes) $\{R_j\}_{j=1}^{J}$ by recursively applying binary splits on an input variable, as shown in Fig. E.2. The figure illustrates the mapping of an input vector $\boldsymbol{x}$ to a region R_j as a sequential decision-making process corresponding to the traversal of a binary tree, which is a tree that splits into two branches at each node, where the regions R_j appear as leaf nodes [21].

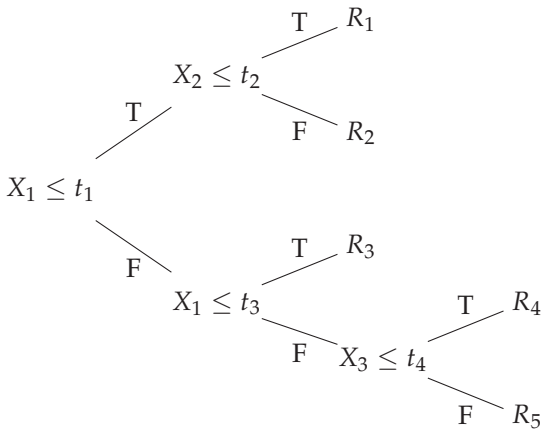


Figure E.2: Principle classification and regression tree. The labeling of the edges mark whether the presiding statement is true (T) or false (F), and $\{t_v\}_{v=1}^{V}$ is the set of threshold values used in the binary splits.

Given a training data set $\mathcal{D} = \{\hat{\boldsymbol{X}}, \hat{\boldsymbol{y}}\} = \{\boldsymbol{x}[n], y[n]\}_{n=1}^{N}$ of N i.i.d. observations of input-output pairs, learning in a CART setting amounts to defining the tree structure, including which input variable is chosen at each node and the corresponding splitting criterion, and the prediction model for each region. The regional prediction models define the task of the machine learning application; for regression problems, an appropriate constant might be assigned to the region, and for classification problems, a class label may be assigned to the region, i.e.,

$$\boldsymbol{x} \in R_j \Rightarrow f(\boldsymbol{x}) = c_j, \tag{E.1}$$

where c_j is an appropriate constant or class label depending on the application. Thus, a tree can be formally expressed as

$$\mathcal{T}(\boldsymbol{x}; \boldsymbol{\Theta}) = \sum_{j=1}^{J} c_j \mathbb{I}[\boldsymbol{x} \in R_j], \tag{E.2}$$

where $\mathbb{I}$ is the indicator function, and $\boldsymbol{\Theta} = \{R_j, c_j\}_{j=1}^{J}$ is the parameter vector, see e.g., [19] for further details on CARTs and their training. Moreover, Appendix A gives some recommendations on how to evaluate the performance of general classifiers.

3.2 ENSEMBLE LEARNING WITH BOOSTING

CARTs as presented in the previous section are simple, approximately unbiased models that suffer from high predictive variance. Thus, a small change in the data can result in a very different series of splits and thus predictions. The literature contains a variety of general-purpose procedures for reducing the variance of statistical learners, which e.g., leverage simple models like trees. Among the most popular are bagging and boosting [19, 22].

In boosting, an ensemble of tree models is established, but opposite to bagging where the trees are grown in parallel, they are grown sequentially, i.e., each tree is grown using information from previous trees in the sequence. That is, at each stage, a new CART is estimated that focuses on the errors of the current ensemble model and consequently add this new CART to the ensemble model. After B trees are grown, the boosted (average) predictor becomes

$$f(\boldsymbol{x}; \hat{\boldsymbol{\theta}}) = \sum_{b=1}^{B} \mathcal{T}(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_b), \tag{E.3}$$

where $\hat{\boldsymbol{\theta}}_b$ is the estimated parameter vector for one tree model. A popular variant of boosting that is continuously coming out in the top of machine learning competitions, like Kaggle,[1] is gradient boosting (machines). This is not at least to due efficient implementations, such as XGBoost [23].

Tree-based ensemble learners come with a set of hyper-parameters that needs to be tuned in order to gain the best possible performance. Among the set of parameters are the tree depth J, the number of trees in the ensemble B, the amount of shrinkage when boosting, and the extent of subsampling among the inputs and training points when fitting the tree models [19, 22, 23]. Hyper-parameter tuning is therefore discussed in Sec. 3.3.

3.3 BAYESIAN OPTIMIZATION FOR MODEL SELECTION

In this section, the problem of finding a global minimizer of a function f defined by covariate(s) $\boldsymbol{x}$ is considered, i.e.,

$$\boldsymbol{x}_{\min} = \arg\min_{\boldsymbol{x} \in \boldsymbol{X}} f(\boldsymbol{x}). \tag{E.4}$$

Bayesian optimization (BO) is a sequential model-based approach for optimizing a loss function, which is computationally expensive to evaluate

[1] `https://www.kaggle.com/`

and/or has no closed-form expression, but from which (noisy) observations can be obtained [24]. BO techniques are among the most efficient optimization techniques in terms of number of functional evaluations required, due to their use of Bayesian updating as

$$p(f|\mathcal{D}) \propto p(\mathcal{D}|f)p(f),$$

where $\mathcal{D} = \{\hat{\boldsymbol{X}}, \hat{\boldsymbol{y}}\} = \{\boldsymbol{x}[n], y[n]\}_{n=1}^{N}$ is a data set of observations of the loss function [25].

The example considered in the present section is the optimization of hyper-parameters for a general machine learning model, like gradient boosting, where the objective is to find the hyper-parameters that result in the lowest validation loss, see Appendix A. Traditionally, strategies such as manual-, grid- and random-search are employed for the optimization, where random-search is found superior to grid-search [26], but BO techniques have been show to outperform manual- and random-search in terms of both performance and efficiency, see e.g., [27–29].

Bayesian optimization using Gaussian processes (GPs) leverage Bayes rule to build a surrogate model of the loss function (validation loss) with a prior over functions and combine it with new observations to form a posterior over functions, in an online fashion. This permits a utility-based selection of the next point to sample from the loss function, which should account for the trade-off between exploration (sampling from areas of high uncertainty) and exploitation (sampling from areas that are likely to provide an improvement over the current best setting $\boldsymbol{x}_{\text{min}}^{(t)}$) [24, 25]. See [21, 30] for details on GPs, and their training.

In order to conduct a utility-based selection of the next sampling point, a utility function is needed. Such functions are commonly referred to as acquisition functions in the BO literature. The acquisition function takes the mean and variance information of the predictions into account to model the utility of new sampling points, such that high acquisition values correspond to potentially low loss values, either because the prediction is low or the uncertainty is great, or both. The argmax value of the acquisition function is chosen as the next sampling point of the loss function, and the process is repeated, considering the data set $\mathcal{D}$ augmented with the new sample point $\{\boldsymbol{x}[N+1], y[N+1]\}$ [25].

Acquisition functions traditionally used in relation to BO are (i) the probability of improvement, (ii) the expected improvement, and (iii) the lower confidence bound. See e.g., [24] for a detailed listing of acquisition functions used in practice as well as how to conduct the optimization. Moreover, note that the choice of probabilistic model is often considered more important than the choice of acquisition function [24].

4 NUMERICAL EXAMPLES ON STRUCTURAL DAMAGE IDENTIFICATION

4.1 INTRODUCTION

In order to illustrate the application of the foregoing ideas and techniques, two systems from structural engineering are considered, namely a simple moment resisting, portal frame structure and a more complex 3-story, 3-bay moment resisting frame structure. In both cases, the structure is subjected to concentrated horizontal load(s) in the beam element(s) and vertical load(s) at mid-span of the beam element(s); the former representing annual extreme wind loading, and the latter representing daily extreme operational loading. Appendix B contains a detailed description of the numerical modeling of the two systems.

4.2 MONTE CARLO SIMULATIONS

In the present study, Monte Carlo simulation is initially applied for generating the realizations of random variables (patterns) and the corresponding failure scenarios. As the number of realizations of the different failure scenarios in the resulting database may vary significantly – and for some scenarios be very low – a strategy for increasing these is devised. Thus, for each realization of a failure scenario, a random-walk Markov chain Monte Carlo (MCMC) sampler is initiated to produce additional realizations of that failure scenario. Note that these additional MCMC samples are not used to alter the probability of a failure scenario but only contribute with additional realizations to balance the database in the failure scenarios. This scheme is similar to the scheme used in subset simulation [31, 32]. In this regard, a new sample is accepted if it belongs to the considered failure scenario, i.e., a hard-assignment, and the standard deviation of the random-walk, Gaussian proposal distribution is tuned to give an acceptance rate of approximately 0.25. Moreover, in order to reduce the auto-correlation of the resulting samples, only every fifth sample is used for further analysis. This is referred to as thinning, see e.g., [33] for further details on MCMC simulation.

4.3 EXAMPLE I: MOMENT RESISTING PORTAL FRAME STRUCTURE

This example considers a portal frame structure subjected to a concentrated horizontal and vertical load, i.e., P_1 and P_2, respectively [34, 35]. The model has five nodes and four elements, and a total of 8 hinge locations are considered, which are denoted by $1, 2, \ldots, 8$ in Fig. E.3. The moment of inertia of

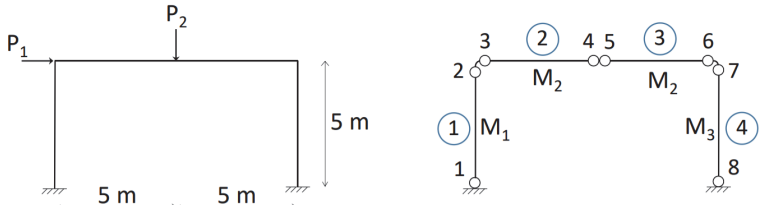


Figure E.3: Left: Portal frame structure of Example I. Right: Structural model properties.

the cross sections are 4.412×10^{-5} m^4 and 4.770×10^{-5} m^4 for the beam and the column, respectively, both with a Youngs modulus of 210 GPa.

A load case is considered, which is dominated by the horizontal loading, where the annual maximum horizontal load P_1 and the daily maximum vertical load P_2 are assumed to follow Weibull distributions. For P_1 the expected value and coefficient of variation (CoV) are 21.75 kN and 0.30, respectively, and for P_2 the expected value and CoV are 28.70 kN and 0.10, respectively. The yield stress capacities of the nodes are assumed to follow Log-normal distributions with expected values equal to 250 MPa and CoVs equal to 0.05.

Based on a total of 10^8 Monte Carlo simulations, Fig. E.4 and Tab. E.1 show the resulting failure scenarios $\{FS_c, c = 1, 2, .., N_{FS}\}$, along with the corresponding numbers of realizations N_F and probabilities of occurrences P_F. In this regard, a state vector describes the order in which the plastic hinges are formed in each state of the simulation, where 0 indicates that the node does not become plastic, 1 indicates that the node is the first to form a plastic hinge, 2 indicates that the node becomes plastic after the first redistribution of internal forces, and so on. After completed analysis, the state vector contains the so-called failure scenarios.

Figure E.5 shows the MC realizations of the random variables leading

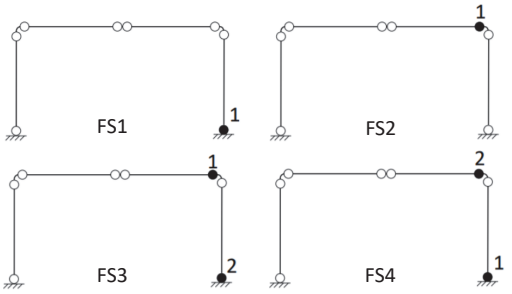


Figure E.4: Failure scenarios identified in Example I.

Table E.1: Failure scenarios identified in Example I.

	1	2	3	4	5	6	7	8	N_F	P_F
FS_1	0	0	0	0	0	0	0	1	942	9.42×10^{-6}
FS_2	0	0	0	0	0	1	0	0	9,881	9.88×10^{-5}
FS_3	0	0	0	0	0	1	0	2	171	1.71×10^{-6}
FS_4	0	0	0	0	0	2	0	1	123	1.23×10^{-6}
$\sum$									11,117	

to the different failure scenarios and their expected values. It is observed that the failure scenarios are generally governed by large positive realizations of P_1 and to a lesser degree P_2. Note in this regard that all realizations of P_1 reflect this tendency ($P_1 > 0$), i.e., it is significant, whereas this in not the case for P_2. Moreover, the mean values of the realizations of M_1 are approximately equal to zero for all failure scenarios. This indicates that this variable is insignificant and in principle might be omitted. The realizations of M_2 and M_3 alternate between two bounding patterns, i.e., (i) the realizations of M_2 are equal to zero, and the realizations of M_3 take on a large negative value; and (ii) the realizations M_2 take on large negative values, and the realizations of M_3 are equal to zero. Bounding pattern (i) reflects a failure scenario governed by a weak right column, and (ii) reflects a failure scenario governed by a weak beam, see Figs. E.3 and E.4.

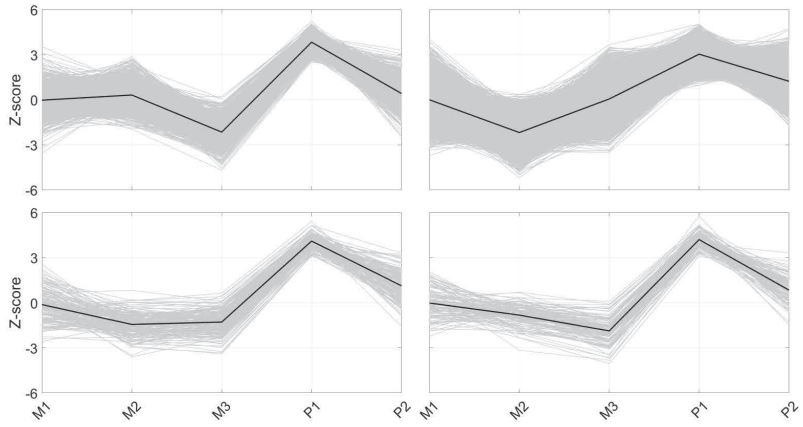


Figure E.5: Realizations in standard normal space of the random variables (gray) and expected values (black) in Example I. Top-left: FS_1. Top-right: FS_2. Bottom-left: FS_3. Bottom-right: FS_4.

DAMAGE IDENTIFICATION

In this section, observations are utilized for the purpose of identifying whether the structure under an extreme load event is loaded to failure, and if so, which failure scenarios might have developed. To this end, observations of P_1 and P_2 are considered, along with the corresponding horizontal displacements at the upper right corner of the frame (δ_1) and vertical displacements at the mid-span of the beam element (δ_2), at the final stage of a failure scenario, see e.g., Fig. E.3. In this regard, the structure may still be undamaged (baseline, FS_0), or it may have transitioned to one of the failure scenarios considered above, i.e., $\{FS_c, c = 1, \ldots, 4\}$. Thus, we seek an identification of the existence and probable location of damage, represented by the most likely failure scenarios.

A training set of 8,000 realizations of each failure scenario and a test set of 2,000 realizations are considered. That is, in total 40,000 training realizations and 10,000 test realizations (including the scenario with no damage). A gradient boosting classifier is fitted to the training database based on the cross-entropy loss function, as described in Secs. 3.1 and 3.2, using functionalities from the publicly available toolbox XGBoost [23]. In this regard, the hyper-parameters of the classifier are chosen based on 5-fold cross-validation using GP-based Bayesian optimization with a squared exponential kernel and the expected improvement acquisition function, see Sec. 3.3, using functionalities from the publicly available toolbox GPyOpt [36]. Moreover, to maximize the performance of the classifier, the principal components of the original training and test set are used as inputs to the classifier, as CARTs generally prefer orthogonal inputs, see e.g., [37].

The test set confusion matrix of the final classifier is shown in Fig. E.6. The final classifier has an overall training set accuracy of 1.00, and a test set accuracy of 0.97. Furthermore, the macro F1-score as well as the corresponding macro precision and recall are 0.97, and the test set cross-entropy is 0.097. For this simple example, it appears that the classifier can almost perfectly classify the test set examples, thus the main diagonal of the confusion matrix contains most of the test data. However, when it misclassifies test data, they typically belong to FS_2, FS_3 or FS_4, see Fig. E.6. Considering the corresponding rows of the confusion matrix, it is seen that FS_2 is sometimes mistaken for FS_3, FS_3 is mistaken for FS_2 or FS_4, and FS_4 is mistaken for FS_1 or FS_3. This makes sense, as FS_2 may lead to FS_3 and FS_1 may lead to FS_4, and the only difference between FS_3 and FS_4 is the order in which the hinges form, see Fig. E.4. These effects may further be emphasized by setting the elements on the main diagonal in Fig. E.6 to zero and rescaling the color map, leading to the error matrix in Fig. E.7. In this figure, the error pattern of the classifier is easily identified, e.g., also along the columns, when the classifier predicts FS_4 and it makes an error, the misclassified realization most likely belongs to

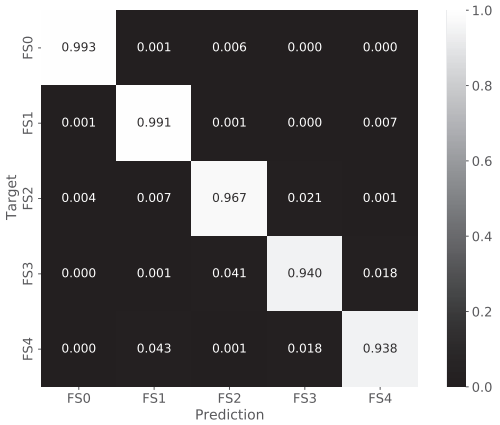


Figure E.6: Scaled test set confusion matrix when considering 8000 training samples of each failure scenario in Example I. The confusion matrix is scaled by the number of test samples of each failure scenario, such that the diagonal terms reflect the accuracy related to each failure scenario.

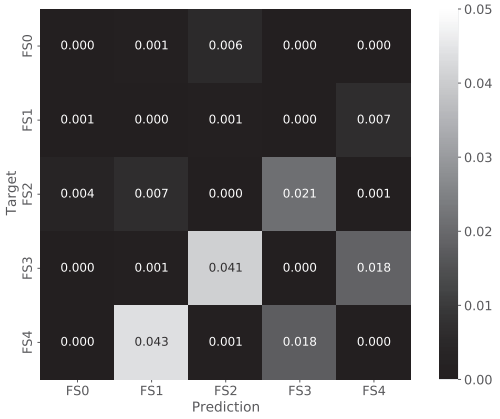


Figure E.7: Scaled test set error matrix corresponding to the confusion matrix in Fig. E.6.

FS_1 or FS_3, thus in this case there are symmetries in the error pattern. Note that this is not always the case, see e.g., [37].

So far, the performance of the classifier has been assessed mostly by considering hard class assignments through the performance metrics and the confusion matrix, but for probabilistic damage identification the multi-class posterior probabilities for each realization are needed. These are provided in Tab. E.3 for the random realizations of Tab. E.2. In Tab. E.3, the conditional

class probabilities are provided first, followed by the posterior class probabilities in parenthesis, which thus account for the prior class probabilities in Tab. E.1.

Again, it appears from the conditional class probabilities that the classifier in general assigns most probability mass to the target label, but additional information on the certainty with which the classifier assigns the hard class assessments is gained from Tab. E.3, e.g., the off-diagonal elements (probability masses assigned to classes other than the target class) are generally small for the conditional class probabilities of the realizations in Tab. E.2, except for the first realization with target label FS_3, where FS_4 also attracts significant probability mass. For decision support, the posterior class probabilities in parenthesis should be considered in combination with associated class utilities in accordance with Bayesian decision theory [38], and the axioms of utility theory [39], to choose the decision alternative that optimizes the expected utility.

Table E.2: Realizations of random variables for multi-class posterior probability estimation in Example I.

		Input *			
		P_1	P_2	δ_1	δ_2
Target	FS_0	0.124	0.961	-0.009	0.002
		-0.973	0.219	-0.007	0.001
		-0.808	0.287	-0.007	0.002
	FS_1	3.447	0.865	-0.013	0.004
		3.645	-0.296	-0.012	0.004
		3.505	1.684	-0.013	0.004
	FS_2	3.257	1.946	-0.013	0.004
		2.331	2.035	-0.012	0.003
		2.916	1.019	-0.012	0.004
	FS_3	4.063	1.303	-0.014	0.004
		4.067	1.630	-0.014	0.004
		3.266	1.559	-0.013	0.004
	FS_4	4.084	0.207	-0.014	0.004
		3.763	0.611	-0.014	0.004
		4.021	1.41	-0.014	0.004

* The realizations are shown in standard normal space.

4.4 EXAMPLE II: 3-STORY, 3-BAY MOMENT RESISTING FRAME STRUCTURE

This example considers a 3-story, 3-bay frame structure subjected to concentrated horizontal loads and vertical loads at the mid-span of the beam elements, see Fig. E.8. The structural analysis model has 30 elements and a total of 60 hinge locations are considered. Note that for reasons of modeling

Table E.3: Multi-class probabilities for realizations in Tab. E.2. The conditional class probabilities are provided first, followed by the posterior class probabilities in parenthesis, which thus account for the prior class probabilities in Tab. E.1.

		Prediction				
		FS_0	FS_1	FS_2	FS_3	FS_4
Target	FS_0	≈ 1 (≈ 1)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	≈ 0 (≈ 0)
		≈ 1 (≈ 1)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	≈ 0 (≈ 0)
		≈ 1 (≈ 1)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	≈ 0 (≈ 0)
	FS_1	≈ 0 (0.010)	0.992 (0.989)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	0.008 (0.001)
		≈ 0 (0.202)	0.997 (0.798)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	0.002 (≈ 0)
		≈ 0 (0.140)	0.978 (0.858)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	0.022 (0.002)
	FS_2	≈ 0 (≈ 0)	≈ 0 (≈ 0)	0.928 (0.998)	0.072 (0.001)	≈ 0 (≈ 0)
		≈ 0 (≈ 0)	≈ 0 (≈ 0)	≈ 1 (≈ 1)	≈ 0 (≈ 0)	≈ 0 (≈ 0)
		≈ 0 (0.010)	≈ 0 (≈ 0)	0.995 (0.990)	0.005 (≈ 0)	≈ 0 (≈ 0)
	FS_3	≈ 0 (0.480)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	0.802 (0.442)	0.198 (0.078)
		≈ 0 (0.243)	≈ 0 (≈ 0)	≈ 0 (0.005)	≈ 1 (0.751)	≈ 0 (≈ 0)
		≈ 0 (0.007)	≈ 0 (≈ 0)	≈ 0 (0.012)	≈ 1 (0.981)	≈ 0 (≈ 0)
	FS_4	≈ 0 (0.032)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	≈ 1 (0.968)
		≈ 0 (0.021)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	≈ 1 (0.979)
		≈ 0 (0.001)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	≈ 0 (≈ 0)	≈ 1 (0.999)

convenience the beam elements are split into two equal model elements at the location of the vertical loads.

A load case dominated by the horizontal loading is considered, where the annual maximum horizontal loads $\{P_i^H, i = 1, 2, 3\}$ are assumed to follow Weibull distributions with expected values equal to 21.75 kN, 26.97 kN and 16.97 kN, respectively, and coefficients of variation (CoV) equal to 0.30. The horizontal loads are modeled as dependent random variables with a correlation factor of 0.40. The daily maximum vertical loads $\{P_{ij}^V, i = 1, 2, 3, j = 1, 2, 3\}$ are also assumed to follow Weibull distributions with expected values equal to 34.34 kN and CoVs equal to 0.05. The vertical loads are modeled as independent random variables. The yield stress capacities at the location of the nodes are assumed to follow Log-normal distributions with an expected value of 250 MPa and a CoV of 0.05. The yield stress capacities are modeled as dependent random variables by assuming a correlation coefficient equal to 0.8 for elements sharing the same design variable, according to Tab. E.4, and 0.4 otherwise. Thus, the total set of random variables involved in the reliability problem amounts to 33, i.e., 3 horizontal load components, 9 vertical load components, and 21 yield stress capacities, whereof 9 corresponds to beam elements and 12 to column elements. Note that as the beam elements are split into two model elements at the location of the vertical loads, as mentioned above, a total of 18 model elements correspond to beams. Thus, two consecutive beam model elements share the same random variable cor-

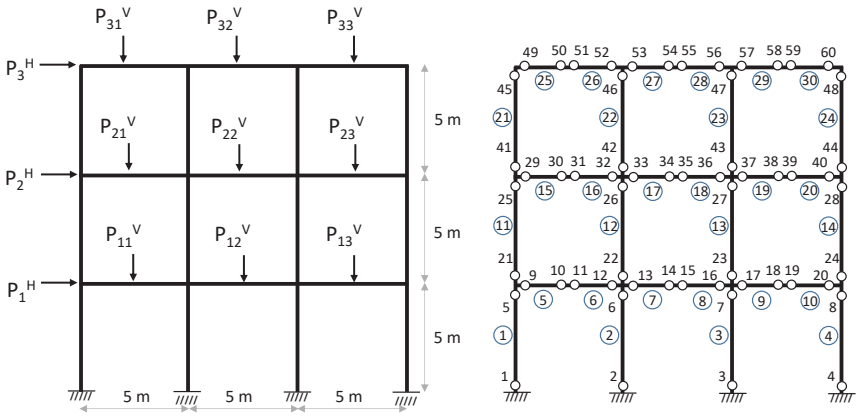


Figure E.8: Left: 3-story, 3-bay frame structure of Example II. Right: Structural model properties.

responding to the yielding stress capacity.

For the calibration process, the target failure probabilities for the columns and beams are 10^{-5} and 10^{-4}, respectively. In this case, 9 design variables are considered in the optimization process, and their resulting numerical values and the associated elements are indicated in Tab. E.4. The Young modulus for all elements is equal to 210 GPa.

Figure E.9 and Tab. E.5 show the resulting failure scenarios $\{FS_c, c = 1, \ldots, N_{FS}\}$, along with the number of realizations N_F that lead to each failure scenario and the corresponding probabilities of occurrences P_F, resulting from a total of 10^8 Monte Carlo simulations. Note that a total of 913 failure scenarios are identified from the MC simulations, but the assessments in

Table E.4: Design variables and associations considered in Example II.

Design variable	Value $[10^{-5}$ m$^4]$	Model elements
X_1	3.62	1, 4
X_2	2.19	11, 14
X_3	1.64	21, 24
X_4	5.73	2, 3
X_5	3.83	12, 13
X_6	1.45	22, 23
X_7	5.02	5, 6, 7, 8, 9, 10
X_8	3.63	15, 16, 17, 18, 19, 20
X_9	2.09	25, 26, 27, 28, 29, 30

this example, only considers failure scenarios for which $P_F \geq 10^{-6}$, i.e., failure scenarios with a relevant contribution to the system failure probability. Moreover, due to the large number of nodes, only the nodes that fail in the failure scenarios appear in Tab. E.5.

Table E.5: Failure scenarios identified in Example II for which $P_F \geq 10^{-6}$. Note that the nodes indicated in the upper row only include those for which failure are involved in the failure scenarios.

	2	4	12	16	26	28	32	46	48	52	N_F	P_F
FS_1	0	0	0	0	0	0	0	0	0	1	8642	8.64×10^{-5}
FS_2	0	0	0	0	0	0	0	0	1	0	419	4.19×10^{-6}
FS_3	0	0	0	0	0	0	0	1	0	0	626	6.26×10^{-6}
FS_4	0	0	0	0	0	0	1	0	0	0	5811	5.81×10^{-5}
FS_5	0	0	0	0	0	0	2	0	0	1	119	1.19×10^{-6}
FS_6	0	0	0	0	0	0	1	0	0	2	151	1.51×10^{-6}
FS_7	0	0	0	0	0	1	0	0	0	0	286	2.86×10^{-6}
FS_8	0	0	0	0	1	0	0	0	0	0	193	1.93×10^{-6}
FS_9	0	0	1	0	0	0	0	0	0	0	6390	6.39×10^{-5}
FS_{10}	0	0	1	0	0	0	2	0	0	0	271	2.71×10^{-6}
FS_{11}	0	0	1	2	0	0	0	0	0	0	236	2.36×10^{-6}
FS_{12}	0	0	2	0	0	0	1	0	0	0	341	3.41×10^{-6}
FS_{13}	0	1	0	0	0	0	0	0	0	0	317	3.17×10^{-6}
FS_{14}	0	2	1	0	0	0	0	0	0	0	111	1.11×10^{-6}
FS_{15}	1	0	0	0	0	0	0	0	0	0	185	1.85×10^{-6}
$\sum$											24,098	

Figure E.10 shows the MC realizations of the random variables leading to FS_1, FS_3, and FS_{14}, and their expected values. The random variables corresponding to the yield stress capacities are indicated with the number of the associated element. For FS_1 and FS_3, it appears that failures are generally governed by low capacities of the beam element 25–26 (the failing beam) and column element 22 (the failing column), respectively, along with large valued realizations of P_3^H (the horizontal load at story 3) and to a lesser degree P_1^H, P_2^H, and P_{31}^V (the vertical load at story 3, left bay). Regarding FS_{14}, it is observed that failures are generally governed by low capacities of the column elements 1 and 4 (the lower leftmost and rightmost column) and beam element 5–6, along with large valued realizations of P_1^H (the horizontal load at story 1) and to a lesser degree P_2^H, and P_3^H. This is the most frequent failure scenario involving both a beam and column failure.

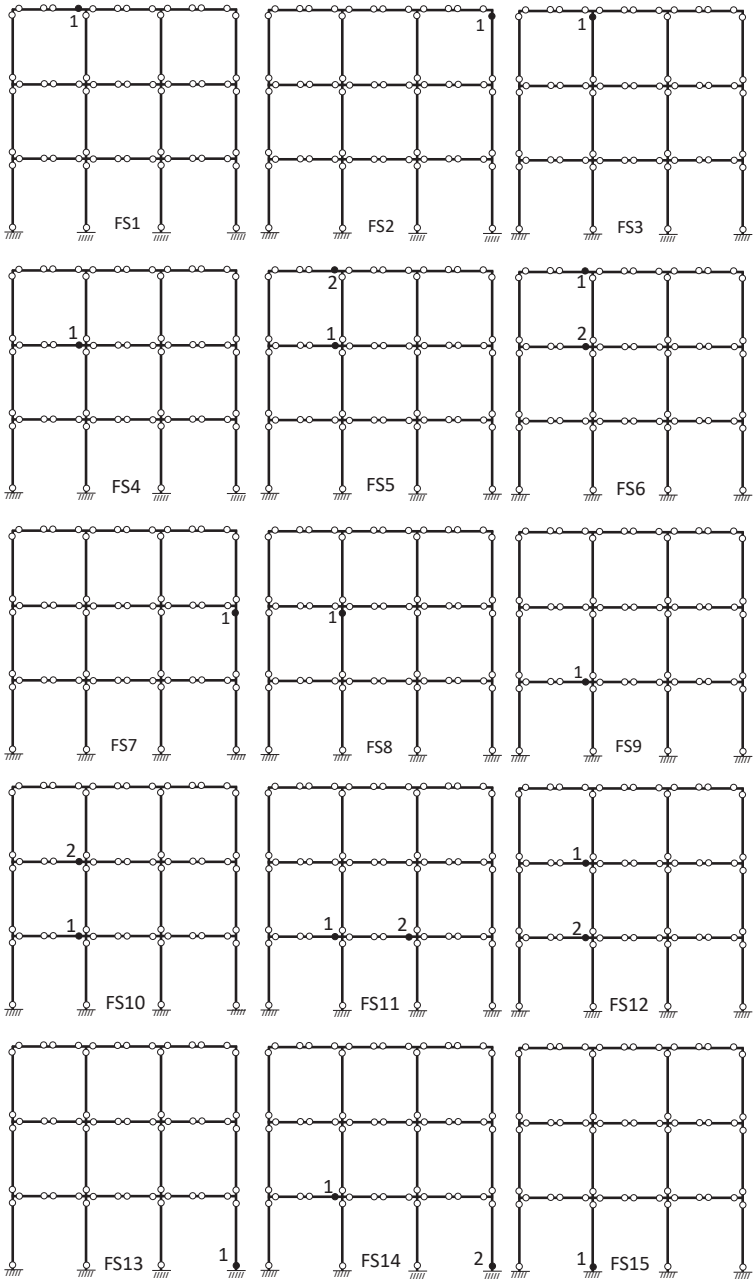


Figure E.9: Failure scenarios identified in Example II for which $P_F \geq 10^{-6}$.

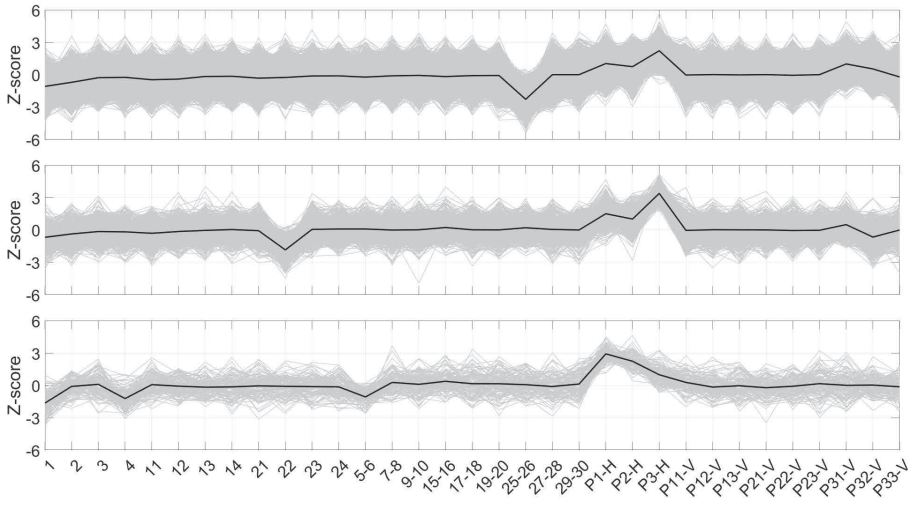


Figure E.10: Realizations in standard normal space of the random variables (gray) and expected values (black) in Example II. Top: FS_1. Middle: FS_3. Bottom: FS_{14}.

DAMAGE IDENTIFICATION

This section considers the identification of the 15 failure scenarios in Fig. E.9 through observations of loads and displacements. To this end, different cases are assumed with respect to the number and locations of load and displacement observations, and the associated uncertainties. Again, the observations are assumed to be collected at the occurrence of an annual extreme load event, and the target of the analysis is to assess whether the structure is still undamaged (baseline, FS_0) or whether one of the failure scenarios considered above, i.e. $\{FS_c, c = 1, \ldots, 15\}$, have occurred. The classification approach is the same as the one used in Example I, i.e. using a gradient boosting classifier, and setting the hyper-parameters based on 5-fold cross-validation using GP-based Bayesian optimization with a squared exponential kernel and the expected improvement acquisition function, see also Sec. 3 for further details.

Perfect information This assessment considers observations of all load variables, along with the corresponding horizontal displacements of the beam elements, measured on the right hand side of the structure, and vertical displacements at mid-span of the beam elements, at the final stage of a failure scenario, see e.g., Fig. E.8. Like in Example I, the realizations are taken directly from the structural analysis program and are thus free of potential measurement uncertainties (noise).

To target the analysis, a sensitivity analysis is initially undertaken to assess the influence of the size of the training data set. In this regard, three settings are considered: (i) a training set of 8000 realizations of each failure scenario, as in Example I, i.e., 128,000 training realizations in total; (ii) a training set size of 4000 realizations of each failure scenario, i.e., 64,000 training realizations in total; and (iii) a training set of 2000 realizations of each failure scenario, i.e., 32,000 training realizations in total. In all situations, the same test set of realizations is considered, which is comprised of 2000 realizations of each failure scenario, i.e., 32,000 test realizations in total.

The results of the sensitivity analysis are summarized in Fig. E.11 in terms of the test set accuracy, the macro F1-score, and the cross-entropy. The corresponding error matrices appearing in Fig. E.12. Figure E.11 shows that the test set performance decreases steadily as the number of training examples are reduced, thus the accuracy and the macro F1-score decrease and the cross-entropy (loss) increases. At the same time, the classifier training time is approximately halved as the training samples are halved. Thus, there is a trade-off between performance and training time; up to a certain level, the performance may be increased by adding more training samples, but this comes with the cost of increased training time. Within the range of training samples considered in Fig. E.11, it is concluded that the classifier globally performs rather robust, as e.g., the test set accuracy reduces by only 5% (from 0.89 to 0.84) when the number of training samples are reduced by a factor of four. From Fig. E.12, it is seen that this conclusion also holds on a local scale, as the patterns in the error matrices are the same, and only the error rates change as a function of training set size.

In the remainder of this example, the influence of uncertainty associated with the observations, as well as the amount and location of observations,

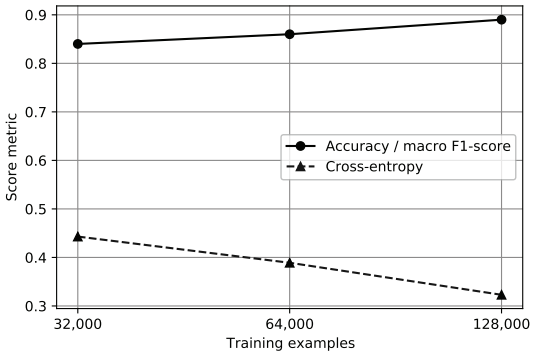


Figure E.11: Influence of training set size in case of perfect information in Example II.

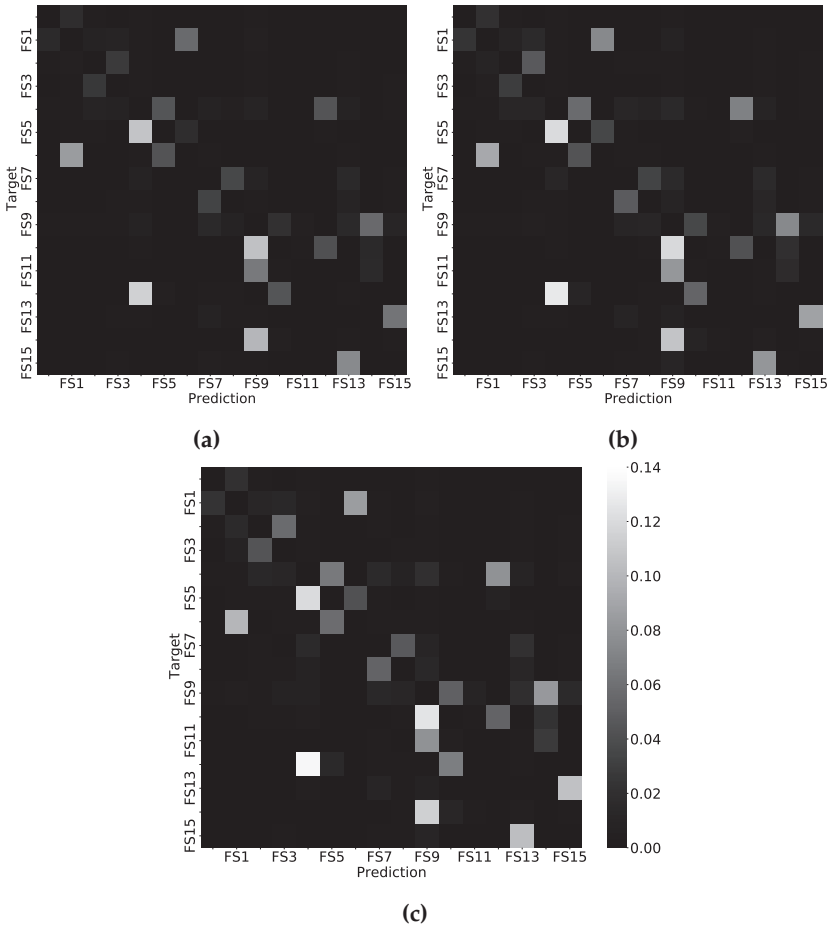


Figure E.12: Scaled test set error matrices in case of perfect information in Example II: (a) using 8000 training samples for each failure scenario; (b) using 4000 training samples of each failure scenario; and (c) using 2000 training samples of each failure scenario. The error matrices are scaled by the number of test samples of each failure scenario.

are studied. To this end, situation (iii) is considered as the baseline, i.e., 2000 training and test realizations of each failure scenario.

Uncertain information In the following, a training and test set of 2000 realizations of each failure scenario, respectively, are considered, corresponding to situation (iii) above. For this case, all observations are now modeled as independent random variables with expected values equal to the realizations

returned by the structural analysis program and additive uncertainty modeled by a zero mean Gaussian random variable, with different variances to reflect the influence of measurement uncertainty on the classifier performance. In this regard, the random variables representing measurement uncertainties associated with load measurements are assigned variances equal to 5%, 10% and 15%, respectively, of the variance of the realizations of the loads acting on the structure. Similarly, the random variables representing the uncertainties associated with observations of the displacements are assigned variances corresponding to 1% and 5%, respectively, of the variance of the realizations of displacements as returned by the structural analysis program. This gives a total of 6 combinations of uncertainties associated with the observations for which the analysis summaries are provided in Tab. E.6.

Table E.6: Performance of the classifier in Example II when uncertainties associated with observations of loads and displacements are accounted for.

Added noise to		Performance metric		
Loads	Displacements	Accuracy	Macro F1-score	Cross-entropy
0.05	0.01	0.71	0.71	0.80
0.05	0.05	0.66	0.66	0.93
0.10	0.01	0.69	0.69	0.87
0.10	0.05	0.64	0.64	0.98
0.15	0.01	0.67	0.67	0.89
0.15	0.05	0.63	0.63	1.01

Table E.6 shows that the test set performance reduces steadily with increasing uncertainty associated with observations of loads and displacements; the accuracy and the macro F1-score reduce and the cross-entropy (loss) increases. Moreover, it is observed that the classifier seems to be more sensitive to uncertainties associated with observations of displacements than uncertainties associated with observations of loads. This is seen from e.g., the reduction in accuracy from 0.71 to 0.66 (5%) when changing the variance of the random variables modeling the uncertainties associated with observations of displacements from 1% to 5%, while keeping the variance of the random variables modeling the uncertainties associated with observations of loads at 5%. The accuracy only reduces from 0.71 to 0.69 (2%) when changing the variance of the random variables modeling the uncertainties associated with observations of loads from 5% to 10%, while keeping the variance of the random variables modeling the uncertainties associated with observations of displacements at 1%. Within the range of uncertainties considered in Tab. E.6, it is again concluded that the classifier globally performs rather robust, as e.g., the test set accuracy reduces by only 8% (from 0.71 to 0.63) when the variance of the random variables modeling the uncertainties in ob-

servations of loads and displacements are increased from 5% and 1% to 15% and 5%, respectively. By considering Fig. E.13, it is seen that this conclusion also holds on a local scale, as the patterns in the error matrices are very similar, but, of cause, the rate of erroneous classifications increases with the level of uncertainties associated with observations.

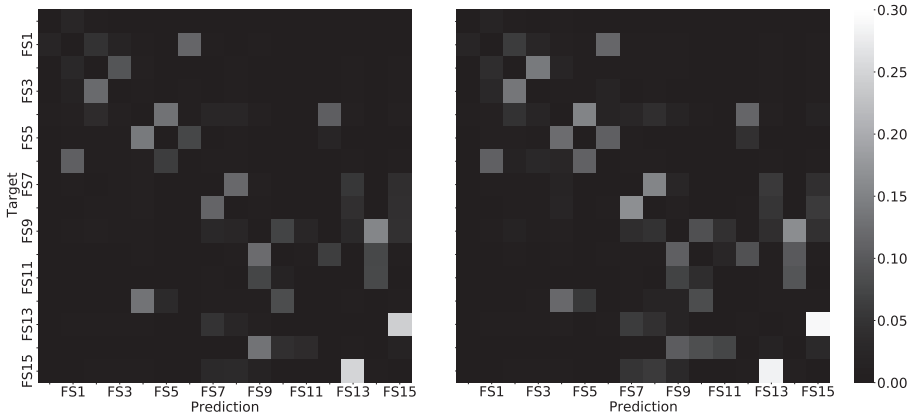


Figure E.13: Scaled test set error matrices in case of uncertain information in Example II: (a) 5% of load variances and 1% of displacement variances; and (b) 15% of load variances and 5% of displacement variances. The error matrices are scaled by the number of test set examples of each failure scenario.

Value of information In order to study the significance of the number of observations and their location on the structure, the following situations are studied: (i) only the horizontal displacement of all beam elements, measured on the right hand side of the structure, are used as input to the classifier; (ii) same as situation (i), but including the vertical displacement at mid-span of the beam elements of the upper story; (iii) same as situation (ii), but including the vertical displacement at mid-span of the beam elements of the middle story; (iiii) same as situation (iii), but including the vertical displacement at mid-span of the beam elements of the lower story, i.e., this situation considers observations of all displacements as input to the classifier. Note that in this assessment, the realizations of displacements are again taken directly from the structural analysis program and are thus not associated with any uncertainty.

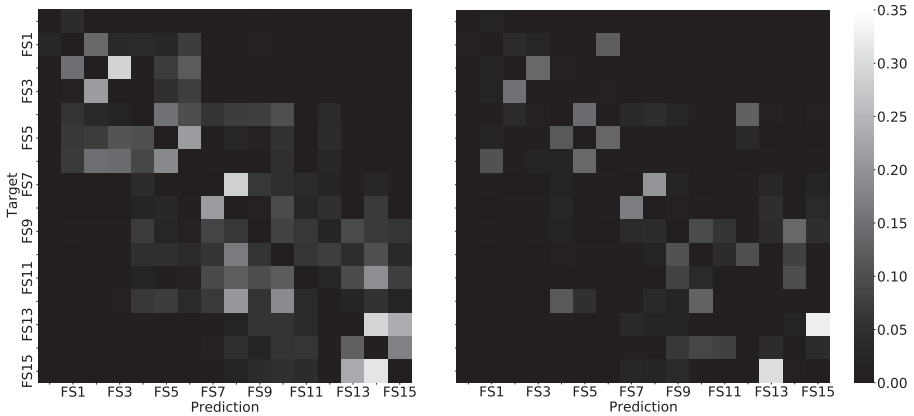
Table E.7 shows the test set performance as a function of the available information in terms of displacement observations. It appears that the performance of the classifier increases steadily with increasing information on the

structural displacements, thus the accuracy and the macro F1-score increase and the cross-entropy (loss) decreases. Moreover, by considering Fig. E.14, it appears that the patterns in the error matrices are similar, and as more information is considered, the confusion around the main diagonal diminishes, i.e., the main diagonal in Fig. E.14a is wider than in Fig. E.14b.

Table E.7: Performance of the classifier in Example II, when different situations regarding available information are considered.

Input information	Performance metric		
Displacements *	Accuracy	Macro F1-score	Cross-entropy
H	0.34	0.33	1.78
H, V-3	0.42	0.41	1.59
H, V-3, V-2	0.50	0.50	1.34
H, V-3, V-2, V-1	0.63	0.63	1.02

* H (horizontal), V (vertical) with index referring to story.



(a) **(b)**

Figure E.14: Scaled test set error matrices in Example II, when different situations regarding available information are considered: (a) Horizontal displacements only; (b) All horizontal and vertical displacements. The error matrices are scaled by the number of test set examples of each failure scenario.

The value of the additional information between the four situations may be quantified in different ways. Ideally, a full structural analysis assessment should be performed to assess the actual cost related with the individual failure scenarios, and the value of information (VoI) should be quantified as the cost reduction resulting from a more optimal choice of structural design – facilitated by the knowledge gained by the observations – in a pre-posterior

decision analysis context, see e.g., [38]. This elaboration will be considered in future research. Another common way of quantifying the VoI in data science is through a model fitness metric, like the ones reflected in Tab. E.7. Thus, using e.g., the cross-entropy to quantify the VoI, it appears that the value of considering all displacement observations (situation (iiii)), as opposed to only horizontal displacement observations (situation (i)), is 0.76. Again, whereas this indicates the effect of improvements in knowledge with respect to damage identification, it does not provide much information with respect to whether such an effect is actually worthwhile.

5 CONCLUSIONS AND DISCUSSIONS

In the present contribution, a novel approach for modeling and analysis of systems based on modern techniques of big data analysis is proposed. Due to the generic character of the suggested approach, it is in principle applicable for any type of real-life system for which a probabilistic model may be established to represent the relationship between realizations of system characteristics, system performances, and observations of these.

The approach utilizes Monte Carlo simulation as a means to generate large numbers of realizations of system states of interest together with corresponding realizations of observable system performances. Classification techniques from big data analysis are then applied to identify probabilistic relationships between specific observations of system performances and system states of interest. These probabilistic relationships may finally be utilized for the management of a real-life system supported by observations of its performances under classified conditions. The proposed approach is illustrated on two principle examples addressing damage identification in idealized structural systems subject to extreme loading. These examples illustrate the application of the suggested approach and its robustness with respect to the complexity of the considered system as well as to the uncertainty associated with observations of system performances.

From the examples, it is seen that the suggested approach with a high level of precision (small type I and type II errors) identifies the correct states of damages. This is indeed very promising since, besides being fully information-consistent, the suggested approach is very efficient in terms of computational efforts. The ability of establishing the damage state assignments in terms of type I and type II errors is a strong feature of the proposed approach, as it facilitates for direct inclusion of the modeling into the framework of risk-informed decision-making.

From the 3-story, 3-bay frame structure example, it is seen that the proposed scheme is indeed robust, even when relatively substantial uncertainties are associated with observations of structural responses, and when the num-

ber of observed structural responses are reduced. Moreover, initial value of information assessments point to the potential benefits of embedding the proposed scheme in a pre-posterior decision analysis to optimize strategies for collecting observations of system responses.

It should be highlighted that an instance-based classification of system performances has been applied in the present study, i.e., classifications based on only one realization of system performances. However, a straightforward extension of the proposed scheme allows for a classification of system states based on any set of functions of system performances; typically referred to as features in structural health monitoring.

In the examples, it is assumed that all possible states of the systems are considered, i.e., the system representations are exhaustive with respect to possible states. This assumption, which is common in the research literature on systems and damage identification, could be argued to be overly idealized, i.e., neglecting the possibility of system states that due to considerations of practical character, or due to model uncertainties, have not been included in the database of system responses. When addressing systems and system states for which a large number of observations from reality are available, it may be assumed that the effect of this is negligible. However, for more unusual systems and system states, which only occur rarely, observations of relevance from reality may be very sparse or not available. To circumvent this potential problem is in principle straight forward; introduce an additional system state that represents any system state not explicitly accounted for. Questions relating to the assignment of probabilities of such events however remain to be adequately considered and answered. The assignment of model uncertainties in probabilistic models concerning rare systems performances is still an important research area.

ACKNOWLEDGMENT

The authors gratefully acknowledge the funding received from Centre for Oil and Gas – DTU / Danish Hydrocarbon Research and Technology Centre (DHRTC).

A ASSESSING CLASSIFIER PERFORMANCE

The performance of a classifier is commonly assessed using a so-called confusion matrix, see Fig. E.15, where the rows represents the true target labels and the columns represent the predictions. Thus, along the rows, the diagonal terms reflect the so-called true positives (TP) for a class, while the off-diagonal elements reflect the so-called false negatives (FN) or type II er-

rors; and along the columns, off-diagonal elements reflect the so-called false positives (FP) or type I errors [37]. That is, an ideal confusion matrix has high non-zero entries along its main diagonal, from left to right, as in Fig. E.15.

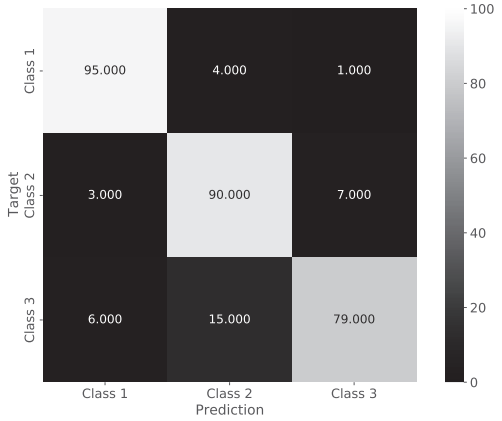


Figure E.15: Principle confusion matrix for a three-class classification problem considering 100 realizations of each class.

A common metric used in relation to the confusion matrix is the accuracy, i.e., the complement of the misclassification error, which is defined as the sum of the diagonal terms to the total number of realizations N used to build the confusion matrix. Furthermore, for all classes, the precision, i.e., TP/(TP+FP), and recall, i.e., TP/(TP+FN), may be calculated and jointly summarized by their harmonic mean, called the F1-score, to reflect the trade-off between FP and FN by giving higher weight to low values [37], i.e.,

$$\mathrm{F1} = 2\,\frac{\mathrm{precision}\cdot\mathrm{recall}}{\mathrm{precision}+\mathrm{recall}}. \tag{E.5}$$

Thus, the F1-score favors classifiers that have similar precision and recall. If this is not a desirable feature, and we want to e.g., give higher weight to FP than to FN, the so-called Fβ-score may be used instead of the F1-score, see e.g., [40]. The per-class F1-score may now be averaged to produces a so-called macro F1-score [20], i.e.,

$$\text{macro F1} = \frac{1}{C}\sum_{c=1}^{C}\mathrm{F1}(c), \tag{E.6}$$

where C is the number of classes in the classification problem.

Similarly, the general fitness of a classifier may be summarized by the cross-entropy [19, 37], or log-loss, which is also a commonly used loss function when fitting gradient boosting classifiers in classification problems. The

cross-entropy for a multi-class problem is defines as

$$H(p,q) = -\sum_{n=1}^{N} p(\boldsymbol{x}[n]) \log q(\boldsymbol{x}[n]), \tag{E.7}$$

which reflects the closeness between the true class-probabilities $p(\boldsymbol{x}[n])$ and the predicted class-probabilities $q(\boldsymbol{x}[n])$. Thus, a classifier that accurately predicts the class-labels receives a value close to 0.

B NUMERICAL MODELING

B.1 RELIABILITY MEASURES AND CALIBRATION

The probability of failure P_{F_j} associated with the failure event F_j is expressed in terms of the probability integral

$$P_{F_j} = \int_{g_j(\boldsymbol{x})} p(\boldsymbol{x}) d\boldsymbol{x}, \quad j = 1, \ldots, n_r, \tag{E.8}$$

where $p(\boldsymbol{x})$ is a multidimensional probability density function defining the random input vector $\boldsymbol{X}$, n_r corresponds to the number of control points considered, and the failure domain g_j, corresponding to the failure event F_j, is defined as

$$g_j = 1 - D_j \leq 0, \quad j = 1, \ldots, n_r. \tag{E.9}$$

Failure is defined as the event where the internal moment $\widehat{M}_j$ exceeds the capacity M_j at any control point, which are located in the nodes at both ends of each element of the structural model, in which case a plastic hinge will appear. Thus, a failure event F_j associated with one hinge location is given by

$$F_j = D_j > 1, \quad j = 1, \ldots, n_r, \tag{E.10}$$

where the normalized demand D_j is defined as

$$D_j = \frac{\widehat{M}_j}{M_j}, \quad j = 1, \ldots, n_r. \tag{E.11}$$

In this regard, the structure is calibrated such that the reliability of the failure events satisfies the following conditions

$$P_{F_j} \leq P^*_{F_j}, \quad j = 1, \ldots, n_r, \tag{E.12}$$

according to a linear structural analysis, where $P^*_{F_j}$ corresponds to the target failure probability of the control point j. The calibration is performed using

the moment of inertial of the structural elements as design variables. An optimization problem is defined aiming to reduce the total cost of the structure, which is assumed to be proportional to the sum of the design variables. To solve the optimization problem, the approach proposed in [41] is adopted, where subset simulation [31] is used as reliability technique to solve the reliability problem.

B.2 NON-LINEAR STRUCTURAL ANALYSIS

In the following, the approach followed for non-linear structural analysis in the examples is outlined. The purpose of this outline and the scope of the non-linear analysis as such is solely to facilitate tractability of the results presented in the examples. Surely more refined and accurate non-linear structural analysis approaches are available and may be selected as found appropriate. The particular choice of non-linear structural analysis has no importance as such in the context of the present paper. An incremental non-linear structural analysis is performed based on a finite element beam representation of structural elements. The loads acting on the structure are increased gradually by a load factor $r_1 \in [0,1]$ and $r_2 \in [0,1]$, respectively, where the former is a multiplier for the horizontal loads, and the latter is a multiplier for the vertical loads. In this regard, r_2 is first gradually increased to 1, while keeping $r_1 = 0$, and then r_1 is increased gradually, while keeping $r_2 = 1$.

This loading pattern allows for solving successive states of equilibrium. For each of these states, a linear analysis is performed and the normalized demand for each node is verified. In case it is exceeded for one or more node(s), taking advantage of the linear analysis properties, the value r_1^* or r_2^* of the load factors that achieves $D_{j^*} = 1$ is identified, where j^* corresponds to the control point with the larger value of the normalized demand. Then, a plastic hinge is imposed at node j^*, allowing to continue increasing the horizontal or vertical load from the value r_1^* or r_2^*, previously identified, and a new state of equilibrium is sought. In this manner, the procedure continues until r_1 and r_2 are equal to 1, unless a global or local mechanism is formed in which case the procedure is terminated.

REFERENCES

[1] L. Nielsen, S. T. Glavind, J. Qin, and M. H. Faber, "Faith and fakes–dealing with critical information in decision analysis," *Civ Eng Environ Syst*, vol. 36, no. 1, pp. 32–54, 2019.

[2] M. H. Faber and M. A. Maes, "Epistemic uncertainties and system choice in decision making," in *Ninth International Conference on Structural Safety and Reliability, ICOSSAR*, 2005, pp. 3519–3526.

[3] D. G. Vlachos, "A review of multiscale analysis: examples from systems biology, materials engineering, and other fluid–surface interacting systems," *Advances in Chemical Engineering*, vol. 30, pp. 1–61, 2005.

[4] G. Stefanou, "The stochastic finite element method: Past, present and future," *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 9-12, pp. 1031–1051, 2009.

[5] Joint Committee on Structural Safety (JCSS), *Risk assessment in engineering: principles, system representation & risk criteria*. Ed. M. H. Faber, 2008.

[6] J. Davidson and J. V. Ringwood, "Mathematical modelling of mooring systems for wave energy converters - a review," *Energies*, vol. 10, no. 5, p. 666, 2017.

[7] O. Mahian, L. Kolsi, M. Amani, P. Estellé, G. Ahmadi, C. Kleinstreuer, J. S. Marshall, R. A. Taylor, E. Abu-Nada, S. Rashidi, H. Niazmand, S. Wongwises, T. Hayat, A. Kasaeian, and I. Pop, "Recent advances in modeling and simulation of nanofluid flows—Part II: Applications," *Physics Reports*, vol. 791, pp. 1–59, 2019.

[8] Y. J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.

[9] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, and D. J. Inman, "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks," *Journal of Sound and Vibration*, vol. 388, pp. 154–170, 2017.

[10] G. Gui, H. Pan, Z. Lin, Y. Li, and Z. Yuan, "Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection," *KSCE Journal of Civil Engineering*, vol. 21, no. 2, pp. 523–534, 2017.

[11] S. Mangalathu and J. S. Jeon, "Machine learning-based failure mode recognition of circular reinforced concrete bridge columns: Comparative study," *Journal of Structural Engineering*, vol. 145, no. 10, p. 04019104, 2019.

[12] G. Mariniello, T. Pastore, C. Menna, P. Festa, and D. Asprone, "Structural damage detection and localization using decision tree ensemble and vibration data," *Computer-aided Civil and Infrastructure Engineering,*, pp. 1–21, 2020.

[13] J. L. Beck, S. Au, and M. W. Vanik, "Monitoring structural health using a probabilistic measure," *Computer-Aided Civil and Infrastructure Engineering*, vol. 16, pp. 1–11, 2002.

[14] W. Zheng and Y. Yu, "Bayesian probabilistic framework for damage identification of steel truss bridges under joint uncertainties," *Adv Civ Eng*, vol. 2013, p. 307171, 2013.

[15] S. T. Glavind, J. G. Sepulveda, J. Qin, and M. H. Faber, "Systems modeling using big data analysis techniques and evidence," in *4th International Conference on System Reliability and Safety (ICSRS)*. IEEE, 2019, pp. 51–59.

[16] B. Kurian and R. Liyanapathirana, "Machine learning techniques for structural health monitoring," in *Lecture Notes in Mechanical Engineering*, M. A. Wahab, Ed. Pleiades Publishing, 2020, pp. 3–24.

[17] J. G. Sepúlveda and M. H. Faber, "Benchmark of emerging structural reliability methods," in *13th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP)*, 2019, pp. 1–8.

[18] L. Sun, Z. Shang, Y. Xia, S. Bhowmick, and S. Nagarajaiah, "Review of bridge structural health monitoring aided by big data and artificial intelligence: From condition assessment to damage detection," *Journal of Structural Engineering*, vol. 146, no. 5, p. 04020073, 2020.

[19] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*. Springer New York, 2009.

[20] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[21] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[22] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning : with applications in R*. Springer, 2013.

[23] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794. [Online]. Available: http://dx.doi.org/10.1145/2939672.2939785

[24] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of bayesian optimization," in *Proceedings of the IEEE*, vol. 104, no. 1, 2016, pp. 148–175.

[25] E. Brochu, V. M. Cora, and N. de Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," 2010, (accessed on 10 August 2020). [Online]. Available: https://arxiv.org/abs/1012.2599

[26] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[27] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in Neural Information Processing Systems*, 2011.

[28] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proceedings of the 30th International Conference on Machine Learning*. International Machine Learning Society (IMLS), 2013, pp. 115–123.

[29] W. M. Czarnecki, S. Podlewska, and A. J. Bojarski, "Robust optimization of svm hyperparameters in the classification of bioactive compounds," *Journal of Cheminformatics*, vol. 7, no. 1, p. 38, 2015.

[30] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT press, 2006.

[31] S.-K. Au and J. Beck, "Estimation of small failure probability in high dimensions by subset simulation," *Probabilistic Engineering Mechanics*, vol. 16, no. 4, pp. 263–277, 2001.

[32] S. K. Au and Y. Wang, *Engineering Risk Assessment with Subset Simulation*. Wiley, 2014, vol. 9781118398043.

[33] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*. CRC press, 2013.

[34] P. Thoft-Christensen and Y. Murotsu, *Application of structural systems reliability theory*. Springer-Verlag, 1986.

[35] D.-S. Kim, S.-Y. Ok, J. Song, and H.-M. Koh, "System reliability analysis using dominant failure modes identified by selective searching technique," *Reliability Engineering & System Safety*, vol. 119, pp. 316–331, 2013.

[36] The GPyOpt authors, "GPyOpt: A bayesian optimization framework in python," http://github.com/SheffieldML/GPyOpt, 2016.

[37] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[38] H. Raiffa and R. Schlaifer, *Applied statistical decision theory*. MIT Press, 1961.

[39] J. von Neumann and O. Morgenstern, *Theory of games and economic behavior*. Princeton University Press, 1953.

[40] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from imbalanced data sets*. Springer, 2018.

[41] H. Jensen and J. Sepúlveda, "Structural optimization of uncertain dynamical systems considering mixed-design variables," *Probabilistic Engineering Mechanics*, vol. 26, no. 2, pp. 269–280, 2011.

PAPER F

ON SYSTEMS MODELING AND CONTEXT-SPECIFIC MODEL SELECTION IN OFFSHORE ENGINEERING

Sebastian T. Glavind, Henning Brüske, Erik D. Christensen, and Michael H. Faber

The paper has been submitted to
Computer-Aided Civil and Infrastructure Engineering.

In peer-review
The layout has been revised.

ABSTRACT

Motivated by an increasing industrial and regulatory focus on integrity management of ageing offshore oil and gas production plants, we take up the challenge of developing an improved basis for the probabilistic representation of systems and apply this to model the offshore load environment. To this end, we propose and outline a framework that takes basis in Bayesian networks and Gaussian processes. This framework provides for a consistent treatment and representation of all prevailing epistemic and aleatory uncertainties, and it facilitates for model updating at the same rate as new information is made available. The main novelty of our contribution is that we account explicitly for possible competing system representations and integrate the systems representation problem – the modeling – into the context of the decision analysis problem the models aim to serve. Our proposed approach is first illustrated by a principle example showing all the aspects of our approach. Thereafter, we introduce a real case application, where we derive a Bayesian network for the probabilistic representation of storm events. Finally, we embed the probabilistic storm representation into a decision problem concerning the ranking of possible alternatives for the evacuation of a set of offshore structures, provided information on an emerging storm event.

Keywords*: systems modeling, Bayesian network, Gaussian processes, multiple data sources, model selection and decision optimization.*

1 INTRODUCTION

Over the last decade, a variety of different probabilistic modeling frameworks have been proposed to facilitate for an information consistent joint representation of the metocean variables, which are commonly used to represent the load environment for offshore energy production facilities. These modeling frameworks aim to adequately account for the available statistical information in the probabilistic representation of the metocean variables marginally as well as jointly, especially in domains of realizations where one or more of the variables take extreme values. Such dependencies are often not accounted for consistently in normal engineering design practice, where e.g., the maximum load effects originating from the jointly occurring metocean variables are typically modeled by an aggregation of the maximum load effects originating from the metocean variables individually [1, 2]. This may thus result in design loads of unknown probabilities of occurrences.

Whereas the recent progress on modeling frameworks for the probabilistic representation of the offshore load environment comprise significant contributions to enhanced consistency in the modeling, they still leave potential for important improvements. Indeed, the present state-of-the-art on planning of

statistical experiments and the associated probabilistic modeling of systems do not account for two important aspects, namely (i) the decision context in which the results and the associated probabilistic models are applied, and (ii) the fact that more than one model might be adequate for the probabilistic representation of the system at hand.

In the context of probabilistic systems modeling in civil engineering, only a few contributions related to the consideration of these two aspects are known to the authors. In [3, 4], the problem of optimizing statistical experiments is integrated into the context in which the resulting model is applied, and in [5], the possibility of competing probabilistic models in engineering decision-making is accounted for in the ranking of decision alternatives. In [6] the philosophical and theoretical framework for the representation and treatment of critical information in decision-making is outlined. This framework forms the basis for [7, 8] in which a novel contribution is presented on how to consistently integrate both optimization of collection of information (aspect (i)) and the possibility of competing probabilistic system representations (aspect (ii)) into the decision context, which the probabilistic system representations aim to support.

In the present contribution, we build on [6] and [7, 8] to formulate and demonstrate how Bayesian modeling techniques may be applied to establish probabilistic representations of offshore load environments in the context of risk modeling and integrity management for offshore structures with a joint consideration of both aspect (i), i.e., decision context, and (ii), i.e., model multiplicity. This approach allows for an integration of both classical engineering bottom-up modeling, i.e., models based on phenomenological understanding of the problem domain, and the emerging techniques of top-down (data-driven) modeling using e.g., big data and deep learning.

To illustrate the novel modeling framework and the associated techniques, we consider a full-scale application example, where offshore environmental load models are established in the decision context of safety management and re-qualification of offshore structures in the Danish part of the North Sea. The example addresses the general aspects of developing a probabilistic environmental load model based on records of past storm events. Considering this load model, it is shown how decisions regarding possible platform evacuation in the face of an approaching storm event may be supported based on the associated risks. The example further points to the potentials of the developed model framework in the context of establishing load models for ultimate and fatigue limit states.

The remainder of the this paper is organized as follows: Sec. 2 reviews the literature on load environment modeling in offshore engineering, which forms the basis for a discussion on system representations using Bayesian networks and Gaussian processes in Secs. 3 and 4, respectively. Section 5 presents how competing system representations may be accounted for in rela-

tion to inference and decision support, and the ideas are applied on a simple, principle example in Sec. 6. Sections 7 and 8 apply our novel methodology to a full scale application, and the paper is summarized and concluded in Sec. 9.

2 LOAD ENVIRONMENT MODELING IN OFFSHORE ENGINEERING

Probabilistic load modeling in relation to safety assessments of offshore structures normally focus on either the fatigue limit state (FLS), the ultimate limit state (ULS), or the accidental limit state. The most important load component in relation to FLS assessments of welded offshore structures is wave loading, but also winds and currents contribute to fatigue damage accumulation.

The statistical properties of ocean waves are most often represented by integral properties of sea states, such as significant wave height and zero-crossing period, with a reference frame of 1 to 3 hours over which the parameters are assumed to be stationary. Within each sea state, the water surface elevation is commonly modeled by a stationary Gaussian process with a site-specific spectral representation, e.g., the JONSWAP spectrum for the North Sea region. Moreover, this representation of the surface elevation may be supplemented with a correction for non-linear effects [7, 9].

A FLS assessment of an offshore structure requires that we specify all relevant environmental conditions that are expected to occur during its period of exposure, i.e., its construction phase (including transport) and its design life [2]. This means that we need the long-term distribution of the sea states variables.

Traditionally, the long-term distribution of sea state variables is established by bottom-up modeling approaches, i.e., by combining constituent-based phenomenological models to form a sea state system representation. If the available information about the joint set of sea state variables is limited to the marginal distributions and the mutual correlations, then the Nataf transform may be used [10, 11]. Provided that sufficient domain specific knowledge is available to directly apply the chain rule of probability theory, a joint model may be established by a sequence of conditional probability distributions, see e.g., [11–13]. This modeling approach is often referred to as conditional modeling. A modern approach based on assumptions regarding the parametric relations in a problem domain is Copula modeling, where Sklar's theorem [14] is employed to represent a multivariate joint distribution in terms of univariate marginal distribution functions and one or more Copulas, which describe the dependency structure between the variables [15, 16].

In top-down modeling, the joint probabilistic model is constructed by ex-

posing a joint set of observations of the variables to modern machine learning techniques. The reader is refereed to e.g., [17–19] for a comprehensive coverage of the various techniques, and applications in offshore engineering may be found in e.g., [20, 21], which illustrate the use of neural networks and Gaussian process regression, respectively, as statistical emulators in relation to record extension of sea state data.

Assessing the structural safety related to extreme load events, i.e., ULS, requires estimation of e.g., the maximum wave height or crest for a given return period, together with the dependence between their extremes and simultaneous realization of the remaining domain variables. In this regard, the domain variables are the variables included in the system representation. That is, rather than defining the dependency relation for the bulk of the data, as captured by the long-term distribution, focus is directed on the modeling of the tail behavior, which enables extrapolation beyond the original sample to quantify extreme quantiles of the distribution.

When dealing with extremes, a first challenge is to answer the fundamental question: What makes a multivariate observation extreme? Thus, it must be framed whether an extreme event is defined by an extreme realization of one variable, or several variables attaining extreme realizations simultaneously. Furthermore, since extreme value models are motivated by asymptotic assumptions, it is generally required to define a threshold in order to assess which realizations should be considered in the extreme value analysis [22, 23].

In case domain specific knowledge provides the necessary means for the construction of a joint extreme value model, the bottom-up techniques listed above may be applied. In ocean engineering, for instance, extreme events are often defined as observations where one variable, usually the significant wave height, is extreme. For this case, a joint model is constructed by first estimating the marginal distribution for this variable, and then the remaining variables are modeled conditional on this variable being extreme. See e.g., [24–26] for applications of bottom-up modeling of extremes in ocean engineering.

When sufficient domain specific knowledge is not available to formulate an extreme value model, it is generally necessary to resort to the approach of modeling conditional extremes [27] or similar, see [23] for a review on extreme value analysis in ocean engineering. Some applications of the conditional approach to metocean data can found in [28–32].

In this study, focus is directed on the modeling of the long-term, multivariate distribution of storm events using Bayesian networks (BNs), also known as directed graphical models, which is a branch of probabilistic graphical models that uses Bayesian inference for probability computations. Reasoning processes can operate on BNs by propagating information in any direction, which enables not only prediction (forward propagation) and abduc-

tion (backward propagation, or reasoning to a problem cause) but also inter-causal reasoning (explaining-away), where the confirmation of one cause increases or decreases belief in others. This last form of reasoning gives BNs a competitive edge compared to other machine learning techniques, like rule-based systems and neural networks, and makes them an important tool for risk assessment and decision analysis [33]. Applications of BNs in relation to offshore asset integrity management appear in e.g., [34–40].

3 SYSTEM REPRESENTATIONS USING BAYESIAN NETWORKS

In this section, the representation of systems is addressed using the framework of Bayesian networks (BNs). Starting point is taken by defining BNs in Sec. 3.1, and we proceed to discuss learning of BNs in Secs. 3.2 and 3.3 for fully observed and partially observed data sets, respectively. The reader is referred to e.g., [8, 41] for a general introduction to inference in BNs.

3.1 INTRODUCTION TO BAYESIAN NETWORKS

BNs are probabilistic graphical models that allow for reasoning and learning in complex, uncertain domains; where reasoning refers to the task of performing probabilistic inference on one or several variable(s) in the problem domain, e.g., querying the (conditional) distribution of a variable, potentially given observations on some other variables in the model; and learning refers to the task of specifying the BN model, i.e., model structure and parameters given a training data set.[1]

The dependencies between the domain variables $\boldsymbol{X}$ are represented as edges in a directed acyclic graph (DAG) $\mathcal{G}$, wherein the variables appear as vertices. A factor $P(X_i|\mathbf{Pa}_i)$ is specified for each variables X_i, which encodes its conditional probability distribution, given the variables that X_i dependents directly on in $\mathcal{G}$, i.e., the so-called parent set $\mathbf{Pa}_i$ of the variable [42]. The BN model then defines a probability distribution over the domain variables as

$$P(\boldsymbol{X}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}) = \prod_i P(X_i|\mathbf{Pa}_i), \tag{F.1}$$

where $\boldsymbol{\Theta}_{\mathcal{G}}$ denotes the set of model parameters needed to represent the factors, see e.g., [8] for further details.

Like most machine learning frameworks, BNs provides a means for modeling an uncertain domain based on noisy observations, but what distinguishes this modeling scheme from most others is the ease with which

[1] In classical statistics this is often referred to as model estimation.

prior knowledge can be included in the modeling on both the structural level (DAG) and parameter level (factors). This makes BNs an ideal tool for combining bottom-up and top-down modeling [43].

3.2 LEARNING FROM COMPLETE DATA SETS

In this study, a Bayesian approach to learning is adopted, which necessitates that prior beliefs are specified with respect to $\{\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}\}$, i.e., the model structure and parameters, as $P(\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}})$. After we observe some data $\mathcal{D}$, our beliefs about $\{\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}\}$ may then be updated to obtain posterior beliefs as

$$\underbrace{P(\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{D})}_{\text{posterior}} \propto \underbrace{P(\mathcal{D}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}})}_{\text{likelihood}} \underbrace{P(\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}})}_{\text{prior}}. \tag{F.2}$$

In this setting, the data set $\mathcal{D} = \{\boldsymbol{x}[n]\}_{n=1}^{N}$ is composed of N i.i.d. realizations. The change in distribution represented by Eq. F.2 reflects the information gain we get by observing some data, and it further illustrates what it means for a machine to "learn from data". Moreover, the posterior distribution in turn becomes the prior distribution to be used with new observations, which makes the updating process inherently sequential and therefore well suited for online learning [41].

In the remainder of this section, we consider learning of system representations for discrete-valued random variables, or accordingly dynamically discretized continuous-valued random variables. In this way, we try to keep the distributional assumptions for the domain variables at a minimum when learning BN models.

PARAMETER LEARNING

In this section, we show how to learn the parameters of a BN from data, when the corresponding DAG $\mathcal{G}$ is given, and the available data set $\mathcal{D}$ consists of complete assignments to all variables. In this setting, with reference to Eq. F.2, parameter learning is usually performed by searching for a set of parameters in $\boldsymbol{\Theta}_{\mathcal{G}}$ that maximizes

$$P(\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{G}, \mathcal{D}) = \frac{P(\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{D})}{P(\mathcal{G})} \propto P(\mathcal{D}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}) P(\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{G}), \tag{F.3}$$

where $P(\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}) = P(\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{G})P(\mathcal{G})$. The likelihood $P(\mathcal{D}|\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}})$ factorizes according to $\mathcal{G}$, and the prior $P(\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{G})$ is decomposed by assuming global and local parameter independence, together with a Dirichlet equivalent uniform prior for the parent configurations $\boldsymbol{u}_i$ in the factors, see e.g., [44].

Based on these assumptions, the posterior of the parameters also decomposes into a product Dirichlet distribution:

$$P(\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{G},\mathcal{D}) = \prod_i \prod_{\boldsymbol{u}_i \in \text{Val}(\mathbf{Pa}_i)} \text{Dir}\left(\left\{\alpha_{x_i^j|\boldsymbol{u}_i} + N[x_i^j,\boldsymbol{u}_i]\right\}_{j=1}^{|X_i|}\right), \tag{F.4}$$

where $\alpha_{x_i}^j$ and $N[x_i^j,\boldsymbol{u}_i]$ are the prior weight and the number of samples in bin j of variable X_i with parent configuration $\boldsymbol{u}_i$ [44, 45].

STRUCTURE LEARNING

Attention is now directed on how to learn the DAG of a BN from complete data. Taking basis in Eq. F.2, structure learning is usually performed by searching for a DAG $\mathcal{G}$ that maximizes

$$P(\mathcal{G}|\mathcal{D}) \propto P(\mathcal{G})P(\mathcal{D}|\mathcal{G}) = P(\mathcal{G})\int P(\mathcal{D}|\mathcal{G},\boldsymbol{\theta}_{\mathcal{G}})P(\boldsymbol{\theta}_{\mathcal{G}}|\mathcal{G})d\boldsymbol{\theta}_{\mathcal{G}}, \tag{F.5}$$

where $P(\mathcal{G},\boldsymbol{\Theta}_{\mathcal{G}}) = P(\boldsymbol{\Theta}_{\mathcal{G}}|\mathcal{G})P(\mathcal{G})$, and the prior over DAG structures $P(\mathcal{G})$ is usually assumed to be uniform. As is apparent from Eq. F.5, it is not possible to perform this computation without also considering the parameters $\boldsymbol{\Theta}_{\mathcal{G}}$ of the BN model. Therefore, to make $P(\mathcal{G}|\mathcal{D})$ independent of any specific choice of $\boldsymbol{\Theta}_{\mathcal{G}}$, $\boldsymbol{\Theta}_{\mathcal{G}}$ needs to be integrated out of the equation [43, 45].

Under the assumptions stated above for parameter learning, $P(\mathcal{D}|\mathcal{G})$ may be estimated in closed form as

$$P(\mathcal{D}|\mathcal{G}) = \prod_i \prod_{\boldsymbol{u}_i \in \text{Val}(\mathbf{Pa}_i)} \frac{\Gamma(\alpha_{X_i|\boldsymbol{u}_i})}{\Gamma(\alpha_{X_i|\boldsymbol{u}_i} + N[\boldsymbol{u}_i])} \prod_{x_i^j \in \text{Val}(X_i)} \frac{\Gamma(\alpha_{x_i^j|\boldsymbol{u}_i} + N[x_i^j,\boldsymbol{u}_i])}{\Gamma(\alpha_{x_i^j|\boldsymbol{u}_i})}, \tag{F.6}$$

where $\Gamma(\cdot)$ is the Gamma function, $N[\boldsymbol{u}_i]$ is the number of samples with configuration $\boldsymbol{u}_i$, and $N[x_i^j,\boldsymbol{u}_i]$ is the number of samples in bin j of variable X_i with parent configuration $\boldsymbol{u}_i$ [44, 46].

Finding a DAG that maximizes Eq. F.6 is generally an intractable problem [47]. One approach to deal with this problem is to resort to a heuristic search strategy to find a high-scoring DAG. In this study, we apply hill-climbing strategies to perform score-based structure learning. A greedy hill-climbing strategy proceeds as follows: In each iteration, we define the neighborhood of the current DAG $\mathcal{G}^{(t)}$ as all DAGs, we can produce from $\mathcal{G}^{(t)}$ by adding an edge, removing an edge, or reversing an edge. In this neighborhood, we pick the DAG that has the highest score and update $\mathcal{G}^{(t+1)}$. This strategy only guaranties to find a local optimum, but we may improve our chances of finding a "good" optimum by including a tabu list of previously visited structures and/or performing random restarts, when a local optimum

is reached [43, 48]. Score-based algorithms, and tabu search in particular, have been shown to perform well in practice in terms of accuracy and speed of network reconstruction for both small and large sample sizes [48].

STRUCTURE LEARNING AND AUTOMATIC DISCRETIZATION

In this section, we consider how to learn the DAG of a BN and, at the same time, the optimal discretization of continuous variables from complete data. In this setting, we assume that the data are generated by first sampling a realization of the discrete-valued variables form their joint distribution, and then drawing continuous values within the discrete-numbered intervals independently [49].

By including the discretization policy $\Lambda_{\mathcal{G}}$ of the observed continuous-valued data $\mathcal{D}^c$ in the learning problem, we need to specify beliefs on the triple $\{\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}, \Lambda_{\mathcal{G}}\}$. Analogously to Eq. F.2, the joint posterior distribution for this problem takes the following form

$$P(\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}, \Lambda_{\mathcal{G}} | \mathcal{D}^c) \propto P(\mathcal{D}^c | \mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}, \Lambda_{\mathcal{G}}) P(\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}, \Lambda_{\mathcal{G}}), \tag{F.7}$$

where $\Lambda_{\mathcal{G}}$ specifies a set of interval boundary points for each variable. Furthermore, as implicitly implied by the generative process for $\mathcal{D}^c$, it is assumed that given $\{\mathcal{D}, \Lambda_{\mathcal{G}}\}$, $\mathcal{D}^c$ is conditionally independent of $\{\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}\}$, whereby Eq. F.7 may be rewritten as

$$P(\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}, \Lambda_{\mathcal{G}} | \mathcal{D}^c) \propto \underbrace{P(\mathcal{D}^c | \mathcal{D}, \Lambda_{\mathcal{G}})}_{\text{likelihood (continuous)}} \underbrace{P(\mathcal{D} | \mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}, \Lambda_{\mathcal{G}})}_{\text{likelihood (discrete)}} \underbrace{P(\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}, \Lambda_{\mathcal{G}})}_{\text{prior}}. \tag{F.8}$$

Under the assumptions stated above for structure learning, together with the additional assumption of an uniform prior over the discretization policies, the product of the two last terms in Eq. F.8 (the discrete part) corresponds to the evaluation of Eq. F.6. Furthermore, the following formulations of the continuous likelihood are considered in the literature [49, 50]:

$$P(\mathcal{D}^c | \mathcal{D}, \Lambda_{\mathcal{G}}) = \prod_i \prod_{x_i^j \in Val(X_i)} \left(\frac{1}{\overline{\lambda}_i^j - \underline{\lambda}_i^j} \right)^{N[x_i^j]} \tag{F.9a}$$

$$P(\mathcal{D}^c | \mathcal{D}, \Lambda_{\mathcal{G}}) = \prod_i \prod_{x_i^j \in Val(X_i)} \left(\frac{1}{N[x_i^j]} \right)^{N[x_i^j]}, \tag{F.9b}$$

where $N[x_i^j]$ is the number of samples in bin j of variable X_i, regardless of the parent configuration, and $\underline{\lambda}_i^j$ and $\overline{\lambda}_i^j$ are the lower and upper boundary

point in bin j of variable X_i. Moreover, only the $N-1$ midpoints of each data vector $\mathcal{D}_i^c$ are considered as candidate boundary points for X_i.

Based on these assumptions, Eq. F.8 aims to establish a trade-off between model simplicity and representativeness. On one hand, the formulations of $P(\mathcal{D}^c|\mathcal{D},\Lambda_\mathcal{G})$ in Eq. F.9 rewards model complexity and prediction accuracy with respect to the continuous variable, and thus it increases with an increasing number of intervals. On the other hand, the discrete part of Eq. F.8, penalizes model complexity, whereby it balances the resolution of the discretization by decreasing as the number of intervals increase [49, 50].

The learning problem is solved by successively applying the following two steps until convergence: (i) discretize the data based on the current DAG, and (ii) learn a new DAG based on this discretization of the data, see [7, 8, 50, 51] for further details.

3.3 LEARNING FROM INCOMPLETE DATA SETS

In this section, learning of BN models is considered, when the set of available data is incomplete. That is, we have a data set $\mathcal{D} = \{\mathcal{D}_{obs}, \mathcal{D}_{hid}\}$, where $\mathcal{D}_{obs}$ denotes the observed data and $\mathcal{D}_{hid}$ denotes the hidden data. Now, assuming that the incomplete data set has been generated from a complete data set by a process that hides some of the data. One of three assumptions for the missing-data mechanism is typically considered: (i) data missing completely at random (MCAR), where the mechanism is assumed to be independent of the data; (ii) data missing at random (MAR), where it is assumed that the mechanism do not depend on the hidden data; and (iii) data missing not at random (MNAR), where the mechanism depends on both the observed and the hidden data [52].

Most applications assume the data to be MAR, and in the remainder of this section the same assumption is made. In this case, the data likelihood in Eq. F.2 may be evaluated as

$$P(\mathcal{D}|\mathcal{G},\boldsymbol{\Theta}_\mathcal{G}) = \sum_{\mathcal{D}_{hid}} P(\mathcal{D}_{obs},\mathcal{D}_{hid}|\mathcal{G},\boldsymbol{\Theta}_\mathcal{G}). \tag{F.10}$$

In order to evaluate this likelihood, inference for the hidden variables of each instance n must be undertaken, which means that the property of parameter independence is lost, and we thereby also lose the decomposability of the likelihood function [44].

PARAMETER LEARNING

The basis for parameter learning is again Eq. F.3 with a product Dirichlet equivalent uniform prior that satisfies both global and local parameter independence, but as the posterior distribution is a product of likelihood and prior, and the likelihood is not decomposable, no closed form solution exists.

One way of addressing this problem is to learn the parameter setting that maximizes the posterior utilizing, for instance, a generic gradient based optimization algorithm or the expectation maximization (EM) algorithm. Another way of addressing the problem is to use a sampling based method, like Gibbs sampling, to approximate the posterior distribution [44, 53]. A review on common algorithms used for parameter learning in BNs may be found in e.g., [54, 55].

STRUCTURE LEARNING AND AUTOMATIC DISCRETIZATION

Learning the DAG structure of a BN (in addition to the parameters) from an incomplete data set is challenging from both a methodical and a computational point of view. The score metrics defined in the previous sections, i.e., Eqs. F.5 and F.8, are functions of the sufficient statistics $N[\cdot]$ through the definition of the (discrete) data likelihood in Eq. F.6, and thus these are not defined for incomplete data sets. To circumvent this problem, the definition of the (missing) data likelihood in Eq. F.10 may be adopted, and thereby the data likelihood may be established on the basis of the expected sufficient statistics.

As for the case of parameter learning given a graph structure, this may be accomplished by utilizing a deterministic optimization algorithm, such as EM, or a stochastic procedure, like Gibbs sampling. The former approach is now termed structural EM [56, 57], and it proceeds by embedding the structural search inside the EM procedure. The latter approach is usually termed data augmentation [58], and it utilizes a sampling approach to produce several completions of the training data set, which may then be used for structure learning in a complete data setting.

By use of one of the approaches outlined in the foregoing, the (discrete) data likelihood may now be evaluated and learning of the triple $\{\mathcal{G}, \boldsymbol{\Theta}_{\mathcal{G}}, \Lambda_{\mathcal{G}}\}$ can again be performed in accordance with Eq. F.8.

4 DISCREPANCY MODELING USING GAUSSIAN PROCESS REGRESSION

In this section, we discuss how to correct simulator outputs for model bias using measurement results and discrepancy modeling. This is a natural preprocessing step before formulating the system representation(s) in Sec. 3, when measurements exists that correspond to some of the simulator outputs. First, Sec. 4.1 introduces the general concept of discrepancy modeling, and then Secs. 4.2 and 4.3 explain how Gaussian processes can be used for discrepancy modeling in a single-output setting and multiple-output setting, respectively.

4.1 INTRODUCTION TO DISCREPANCY MODELING

Deterministic computer-based simulators are used extensively to predict the response of complex systems, such as weather systems [59], structural systems [60], and systems representing climate changes [61, 62]. For such cases, the relationship between the true response $y(\boldsymbol{x})$ and the simulator output $g(\boldsymbol{x})$, evaluated at input $\boldsymbol{x}$, may be formulated as

$$y(\boldsymbol{x}) = g(\boldsymbol{x}) + f(\boldsymbol{x}), \tag{F.11}$$

where $f(\boldsymbol{x})$ is a systematic, additive discrepancy function. Moreover, if only a set of noisy measurements of the true system responses are available, the relationship may be formulated as

$$z(\boldsymbol{x}) = g(\boldsymbol{x}) + f(\boldsymbol{x}) + \varepsilon, \tag{F.12}$$

where ε is the noise related to our response measurements $z(\boldsymbol{x})$. Now, defining $r(\boldsymbol{x}) = z(\boldsymbol{x}) - g(\boldsymbol{x})$, the following formulation is obtained:

$$r(\boldsymbol{x}) = f(\boldsymbol{x}) + \varepsilon. \tag{F.13}$$

Based on Eq. F.13, we can define a discrepancy function, which enables us to correct the simulator outputs for model bias at unmeasured inputs before proceeding to learn a probabilistic system representation using e.g., the BN approach outlined in the foregoing. In this regard, any paradigm from the literature on regression theory may be applied to define the discrepancy function. In the following sections, it is shown how this can be accomplished by use of single- and multi-output Gaussian process regression.

4.2 SINGLE-OUTPUT GAUSSIAN PROCESSES

A Gaussian process (GP) is a collection of random variables indexed by e.g., time or space, such that any finite subset of the variables have a joint Gaussian distribution.

Taking basis in Eq. F.13, the discrepancies between simulator outputs and measurements at known inputs $\boldsymbol{x}$ are considered, and it is assumed that ε is an additive white noise process given by $\varepsilon = \mathcal{N}(0, \sigma^2)$. Moreover, f is assumed to be a non-linear, non-parametric function with a GP prior, i.e.,

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')), \tag{F.14}$$

where $m(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})]$ is the expected value function, and $k(\boldsymbol{x}, \boldsymbol{x}') = \text{cov}[f(\boldsymbol{x}), f(\boldsymbol{x}')]$ is the positive semi-definite covariance, or kernel, function. This definition allows us to evaluate the mean function at an arbitrary input setting and assess how the value of the function at an input point covary

with the value of the function at other points in input space. Therefore, a GP may be interpreted as defining a probability distribution over functions, and inference accordingly takes place directly in the space of functions [63].

Given a data set $\mathcal{D} = \{\hat{\boldsymbol{X}}, \hat{\boldsymbol{r}}\} = \{\boldsymbol{x}[n], r[n]\}_{n=1}^{N}$ of vector-valued input and scalar output, the GP prior is established by evaluating the expected value and covariance function at the data points, which leads to a multivariate Gaussian distribution over the corresponding function values, i.e.,

$$f(\hat{\boldsymbol{X}}) \sim \mathcal{N}(m(\hat{\boldsymbol{X}}), k(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}})). \tag{F.15}$$

Under proper normalization of the data, the expected value of the process can be assumed to be zero without loss of generality. The covariance function should capture basic aspects of the process, such as stationarity, ergodicity, isotropy, smoothness, and periodicity. When data points that are close in input space tend to produce similar outputs, a common choice of covariance function is the squared exponential kernel or the Matérn kernel [63].

One attractive feature of the GP formulation as described in this section is that exact inference in tractable. Consider the prediction f_* for a new input $\boldsymbol{x}_*$. Under a Gaussian noise assumption in Eq. F.13, i.e., $r \sim \mathcal{N}(f(\boldsymbol{x}), \sigma^2)$, the joint distribution of the observed discrepancies $\hat{\boldsymbol{r}}$ and the discrepancy function at the test location under the prior may be written as

$$\begin{bmatrix} \hat{\boldsymbol{r}} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} k(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) + \sigma^2 \mathbf{I} & k(\hat{\boldsymbol{X}}, \boldsymbol{x}_*) \\ k(\boldsymbol{x}_*, \hat{\boldsymbol{X}}) & k(\boldsymbol{x}_*, \boldsymbol{x}_*) \end{bmatrix}\right) \tag{F.16}$$

where $\mathbf{I}$ is the identity matrix, and it is assumed that the data are properly normalized, so that $m(\boldsymbol{x}) = 0$.

By direct application of the standard rules for conditioning of Gaussian distributed random variables, this joint prior distribution can be restricted to contain only those functions that agree with the observations, i.e., by conditioning the prior on the observed data points. The predictive distribution for $f(\boldsymbol{x}_*)$ may then be written as

$$p(f(\boldsymbol{x}_*)|\mathcal{D}, \boldsymbol{x}_*, \boldsymbol{\Theta}) = \mathcal{N}(m_*(\boldsymbol{x}_*), k_*(\boldsymbol{x}_*, \boldsymbol{x}_*)), \tag{F.17}$$

where $\boldsymbol{\Theta}$ denotes the set of model parameters, and m_* and k_* are defined as

$$\begin{aligned} m_*(\boldsymbol{x}_*) &= \boldsymbol{k}_{\boldsymbol{x}_*}(k(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) + \sigma^2 \mathbf{I})^{-1} \hat{\boldsymbol{r}} \\ k_*(\boldsymbol{x}_*, \boldsymbol{x}_*) &= k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_{\boldsymbol{x}_*}(k(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) + \sigma^2 \hat{\boldsymbol{I}})^{-1} \boldsymbol{k}_{\boldsymbol{x}_*}^T, \end{aligned}$$

with $\boldsymbol{k}_{\boldsymbol{x}_*}$ as a shorthand notation for $k(\boldsymbol{x}_*, \hat{\boldsymbol{X}})$. Note that if the corresponding noisy prediction r_* is desired, all what is needed is to add σ^2 to the predictive variance expression above. See [17, 63] for further details.

4.3 MULTI-OUTPUT GAUSSIAN PROCESSES

In the following, the framework presented in the foregoing section is extended to cover multi-output processes, thus the available data set for this case is $\mathcal{D} = \{\hat{\boldsymbol{X}}, \hat{\boldsymbol{R}}\} = \{\boldsymbol{x}[n], \boldsymbol{r}[n]\}_{n=1}^{N}$, where both the inputs and outputs are vector-valued. Moreover, in the further treatment, it is assumed that all inputs are applied in the regression for all outputs.

In multi-output learning the output space is a vector space, thus leading to a vector-valued estimator $\boldsymbol{f} = \{f_d\}_{d=1}^{D}$, which is assumed to follow a GP, i.e.,

$$\boldsymbol{f} \sim \mathcal{GP}(\boldsymbol{m}, \boldsymbol{K}), \tag{F.18}$$

where $\boldsymbol{m} = \{m_d(\boldsymbol{x})\}_{d=1}^{D}$, i.e., the expected value functions of the outputs, and $\boldsymbol{K} = (\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}'))_{d,d'}$ is a positive semi-definite, matrix-valued function, such that the entries correspond to the covariances between the outputs $f_d(\boldsymbol{x})$ and $f_{d'}(\boldsymbol{x}')$ [64].

The prior distribution over $\boldsymbol{f}$ takes the following form

$$\boldsymbol{f}(\hat{\boldsymbol{X}}) \sim \mathcal{N}(\boldsymbol{m}(\hat{\boldsymbol{X}}), \boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}})), \tag{F.19}$$

where $\boldsymbol{m}(\hat{\boldsymbol{X}})$ is a vector that concatenates the expected value vectors of the outputs, which under proper normalization of the data can be assumed to be the zero vector without loss of generality, and $\boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}})$ is a block partitioned matrix defined as

$$\boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) = \begin{bmatrix} (\boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}))_{1,1} & \cdots & (\boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}))_{1,D} \\ (\boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}))_{2,1} & \cdots & (\boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}))_{2,D} \\ \vdots & \ddots & \vdots \\ (\boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}))_{D,1} & \cdots & (\boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}))_{D,D} \end{bmatrix}$$

If again a Gaussian likelihood model is assumed, i.e., $\boldsymbol{r} \sim \mathcal{N}(\boldsymbol{f}(\boldsymbol{x}), \Sigma)$, where Σ represents a diagonal matrix with diagonal components $\{\sigma_d^2\}_{d=1}^{D}$, the predictive distribution for a new data point $\boldsymbol{x}_*$ has a closed form solution, i.e.,

$$p(\boldsymbol{f}(\boldsymbol{x}_*)|\mathcal{D}, \boldsymbol{x}_*, \boldsymbol{\Theta}) = \mathcal{N}(\boldsymbol{m}_*(\boldsymbol{x}_*), \boldsymbol{K}_*(\boldsymbol{x}_*, \boldsymbol{x}_*)) \tag{F.20}$$

where $\boldsymbol{\Theta}$ denotes the set of model parameters, and $\boldsymbol{m}(\boldsymbol{x}_*)$ and $\boldsymbol{K}_*(\boldsymbol{x}_*, \boldsymbol{x}_*)$ are defined as

$$\begin{aligned} \boldsymbol{m}_*(\boldsymbol{x}_*) &= \boldsymbol{K}_{\boldsymbol{x}_*}(\boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) + \boldsymbol{\Sigma})^{-1}\hat{\boldsymbol{r}}_c \\ \boldsymbol{K}_*(\boldsymbol{x}_*, \boldsymbol{x}_*) &= \boldsymbol{K}(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{K}_{\boldsymbol{x}_*}(\boldsymbol{K}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{X}}) + \boldsymbol{\Sigma})^{-1}\boldsymbol{K}_{\boldsymbol{x}_*}^T, \end{aligned}$$

with $\hat{\boldsymbol{r}}_c$ being a vector of length $N \times D$ that concatenates the observed output vectors, $\boldsymbol{\Sigma} = \Sigma \otimes \mathbf{I}_N$ is the Kronecker product between the noise covariance matrix and an identity matrix of size N, and $\boldsymbol{K}_{\boldsymbol{x}_*} = (\boldsymbol{K}(\boldsymbol{x}_*, \hat{\boldsymbol{X}}))_{d,d'}$. Note that if

the corresponding noisy predictions $\boldsymbol{r}_*$ are of interest, these may be achieved by adding Σ to the predictive variance expression [64].

If we further assume that the kernel function is separable, we can form the kernel as a product between an input kernel and an output kernel as

$$(\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}'))_{d,d'} = k_x(\boldsymbol{x}, \boldsymbol{x}')k_r(d, d'), \tag{F.21}$$

where k_x and k_r encode the covariances between the inputs and outputs, respectively. This is referred to as the intrinsic coregionalization model (ICM) in the Bayesian literature. Other more general kernel structures are sums of separable kernels, as in the linear model of coregionalization (LMC), and process convolutions. See [64, 65] for further details.

5 MODEL AVERAGING AND CONTEXT-SPECIFIC MODEL SELECTION

It is common that the statistical modeling of a problem domain result in multiple system representations that provide an adequate description of the observed data, as only limited amounts of data are available and different modeling approaches may be utilized. In such situations, one best system representation is typically selected from the ensemble according to some criterion, e.g., fit to data or predictive performance. After one model is selected, all inferences are made and conclusions drawn assuming that the selected model is the true model, thus ignoring model uncertainty.

In this section, we describe two avenues for dealing with model uncertainty in statistical modeling and decision-making, namely Bayesian model averaging (BMA), and an approach we will refer to as context-specific model selection (CSMS).

5.1 BAYESIAN MODEL AVERAGING

Bayesian model averaging is an approach to combine queries (predictions and forecasts) from an ensemble of models, where we are uncertain about the modeling assumptions. Consider an ensemble of system representations $\boldsymbol{\mathcal{M}} = \{\mathcal{M}_u\}_{u=1}^U$, where each $\mathcal{M}_u$ corresponds to one system representation. Using Bayesian model averaging, inferences are made by averaging over the ensemble models as

$$P(\Delta|\mathcal{D}^c) = \sum_{u=1}^{U} P(\Delta|\mathcal{M}_u, \mathcal{D}^c)P(\mathcal{M}_u|\mathcal{D}^c), \tag{F.22}$$

where Δ is the query assignment, e.g., a model prediction, and $\mathcal{M}_u$ is specified by the triple $\{\mathcal{G}, \boldsymbol{\Theta}_\mathcal{G}, \Lambda_\mathcal{G}\}$ in the case of BN emulators and continuous data. For this case, the model posterior probability $P(\mathcal{M}_u|\mathcal{D}^c)$ is given

by Eq. F.8. Thus, BMA provides a weighted average of the posterior predictions from each model, weighted by the posterior model probability, see e.g., [66–68].

5.2 CONTEXT-SPECIFIC MODEL SELECTION

Appreciating that scientific models are established to support decision-making in the context of systems performance management, the aim of systems modeling is to represent the available and relevant knowledge about systems in coherency with data obtained from e.g., observations and/or experiments to aid the ranking of decision alternatives.

Following [6–8], a system model $\mathcal{M}(a)$ in this context provides a mapping from input to output, conditional on a decision alternative a, which is measured in terms of utility. Figure F.1a shows this relationship. In general, the system performance is uncertain, and the optimal decision alternative is selected in accordance with Bayesian decision theory [69] and the axioms of utility theory [70] by maximizing the expected utility (benefit):

$$a^* = \arg\max_a(\mathbb{E}[\mathbb{U}(a)]). \tag{F.23}$$

If we now account for multiplicity in the model formulation, i.e., multiple competing system representations, Fig. F.1b shows the true system represented as a random event with possible realizations belonging to the set $\boldsymbol{\mathcal{M}} = \{\mathcal{M}_u\}_{u=1}^{U}$ of known components indexed by u, and s represents the index of a selected system representation. It might be so that some of the decision alternatives only have an influence on some of the competing system representations, and thus an optimization of the decision alternatives has to account for the selected system representation.

Following [5], the joint optimization over decision alternatives and system representations can expressed as

$$\begin{aligned}(s^*, a^*) = \arg\max_{s,a} \mathbb{U}(s,a) = &\arg\max_s \Big(P(u=s) \arg\max_a \Big(\mathbb{E}_{\boldsymbol{X}|s}[\mathbb{U}(a,\boldsymbol{X})] \Big) \\ &+ \mathbb{E}_{u' \in \boldsymbol{u} \setminus s} \Big[\mathbb{E}_{\boldsymbol{X}|u'}[\mathbb{U}(a^*,\boldsymbol{X})] \Big] \Big),\end{aligned} \tag{F.24}$$

where $a^* = \arg\max_a \mathbb{E}_{\boldsymbol{X}|s}[\mathbb{U}(a,\boldsymbol{X})]$. In Eq. F.24, the robustness of the decision, conditional on system choice, may be assessed as the ratio of the first term to the sum of the two terms. This ratio takes a value between 0 and 1 (1=robust) that indicates how sensitive the decision is to the possibility that the optimization is undertaken under an erroneous system assumption [71].

Through this approach, we emphasize that systems modeling should be seen as an integrated part of the decision optimization; the models do not

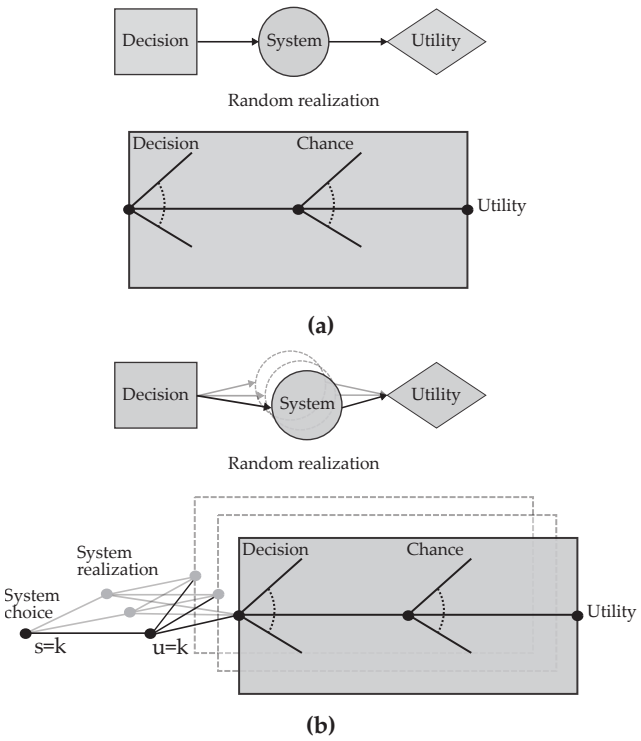


Figure F.1: Decision analysis including system choice: (a) a single system, (b) multiple possible systems.

have to be accurate outside the areas of the problem domain that have an impact on the decisions subject to optimization. By integrating systems modeling into the optimization of decision alternatives, all available knowledge can be utilized to optimize the expected utility for the considered system and thereby consistently rank decision alternatives, see [5–8] for further details.

6 A SIMPLE PRINCIPLE EXAMPLE ON CONTEXT-SPECIFIC MODEL SELECTION

6.1 INTRODUCTION

As a simple, principle example on the application of context-specific model selection (CSMS), we consider an example of probabilistic modeling from structural engineering. Assuming that we have N experiment outcomes from a concrete compression strength tests (MPa) collected in the vector $\boldsymbol{z}$. Based

on the information contained in $\boldsymbol{z}$, the task is now to inform a decision on the optimal design of the cross section of a short column, with the cross-sectional area A as design variable. It is assumed that the design cost of the column $C_D(A)$ is directly proportional to the cross-sectional area and thus may be modeled as

$$C_D(A) = C_d \times A, \tag{F.25}$$

where C_d is the cost per mm^2 of cross sectional area.

The design should however be safe – in the sense that the risk of failure must be accounted for. The expected value of the failure costs $\mathbb{E}[C_F(A)]$ is thus included in the decision problem as

$$\mathbb{E}[C_F(A)] = P_f(A) \times C_f, \tag{F.26}$$

where C_f is the failure cost, and $P_f(A)$ may be assessed through the probability of the event that the load l exceeds the compression capacity of the short column, i.e.,

$$P_f(A) = P(\{R(A) - l \leq 0\}), \tag{F.27}$$

where $R(A) = A \times Z$ is the compression capacity of the short column, and Z is a random variable representing the compression strength per mm^2.

The design optimization problem for a given load may now be written as

$$A^* = \arg\min_A (C_D(A) + \mathbb{E}[C_F(A)]). \tag{F.28}$$

In order to solve this decision problem, a model is needed to represent R in consistency with the observations contained in $\boldsymbol{z}$; however, at the same time the model must be chosen such as to facilitate the optimal design. To this end, it is assumed that the set of possible histograms, which can be established based on $\boldsymbol{z}$, are considered as model candidates. In this regard, the statistical uncertainty in the probability masses of the histograms, arising from the estimation based on the finite sample $\boldsymbol{z}$, must be accounted for.

The situation is further complicated by the fact that l is associated with uncertainty and represented in the problem by the random variable L with given probability distribution function. This situation may be framed under the CSMS framework by considering a set of realizations of L as possible system choices in Eq. F.24. That is, for each choice of loading system $l \in L$, an optimization is undertaken, which results in an optimal capacity model R and an associated optimal area A, and we choose the loading system, with associated R and A, that minimizes the expected cost, which accounts for the possible disbenefits of optimizing under an erroneous system assumption. This optimization problem may be formulated as

$$(l^*, A^*) = \arg\min_{l,A} (C_D(A) + \mathbb{E}[C_F(A, l)]). \tag{F.29}$$

For the numerical evaluations in the section, we assume $C_d = 1$ and $C_f = 100$, both in monetary units (MU). Moreover, the realizations of the compression strength are drawn from the probabilistic model: $\log(z) \sim \mathcal{N}(3.5, 0.13)$ (MPa), and the load is assumed to comply with the probabilistic model: $\log(l) \sim \mathcal{N}(5.3, 0.2)$ (Newton). One possible capacity model for a sample of 1000 realizations from Z is shown in Fig. F.2 together with the load distribution, assuming an area of $12\,\mathrm{mm}^2$.

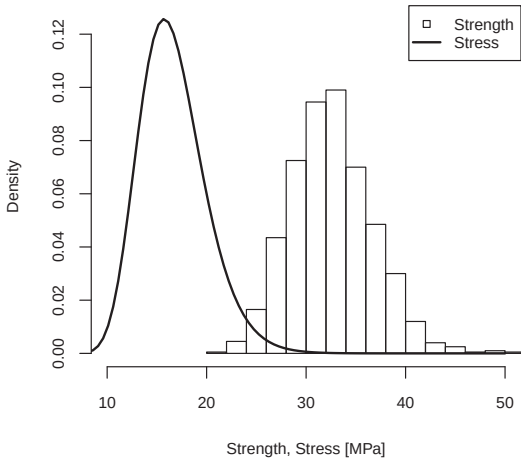


Figure F.2: One possible capacity model and the load model for $A = 12\,\mathrm{mm}^2$.

6.2 OPTIMIZATION PROCEDURE

The numerical optimization for the optimal discrete representation of $\boldsymbol{z}$ is undertaken as described in Sec. 3.2, under consideration of the objective function in Eq. F.29 in place of Eq. F.8, in a coordinate descent procedure, which successively adjusts the area to the current capacity model. In this regard, the statistical uncertainty in the probability masses of a given capacity model (discretization of $\boldsymbol{z}$) is represented by a Dirichlet distribution in the bin counts, which enables the calculation of the expected failure cost in Eq. F.26.

The optimization procedure described above forms an inner loop, which is wrapped by an outer loop in the loading systems. This outer loop proceeds in a divide and conquer mode by first representing the proposal load systems as the central value of an equal-width, discrete representation of the load distribution, and then successively subdividing the bin resulting in the lowest expected cost with respect to Eq. F.24 until convergence in the load. At convergence, the algorithm thus provides an optimal load (system), along with an associated optimal capacity model and area of the concrete column, but as the convergence guarantees of a greedy hill-climbing strategy like this

are only local, multiple random restarts are performed in order to increase our chances of finding a "good" local (global) minimum, and the resulting solutions are again weighted according to Eq. F.24.

6.3 RESULTS AND DISCUSSION

As mentioned, the optimization as described above result in an optimal triple $\{A^*, R^*, l^*\}$, where A^* is the area of the concrete column, R^* is the discrete, capacity model (discrete representation of $\boldsymbol{z}$), and l^* is the load (system). We found the optimal area and load to be 10.89 mm^2 and 257 Newton, respectively, and the corresponding, optimal capacity model is shown in Fig. F.3 for the data sample. It appears from the figure that as the model only needs to be accurate in the lower tail (near the load stress of 23.59 MPa) to reliably estimate the probability of failure, the body and upper tail of the distribution are represented by only one bin. The corresponding distribution of the failure probability, represented by draws from a Dirichlet distribution in the bin counts, is shown in Fig. F.4. Note that the mode in Fig. F.4 coincide with the sample estimate, i.e., $\hat{P}_f = 0.007$. For this solution, the expected cost is 11.68 MU, whereof 10.89 MU ($C_d \times A^*$) are attributed to the design cost and the remaining 0.79 MU (11.68 − 10.89) are attributed to failure costs.

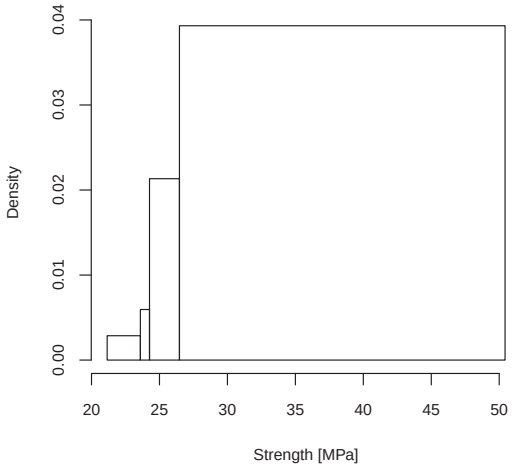


Figure F.3: Optimal capacity model.

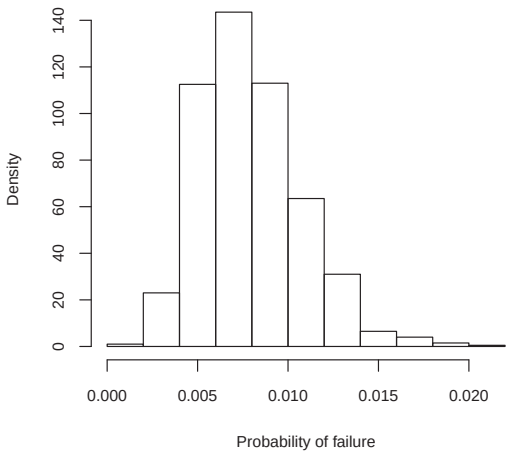


Figure F.4: Sample distribution for the probability of failure under the optimal capacity model.

7 AN EXAMPLE ON STORM EVENT MODELING

7.1 INTRODUCTION

In the Danish sector of the North Sea, the existing offshore facilities for oil and gas exploration are aging and some are about to reach their expected design service lives. Furthermore, new evidence indicates that the original design assumptions regarding the offshore wave load environment result in an underestimation of the extreme loads [72, 73]. Thus, the continued operation of the existing structures necessitates that the structures are reassessed with a refined modeling of loads, load effects and structural performances to ascertain that strategies for structural integrity management fulfill criteria for acceptable risks for personnel and reliability performances of structural systems.

In the following, basis is taken in this decision context to illustrate how the proposed framework for systems modeling may be adequately and efficiently applied on a realistic full-scale application with a particular focus on the probabilistic representation of the storm loading environment. In this regard, information contained in a database of measurements and hindcast simulations of storm events in the Danish North Sea is used to build a discrepancy model for the hindcast simulator, and then a system representation for the discrepancy-corrected hindcasts is defined for the region. Note that Sec. 8 builds on the findings in this section to solve a decision problem con-

cerning risk management in the context of wave in deck (WID) events.

7.2 METOCEAN DATABASE

This study considers the North See ocean environment of an area located approximately 220 km off the west coast of Denmark, where the ocean depth is approximately 40 m. The metocean records for the site includes wind fields and corresponding wave hindcast simulator outputs as well as wave measurements for a period of 37 years from 10th January 1979 to 30th December 2015. The hindcasts are produced for 23 locations using the spectral wind-wave model MIKE21 SW [74] and the hydrodynamic model MIKE21 HD [75], with climate forecast system reanalysis (CFSR) wind fields as input [76], and corresponding incomplete observations of the wave environment are available for seven of the locations, see also [8, 21].

By filtering of the records, 2187 storm events are detected for the reference period by exceedances of a threshold that is non-stationary with respect to season and direction [77], and a set of so-called characteristic variables are defined to summarize the storm events [32], see Tab. F.1. In this regard, both wind and wave directions are measured clockwise from north in degrees as the direction from which they are approaching, and current direction is defined as the direction towards which the current is flowing. Moreover, $Hm0$ and LgS are defined in terms of an equivalent, Gaussian bell-shaped storm profile with mean $Hm0$ and standard deviation proportional to σ_{st} as

$$Hm0_*(t) = Hm0 \cdot \exp\left(-\frac{1}{2}\left(\frac{t}{T02\sigma_{st}}\right)^2\right), \tag{F.30}$$

where $LgS = \log_{10}\sigma_{st}$, and $Hm0_*(t)$ is the equivalent storm significant wave height as a function of time. As noted by [32], the contribution to maximum short-term responses from sea states (1 hour) with $Hm0_*(t) < 0.75Hm0$ is negligible, thus an equivalent storm event only considers sea states for which $Hm0_*(t) \geq 0.75Hm0$ when assessing extreme responses. Further information about these metocean records may be found in [8, 21, 32].

In the following, we consider the data related to the seven platforms for which both hindcasts and incomplete measurements are available.

7.3 PROBABILISTIC MODELING

The probabilistic modeling scheme followed proceeds according to three steps: First, a GP discrepancy model (Sec. 4) is built for the sea state parameters $Hm0$, Tp, $T02$, and LgS based on the subset of complete measurements (4615 realizations) and the corresponding hindcast data of the sea state

Table F.1: Metocean storm records.

Variable	Explanation	Unit
Lng	Geographical longitude	deg
Ltt	Geographical latitude	deg
Dpt	Ocean depth	m
WSm	Wind speed	m/s
CSm	Current speed	m/s
WLa	Residual water level (surge + tide)	m
Ssn	Time of storm	deg
XDm	Wave direction	deg
WDm	Wind direction	deg
CDm	Current direction	deg
$Hm0$	Significant wave height	m
Tp	Peak wave period	s
$T02$	Second-moment wave period	s
LgS	$\log_{10}$ of duration parameter σ_{st}	-

parameters. The remaining variables are considered as covariates in the regression.

Second, an ensemble of BN structures (Sec. 3.2), and corresponding discretization policies, is learned based on the full set of mean-corrected hindcast data ($7 \times 2187 = 15309$ realizations), i.e., hindcast data corrected by the mean function of the GP discrepancy model according to Eq. F.13, by running the learning algorithm multiple times with different non-parametric bootstrap replicates of the data set. In this way, the posterior over BN structures is represented by a set of high scoring structures, and we avoid the need to sum over a (exponentially) large number of equivalence classes in the posterior evaluations [78]. Two DAGs having the same d-separation properties are said to belong to the same equivalent class. For two such DAGs, a probability distribution that factorizes along one of the DAGs also factorizes along the other [79]. Therefore, post-processing is conducted on the ensemble to remove BN structures belonging to the same equivalence class, and thus only the highest scoring BN structure within an equivalence class is kept for further analysis. The remaining BN structures are ranked according to their posterior probability (Eq. F.8), with the effect of $\Theta_{\mathcal{G}}$ marginalized out). That is, this process accounts for the model uncertainty of the BN representation, i.e. graph structure and discretization policy.

Third, the parameter distributions for each member of the ensemble of BN structures considered above are estimated. To account for the model uncertainty of the discrepancy process, a number of realizations are drawn from the discrepancy process and the parameters of the distributions for each realization of the discrepancy function are estimated. In this way, when conduct-

ing inference, initially a graph structure $\mathcal{G}^{(t)}$ is sampled from the ensemble of BN structures. For this graph, a realization of the parameter distribution is sampled, i.e., $\Theta_{\mathcal{G}}^{(t)}$, and finally also a realization of the parameters $\theta_{\mathcal{G}}^{(t)}$ is sampled. Thus, this process preserves the uncertainty of the discrepancy process as well as the parameter uncertainty of the BN representation (parameter vector).

7.4 RESULTS AND DISCUSSION

DISCREPANCY MODELING

A multi-output GP with an intrinsic co-regionalization model (ICM), featuring a squared exponential input kernel, is used for modeling the discrepancy between measurements of the sea state parameters and hindcast simulations, see Eqs. F.20 and F.21. In this regard, the remaining variables of the hindcast simulations (Tab. F.1) are considered as covariates in the regression. Figure F.5 shows the uncorrected hindcast simulator outputs and the mean-corrected simulator outputs for $Hm0$ on a held-out data set (20% of the data), respectively, plotted against the measurements. It appears that a significant reduction in model discrepancy is achieved by use of the discrepancy model, and the remaining scatter around the center line in Fig. F.5b can be attributed to measurement uncertainty. The same holds true for the remaining outputs, i.e., Tp, $T02$, and LgS, for which the plots are not shown to keep the presentation concise.

STORM EVENT MODELING

Model structures An ensemble of BN structures and corresponding optimal discretization policies are learned using different non-parametric bootstraps of the mean-corrected hindcast database. In this regard, the structure learning is constrained by prior causal understanding of the problem domain. First, the wind and wave properties are governed by the season, which is graphically encoded by imposing an edge going from Ssn to WSm, WDm, and $Hm0$; refer to Tab. F.1 for a listing of the variable names. Second, the depth is fully defined by the platform position, which is enforced by edges meeting head-to-head at Dpt from Lng, and Ltt. Furthermore, the discretization policy for some of the domain variables is predefined, namely seasonal, directional, and locational variables. The seasonal variable is defined based on the four seasons, the directions variables are defined based on eight directions (N, NE, E ...), and the locational variables are clustered into two groups (black and gray), as shown in Fig. F.6. A similar binary discretization of the location variables is learned in [8] using the entire hindcast data set (23 platforms).

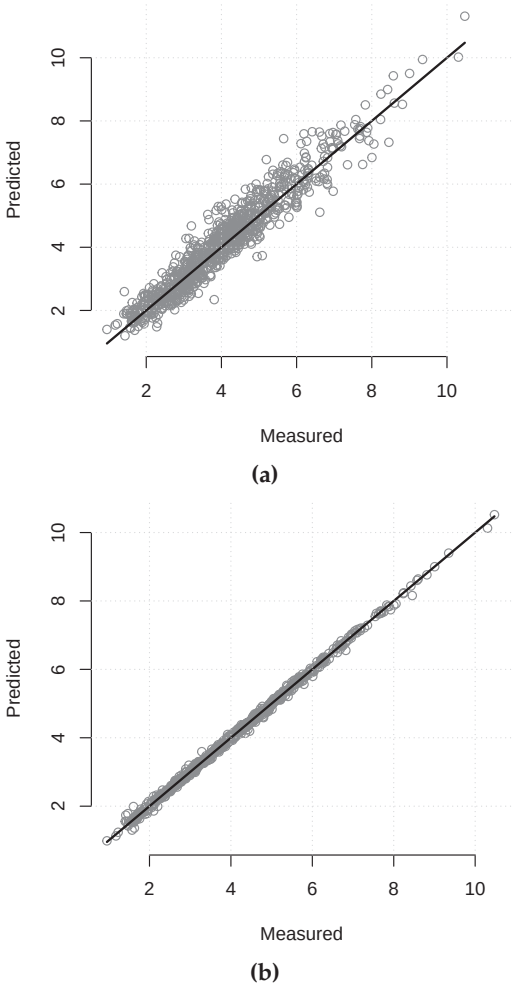


Figure F.5: Realizations of the variable $Hm0$: (a) measured vs uncorrected simulations, (b) measured vs mean-corrected simulations.

The individual ensemble graph structures and corresponding discretization policies are then grouped according to their equivalence class, and only the highest scoring BN structure within each equivalence class is considered for further analysis. The remaining, unique BN structures are then scored on the original database using Eq. F.8, with the effect of $\Theta_{\mathcal{G}}$ marginalized out, and ranked according to their score. When the scores are transformed from log space to probability space, only one model class has a relevant contribution. This may be expected since, in cases when the amount of data is large relative to the size of the model, the posterior will be sharply peaked

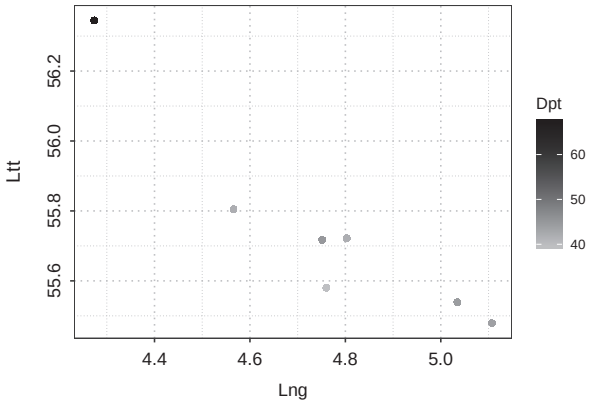


Figure F.6: Geographical location of data points in degrees longitude and latitude, and ocean depth in meters.

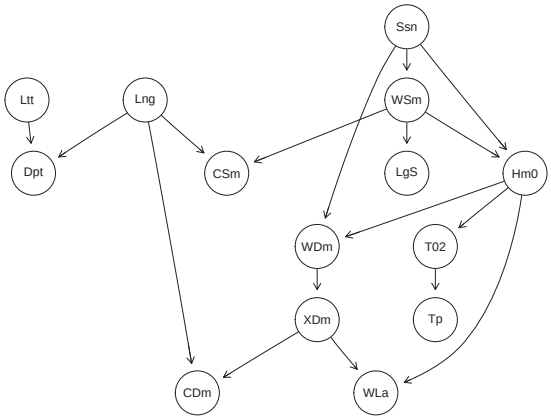


Figure F.7: BN structure.

around a single model, and this model will be a reasonably good approximation to the posterior [44, p.829]. This high scoring graph structure and the corresponding discretization appear in Fig. F.7 and Tab. F.2, respectively.

In Fig. F.7, a set of stochastic relations that were not prespecified appear in the DAG. For example, there are stochastic dependencies between e.g., $Hm0$ and $T02$, $T02$ and Tp, and WSm and CSm. Furthermore, considering the discretization, it is observed that the dynamically discretized variables are generally represented by 6-9 levels, except for the periods ($T02$ and Tp), which are represented by 17 levels. This added complexity in regard to the resolution of the periods is allowed in the optimization, as these variables constitute a separate branch in the DAG coming from $Hm0$, see Fig. F.7.

Table F.2: Discretization.

Variable	Bins	Remarks
Lng	2	predefined $((4, 4.42], (4.42, 5.5])$
Ltt	2	predefined $((55, 56.1], (56.1, 56.5])$
Dpt	2	predefined $((39, 56], (56, 67.5])$
WSm	7	learned
CSm	8	learned
WLa	6	learned
Ssn	4	predefined ($Spring$, $Summer$, ...)
XDm	8	predefined (N, NE, E, ...)
WDm	8	predefined (N, NE, E, ...)
CDm	8	predefined (N, NE, E, ...)
$Hm0$	9	learned
Tp	17	learned
$T02$	17	learned
LgS	7	learned

Figure F.8 shows the discrete marginal representation of $Hm0$ and $T02$, respectively, together with the corresponding empirical densities. It appears that the discretization policies for both variables provide a reasonable approximation to the continuous distributed probabilities. In the remainder of this example, basis is taken in the MAP graph (Fig. F.7) together with its corresponding discretization policy.

Model parameters In order to account for the model uncertainty related to the discrepancy function, 1000 realizations drawn from the discrepancy function are considered and the parameter distributions are estimated based on each individual realization. Thus, when sampling from the model, a realization is first sampled from the discrepancy function, which defines a parameters distribution $\Theta_{\mathcal{G}}$, and subsequently, a realization of the corresponding parameter vector $\theta_{\mathcal{G}}$ is sampled from the posterior distribution (Eq. F.4).

Applications of the storm event model Based on the procedure outlined in the foregoing, a fully specified BN model is established representing the typical composition of storm events for the considered geographical area. This may be utilized for different purposes. First, the model may be used to infer different (conditional) probability queries. As an example, Fig. F.9a shows the joint distribution of $Hm0$ and Tp without conditioning on observations of the remaining variables, and Fig. F.9b shows the joint distribution of $Hm0$ and Tp conditional on an observation of wave direction, i.e., $XDm = NW$. It may be observed that by conditioning on waves coming from NW, the small waves in the lower-left part become less likely, and the large waves in the

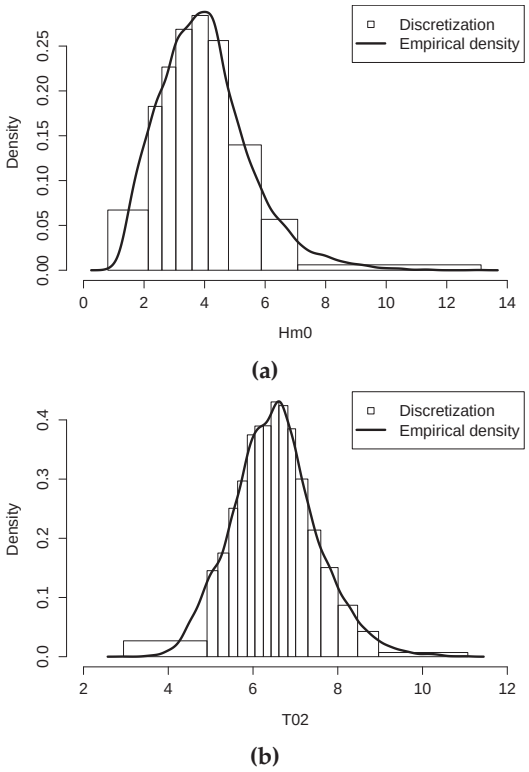
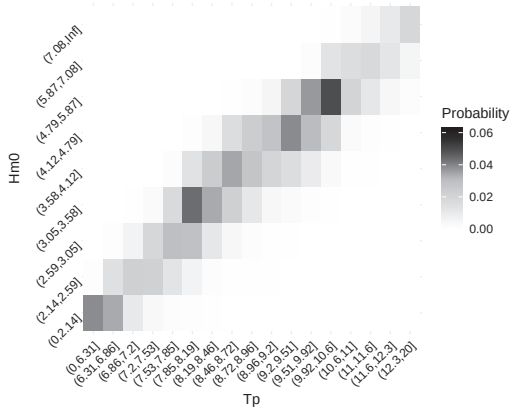


Figure F.8: Optimal discretization policies: (a) marginal distribution of $Hm0$, (b) marginal distribution of $T02$.

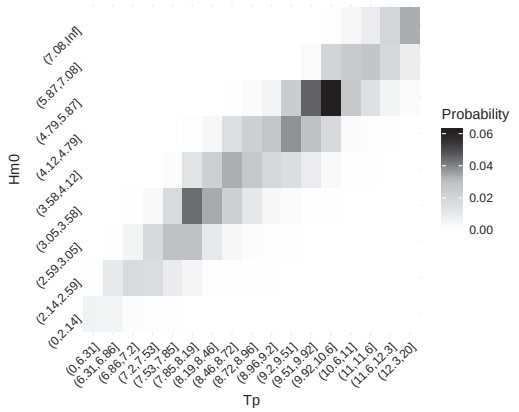
upper-right part become more likely.

Second, the model may be applied in the context of fatigue assessments. In this regard, samples of storm events over the lifetime of a structure may be generated, and Eq. F.30 can be used to define the content of the storms in terms of significant wave heights. These may subsequently be combined with an appropriate spectral representation of the short-term variability within sea states and a fatigue damage model to estimate fatigue damages in structural details. Figure F.10 shows 10 randomly sampled storm events for the lower-right cluster in Fig. F.6. A future research activity will address the development of a response surface in the storm event parameters (Tab. F.1) of the accumulated fatigue damage over a storm event.

Third, the model may be used to explore extreme events originating from different areas of the model. In this regard, focus may be directed on the population of storm events emanating from NW at the leftmost platform in the lower-right cluster in Fig. F.6, which fall in the highest $Hm0$ bin $((7.08, \text{inf}])$



(a)



(b)

Figure F.9: Joint distribution of $Hm0$ and Tp: (a) without evidence on the remaining variables, (b) with evidence on the wave direction, i.e., $XDm = NW$.

	WSm	CSm	WLa	Ssn	XDm	WDm	CDm	Hm0	Tp	T02	LgS
1	(15.8,17.7]	(0.194,0.252]	(0.132,0.302]	Winter	SW	S	SE	(4.12,4.79]	(8.46,8.72]	(6.43,6.61]	(3.67,3.73]
2	(17.7,20]	(0.0793,0.118]	(0.132,0.302]	Fall	N	NW	SW	(4.79,5.87]	(10.6,11]	(8.01,8.47]	(3.83,4.01]
3	(20,23.3]	(0.302,0.412]	(0.0143,0.132]	Winter	W	SW	E	(7.08,Inf]	(11.6,12.3]	(8.47,8.96]	(3.73,3.83]
4	(15.8,17.7]	(0.118,0.152]	(0.302,1.5]	Winter	W	W	SE	(4.79,5.87]	(10.6,11]	(7.6,8.01]	(3.83,4.01]
5	(13.3,14.3]	(0,0.0793]	(-0.0626,0.0143]	Spring	NW	N	S	(2.14,2.59]	(6.86,7.2]	(4.92,5.17]	(3.34,3.52]
6	(17.7,20]	(0.302,0.412]	(0.0143,0.132]	Fall	SW	SW	NE	(4.12,4.79]	(8.96,9.2]	(6.61,6.82]	(3.52,3.67]
7	(15.8,17.7]	(0.194,0.252]	(0.0143,0.132]	Winter	NW	W	SE	(3.58,4.12]	(8.72,8.96]	(6.43,6.61]	(3.73,3.83]
8	(15.8,17.7]	(0.194,0.252]	(0.132,0.302]	Fall	N	N	SW	(3.05,3.58]	(7.53,7.85]	(5.87,6.04]	(4.01,5]
9	(20,23.3]	(0.194,0.252]	(0.302,1.5]	Winter	NW	NW	SW	(5.87,7.08]	(11,11.6]	(7.6,8.01]	(3.73,3.83]
10	(17.7,20]	(0.252,0.302]	(0.0143,0.132]	Fall	W	SW	E	(4.12,4.79]	(8.96,9.2]	(7,7.29]	(3.52,3.67]

Figure F.10: Storm event realizations for lower-right cluster in Fig. F.6, i.e., $Lng \in (4.42, 5.5]$, $Ltt \in (55, 56.1]$, and $Dpt \in (39, 56]$.

Figure F.11: Extreme (mean-corrected) realizations at the leftmost platform of the lower-right cluster in Fig. F.6 for which $XDm = NW$ and $Hm0 > 7.08$.

in Fig. F.8a. On this basis, a joint extreme value model based on the realizations in the database that comply with these conditions may be established. Figure F.11 shows the realizations in the database that comply with these conditions. Note that the BN model additionally provides the probability of realizing a storm event in this area of the model, which when related to the reference period of the database, i.e., 37 years, gives an estimate of the annual probability of a storm event in this area of the model. As an example of an ULS assessment for a given structure, all areas of the model that have a relevant probability contribution in generating extreme responses may be considered, and a joint extreme value distribution model for these areas may be established, which will enable extrapolation to extreme fractile values. This is also on the agenda for future research.

There are several additional applications of the current model; Sec. 8 constitute one such by considering decision optimization in regard to platform

shut down and evacuation of personnel, provided information on an emerging storm event.

8 AN APPLICATION OF THE STORM EVENT MODEL AND CONTEXT-SPECIFIC MODEL SELECTION

8.1 INTRODUCTION

With basis in the storm event model formulated in Sec. 7, a risk-informed assessment of whether or not a set of offshore platforms in operation, i.e., the six platforms in the lower-right part of Fig. F.6, should be closed down and evacuated, given the information that a particular storm event is approaching. In this regard, the inherent clustering imposed by the storm event model is used to predict storm conditions at the lower-right cluster in Fig. F.6 from an observation at the upper-left cluster.

The decision problem, which is illustrated in Fig. F.12, concerns risk management in the context of wave in deck (WID) events. Based on observations on the emerging storm event, decision alternatives with respect to de-manning should be assessed and subsequently ranked by their expected value of benefit. Three decision alternatives are considered possible before the storm arrives, namely (i) do nothing (a_1); (ii) evacuate the personnel of the two most exposed platforms (i.e., the two leftmost platforms of the lower-right cluster in Fig. F.6) to the other four platforms (a_2); and (iii) evacuate the personnel of all six platforms to shore (a_3). The two platforms related to a_2 are considered more exposed, as we assume that these platforms are older and have a lower free air gab under the deck (17.5 m) than the other four platforms (20 m).

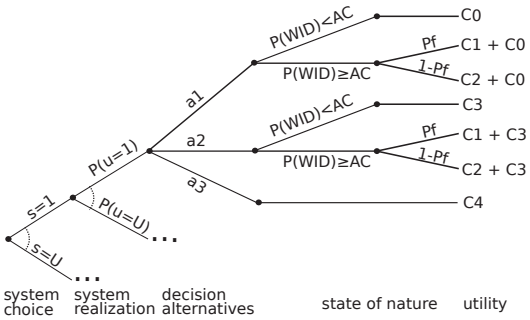


Figure F.12: Decision event tree for the storm risk management problem.

At the point in time where the emerging storm arrives at the location of the six platform, knowledge with respect to the storm characteristics may be observed and the probabilistic models of extreme waves may be updated accordingly. Hereafter, the probability of a WID event for all the platforms may be assessed and compared with the maximum allowed probability of WID events for manned platforms, through the corresponding WID risk acceptance criteria (AC). Depending on the risk management option initially selected before the arrival of the storm and the actual composition of wave events in the storm, a WID event will have different expected values of consequences. The base cost of the mitigation measures related to $\{a_d\}_{d=1}^{3}$ are $C_0 = 0\,\text{MU}$, $C_3 = 2 \times 20 \cdot 10^3\,\text{MU}$ and $C_4 = 6 \times 20 \cdot 10^3\,\text{MU}$, respectively, where $20 \cdot 10^3\,\text{MU}$ is assumed to be the cost per platform of a preventive evacuation, i.e., the case where a platform is de-manned before the arrival of the storm.

When the actual storm event is realized, we know whether $P(WID) \geq AC$, and if this turns out to be the case, the platforms must be de-manned. The associated costs of emergency evacuation are assumed to be $40 \cdot 10^3\,\text{MU}$ per platform (C_2). To account for the possibility that a WID event occurs at a platform before de-manning is completed, it is assumed that four 1-hour extreme sea states may occur before de-manning. For this reason, the probability of a WID event within four hours of extreme sea states, i.e., $P_f = P(WID|4\,\text{hr})$ is also assessed. If a WID event occurs at a platform before de-manning is completed, it is assumed that the lives of 10 crew members are lost. The corresponding costs are assumed to be $10 \times 25 \cdot 10^6\,\text{MU}$ (C_1), where $25 \cdot 10^6\,\text{MU}$ is the compensation cost of a human life, see also [80]. Note that only the cost related to evacuation and the loss of human lives are considered in the decision problem, i.e., costs related to platform damages are neglected.

8.2 PROBABILISTIC MODELING

To calculate the probability of a WID event at a platform, given storm event characteristics (discrete/cluster representation), we need to define the short-term variability of the storm event in terms of sea state content and crest heights within sea states. In this regard, a continuous-valued storm event may be defined using the realizations in the database that comply with the discrete storm representation by e.g., choosing a specific realization using the CSMS framework or a bootstrap approach. In either case, given the continuous representation of the storm event, the storm content can be defined in terms of significant wave heights (Eq. F.30), and for each sea state, the distribution of the hourly maximum crest height can be defined by adopting the model of [81]. This model is based on extensive laboratory-scale experiments performed in the wave basin at the Danish Hydraulic Institute (DHI), see [81, 82] for further details on the experimental data and its

post-processing. The model is provided in terms of a response surface in the sea state characteristics: Ursell number, wave steepness and wave directional spreading angle, and the water depth. Equivalently, it may be specified in terms of the significant wave height ($Hm0$), the sea state peak period (Tp), the wave directional spreading angle ($SpAng$), and the water depth (Dpt). The present BN model does not include a variable for $DirSp$, thus average quantities are used. However, the model may easily be extended to include this information as well.

8.3 RESULTS AND DISCUSSION

SHORT-TERM MODELING

It is assumed that a storm event with properties as in Tab. F.3 is observed at the upper-left cluster in Fig. F.6. Based on the storm event characteristics (Tab. F.3), the realizations in the database, which comply with the storm event, are considered as possible continuous realizations of the storm properties in the lower-right cluster. For each case and for each platform, the storm content in terms of significant wave heights, which have a relevant contribution to the probability of a WID event, is defined according to Eq. F.30. As a simplifying, conservative assumption, the significant wave height is not reduced over the storm event, but instead the storm peak significant wave height is used together with related quantities for all relevant sea states in a storm, i.e., sea state for which $Hm0_* \geq 0.75Hm0$.

Table F.3: Storm event related to short-term modeling.

Variable	Bin
Ssn	$Winter$
XDm	NW
$Hm0$	$(7.08, Inf]$
Tp	$(12.3, 20]$
LgS	$(3.52, 3.67]$

As an example, Fig. F.13 shows the predictive distribution for the maximum crest height (Cr) in a sea state (1 hour) corresponding to one of the realized (mean-corrected) storm events, i.e., $Hm0 = 10.7\,\text{m}$, $Tp = 16.0\,\text{s}$, and $Dpt = 41.9\,\text{m}$, at the leftmost platform of the lower-right cluster. Note that the directional spreading is assumed to be 19.0 deg.

DECISION OPTIMIZATION

For the platforms in the lower-right cluster of Fig. F.6, we compute the probability distribution of the maximum crest height for the storm peak sea states

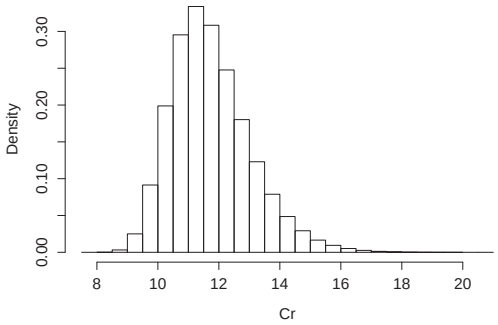


Figure F.13: Probability distribution function of the maximum crest height in a sea state (1 h) characterized by $Hm0 = 10.7\,\mathrm{m}$, $Tp = 16.0\,\mathrm{s}$, $DirSp = 19\,\mathrm{deg}$, and $Dpt = 41.9\,\mathrm{m}$.

corresponding to the individual realizations in the database that comply with the observation in Tab. F.3 at the upper-left cluster. In this regard, the uncertainty of the discrepancy model is propagated by considering variations in the defining sea state parameters corresponding to 1000 realizations from the discrepancy model.

For the numerical evaluations in the example, an annual acceptance criteria for a WID event of 10^{-5} is assumed, whereby the average criteria per storm amounts to $1-(1-10^{-5})^{1/59}$, given that 2178 storm events are observed in 37 years. Thus, the acceptance criteria per storm $AC = 1.695 \times 10^{-7}$. Moreover, six realized storm event in the database comply with the specifications in Tab. F.3, thus providing six competing systems for the decision optimization (Fig. F.12), which are equally likely, i.e., each realized once.

Given these assumptions, the decision problem is solved. The expected costs of the decision alternatives are assessed and ranked, and the optimal decision alternative for each of the six competing storm system realizations is provided in Tab. F.4. It appears that system realizations 1, 3, 4 and 5 agree on the optimal decision alternative being de-manning of the two most vulnerable platforms (a_2), i.e., the two leftmost platforms in the lower-right cluster of Fig. F.6, before the storm arrives at the platforms. These system choices all result in an expected cost of $0.48 \cdot 10^5\,\mathrm{MU}$ when accounting for the competing systems, as only system realization 1 predicts an additional platform for which $P(WID) \geq AC$, besides the two vulnerable platforms that are already evacuated.

The two remaining system realizations, i.e., 2 and 6, are less severe, and the optimal action under these systems is thus to do nothing (a_1). Within the CSMS framework, these system choices receive a significant penalty when evaluating a_1 under the remaining systems, as the expected failure cost is

Table F.4: Solution to the storm evacuation decision problem.

System	Optimal action	Expected cost
1.	a_2 (evacuate two)	$0.48 \cdot 10^5$ MU
2.	a_1 (do nothing)	$7.06 \cdot 10^5$ MU
3.	a_2 (evacuate two)	$0.48 \cdot 10^5$ MU
4.	a_2 (evacuate two)	$0.48 \cdot 10^5$ MU
5.	a_2 (evacuate two)	$0.48 \cdot 10^5$ MU
6.	a_1 (do nothing)	$7.06 \cdot 10^5$ MU

underestimated. In this case, the expected cost is $7.06 \cdot 10^5$ MU when accounting for the competing systems, which is roughly 15 times as much as the expected cost found above for a_2.

In the decision optimization, the same expected cost is found for system realizations 1, 3, 4 and 5, under the corresponding optimal action a_2. This means that these system realizations are equally good at informing the decision in this specific decision context, but as only system realization 1 has an additional relevant contribution to the expected failure cost, apart from the contribution from the two most vulnerable platforms, this may be preferred. This is equivalent to choosing the most robust system realization, as discussed in Sec. 5.2.

9 CONCLUSIONS AND OUTLOOK

In the present contribution, a Bayesian probabilistic framework for the representation – or modeling – of systems in the face of incomplete knowledge and uncertainty is outlined. Its application is illustrated both on a simple principle example, as well as on a full-scale application considering the probabilistic modeling of offshore storm events acting on oil and gas facilities.

Starting point is taken in a general review of current approaches for probabilistic load environment modeling in offshore engineering, after which focus is directed on the most recent developments in the area of Bayesian networks and Gaussian processes. This is followed by a description of how the ever-prevailing challenge of possible competing system representations consistently may be accounted for in the decision analysis, which the system representations serve to inform. The proposed and described approaches are then applied to a very simple but principally representative case of estimating a probabilistic model for the compression strength of concrete in the context of classical risk-based decision optimization of structural reliability. From this example, the fact that optimal model selection depends strongly on the decision context is evident, as the derived discrete probability mass model differs significantly from what would be found through the classical approach.

Subsequently, we apply the Bayesian network and Gaussian process-based systems representation framework to establish a probabilistic representation of storm events based on a large sample hindcast and measurement database from the Danish part of the North Sea. This part of the modeling could be said to follow an advanced but classical approach to modeling, which explicitly includes the representation of possible different systems that might critically affect optimal decision-making. Thus, to illustrate this, we finally provide an example in which the developed storm event model is integrated into a decision analysis, which concerns the ranking of decision alternatives with respect to de-manning of offshore platforms, provided information on an emerging storm event. This example underlines the significance of accounting for possible competing systems in the context of the decision analysis, which the system representations aim to inform.

Based on our experience from the ongoing research, it appears that for complex decision problems like the de-manning options ranking problem addressed in the present contribution, the required analysis efforts will be somewhat extensive. Surely, there is some potential for optimizing analysis strategies and improving algorithms, but no closed form solutions is likely to be identified. However, in more standard and simple decision contexts, similar to the principle example we provided in the present study, there actually could be closed form solutions. The identification of these is one of our objectives for future research.

ACKNOWLEDGMENTS

The authors kindly acknowledge the Danish Underground Consortium (Total E&P Denmark, Noreco & Nordsøfonden) for granting the permission to publish this work. A special thanks goes to Shell Research Ltd., Danish Hydraulic Institute, Total E&P, and Hans Fabricius Hansen (Haw Metocean) for their support in the project. This research has received funding from the Danish Hydrocarbon Research and Technology Centre (DHRTC) under the Structural Integrity and Lifetime Evaluation program.

REFERENCES

[1] European Committee for Standardization (CEN), "Petroleum and natural gas industries - Fixed steel offshore structures," *EN ISO 19902:2008*, 2008.

[2] ——, "Petroleum and natural gas industries - Specific requirements for offshore structures - Part 1: Metocean design and operating considerations," *EN ISO 19901-1:2015*, 2015.

[3] M. H. Faber, J. D. Sørensen, and I. B. Kroon, "Optimal fatigue testing: A reassessment tool," in *Proceedings of IABSE Colloquium on Remaining Structural Capacity*. International Association for Bridge and Structural Engineering, 1993, pp. 61–68.

[4] J. D. Sørensen, S. Engelund, and M. H. Faber, "Reliability analysis and test planning using capo-test for existing structures," in *Applications of Statistics and Probability - Civil Engineering Reliability and Risk Analysis*, 2000, pp. 723–730.

[5] M. H. Faber and M. A. Maes, "Epistemic uncertainties and system choice in decision making," in *Ninth International Conference on Structural Safety and Reliability, ICOSSAR*, 2005, pp. 3519–3526.

[6] L. Nielsen, S. T. Glavind, J. Qin, and M. H. Faber, "Faith and fakes–dealing with critical information in decision analysis," *Civ Eng Environ Syst*, vol. 36, no. 1, pp. 32–54, 2019.

[7] S. T. Glavind and M. H. Faber, "A framework for offshore load environment modeling," in *37th International Conference on Offshore Mechanics and Arctic Engineering, OMAE*, 2018, p. 77674.

[8] ——, "A framework for offshore load environment modeling," *Journal of Offshore Mechanics and Arctic Engineering*, vol. 142, no. 2, p. 021702, 2020. [Online]. Available: https://doi.org/10.1115/1.4045190

[9] Joint Committee of Structural Safety, "Part 2: Load models," *JCSS: Probabilistic Model Code*, 2015.

[10] A. Der Kiureghian and P.-L. Liu, "Structural reliability under incomplete probability information," *Journal of Engineering Mechanics*, vol. 112, no. 1, pp. 85–104, 1986.

[11] O. Ditlevsen, "Stochastic model for joint wave and wind loads on offshore structures," *Structural Safety*, vol. 24, no. 2-4, pp. 139–163, 2002.

[12] E. M. Bitner-Gregersen and S. Haver, "Joint environmental model for reliability calculations," in *First International Offshore and Polar Engineering Conference*. International Society of Offshore and Polar Engineers, 1991, pp. 246–253.

[13] W. Dong, T. Moan, and Z. Gao, "Long-term fatigue analysis of multi-planar tubular joints for jacket-type offshore wind turbine in time domain," *Engineering Structures*, vol. 33, no. 6, pp. 2002–2014, 2011.

[14] M. Sklar, "Fonctions de repartition an dimensions et leurs marges," *Publ. inst. statist. univ. Paris*, vol. 8, pp. 229–231, 1959.

[15] E. Vanem, "Joint statistical models for significant wave height and wave period in a changing climate," *Marine Structures*, vol. 49, pp. 180–205, 2016.

[16] X. Li, Q. Ding, and J. Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering and System Safety*, vol. 172, pp. 1–11, 2018.

[17] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[18] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[19] S. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Pearson Education Limited, 2014.

[20] D. J. Peres, C. Iuppa, L. Cavallaro, A. Cancelliere, and E. Foti, "Significant wave height record extension by neural networks and reanalysis wind data," *Ocean Modelling*, vol. 94, pp. 128–140, 2015.

[21] M. Jones, H. F. Hansen, A. R. Zeeberg, D. Randell, and P. Jonathan, "Uncertainty quantification in estimation of extreme environments," *Coastal Eng*, vol. 141, pp. 36–51, 2018.

[22] J. Beirlant, Y. Goegebeur, J. Segers, and J. L. Teugels, *Statistics of extremes: theory and applications*. John Wiley & Sons, 2006.

[23] P. Jonathan and K. Ewans, "Statistical modelling of extreme ocean environments for marine design: A review," *Ocean Engineering*, vol. 62, pp. 91–109, 2013.

[24] S. Haver, "Wave climate of northern norway," *Applied Ocean Research*, vol. 7, no. 2, pp. 85–92, 1985.

[25] R. Montes-Iturrizaga and E. Heredia-Zavoni, "Environmental contours using copulas," *Applied Ocean Research*, vol. 52, pp. 125–139, 2015.

[26] ——, "Multivariate environmental contours using c-vine copulas," *Ocean Engineering*, vol. 118, pp. 68–82, 2016.

[27] J. Heffernan and J. Tawn, "A conditional approach for multivariate extreme values," *Journal of the Royal Statistical Society Series B – statistical Methodology*, vol. 66, no. 3, pp. 497–530, 2004.

[28] P. Jonathan, J. Flynn, and K. Ewans, "Joint modelling of wave spectral parameters for extreme sea states," *Ocean Engineering*, vol. 37, no. 11-12, pp. 1070–1080, 2010.

[29] P. Jonathan, K. Ewans, and J. Flynn, "Joint modelling of vertical profiles of large ocean currents," *Ocean Engineering*, vol. 42, pp. 195–204, 2012.

[30] ——, "On the estimation of ocean engineering design contours," *Journal of Offshore Mechanics and Arctic Engineering*, vol. 136, no. 4, p. 041101, 2014.

[31] E. Ross, S. Sam, D. Randell, G. Feld, and P. Jonathan, "Estimating surge in extreme north sea storms," *Ocean Engineering*, vol. 154, pp. 430–444, 2018.

[32] H. F. Hansen, D. Randell, A. R. Zeeberg, and P. Jonathan, "Directional–seasonal extreme value analysis of north sea storm conditions," *Ocean Engineering*, vol. 195, p. 106665, 2020.

[33] N. Fenton and M. Neil, *Risk assessment and decision analysis with Bayesian networks*. CRC Press, 2018.

[34] M. H. Faber, I. B. Kroon, E. Kragh, D. Bayly, and P. Decosemaeker, "Risk assessment of decommissioning options using bayesian networks," *Journal of Offshore Mechanics and Arctic Engineering*, vol. 124, no. 4, pp. 231–238, 2002.

[35] J. Ren, I. Jenkinson, J. Wang, D. L. Xu, and J. B. Yang, "A methodology to model causal relationships on offshore safety assessment focusing on human and organizational factors," *Journal of Safety Research*, vol. 39, no. 1, pp. 87–100, 2008.

[36] J. E. Vinnem, R. Bye, B. A. Gran, T. Kongsvik, O. M. Nyheim, E. H. Okstad, J. Seljelid, and J. Vatn, "Risk modelling of maintenance work on major process equipment on offshore petroleum installations," *Journal of Loss Prevention in the Process Industries*, vol. 25, no. 2, pp. 274–292, 2012.

[37] N. Khakzad, F. Khan, and P. Amyotte, "Quantitative risk analysis of offshore drilling operations: A bayesian approach," *Safety Science*, vol. 57, pp. 108–117, 2013.

[38] A. Sarwar, F. Khan, L. James, and M. Abimbola, "Integrated offshore power operation resilience assessment using object oriented bayesian network," *Ocean Engineering*, vol. 167, pp. 257–266, 2018.

[39] N. Norazahar, F. Khan, B. Veitch, and S. MacKinnon, "Dynamic risk assessment of escape and evacuation on offshore installations in a harsh environment," *Applied Ocean Research*, vol. 79, pp. 1–6, 2018.

[40] S. Loughney and J. Wang, "Bayesian network modelling of an offshore electrical generation system for applications within an asset integrity case for normally unattended offshore installations," *Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment*, vol. 232, no. 4, pp. 402–420, 2018.

[41] C. M. Bishop, "Model-based machine learning," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, p. 20120222, 2013.

[42] J. Pearl, *Probabilistic Reasoning in Intelligent Systems. Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[43] M. Scutari and J.-B. Denis, *Bayesian networks: with examples in R*. CRC press, 2014.

[44] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.

[45] D. Barber, *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[46] R. E. Neapolitan, *Learning bayesian networks*. Pearson Prentice Hall, 2004.

[47] D. M. Chickering, "Learning bayesian networks is np-complete," in *Learning from Data: Artificial Intelligence and Statistics V*. Springer-Verlag, 1996, pp. 121–130.

[48] M. Scutari, C. E. Graafland, and J. M. Gutiérrez, "Who learns better bayesian network structures: Accuracy and speed of structure learning algorithms," *International Journal of Approximate Reasoning*, vol. 115, pp. 235–253, 2019.

[49] S. Monti and G. F. Cooper, "A multivariate discretization method for learning bayesian networks from mixed data," in *Fourteenth International Conference on Uncertainty in Artificial Intelligence*, 1998, pp. 404–413.

[50] K. Vogel, "Applications of bayesian networks in natural hazard assessments," Ph.D. dissertation, University of Potsdam, 2013.

[51] N. Friedman and M. Goldszmidt, "Discretizing continuous attributes while learning bayesian networks," in *Thirteenth International Conference on Machine Learning*, 1996, pp. 157–165.

[52] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*. CRC press, 2013.

[53] K. B. Korb and A. E. Nicholson, *Bayesian artificial intelligence*. CRC press, 2010.

[54] R. Daly, Q. Shen, and S. Aitken, "Learning bayesian networks: approaches and issues," *The Knowledge Engineering Review*, vol. 26, no. 2, p. 99–157, 2011.

[55] Z. Ji, Q. Xia, and G. Meng, "A review of parameter learning methods in bayesian network," in *Advanced Intelligent Computing Theories and Applications*, 2015, pp. 3–12.

[56] N. Friedman, "Learning belief networks in the presence of missing values and hidden variables," in *Fourteenth International Conference on Machine Learning*, 1997, pp. 125–133.

[57] ——, "The Bayesian structural EM algorithm," in *Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1998, pp. 129–138.

[58] M. A. Tanner and W. H. Wong, "The calculation of posterior distributions by data augmentation," *Journal of the American statistical Association*, vol. 82, no. 398, pp. 528–540, 1987.

[59] L. A. Lee, K. S. Carslaw, K. J. Pringle, G. W. Mann, and D. V. Spracklen, "Emulation of a complex global aerosol model to quantify sensitivity to uncertain parameters," *Atmospheric Chemistry and Physics*, vol. 11, no. 23, pp. 12 253–12 273, 2011.

[60] F. A. Diazdelao and S. Adhikari, "Gaussian process emulators for the stochastic finite element method," *International Journal for Numerical Methods in Engineering*, vol. 87, no. 6, pp. 521–540, 2011.

[61] K. Sham Bhat, M. Haran, R. Olson, and K. Keller, "Inferring likelihoods and climate system characteristics from climate models and multiple tracers," *Environmetrics*, vol. 23, no. 4, pp. 345–362, 2012.

[62] R. Olson, R. Sriver, W. Chang, M. Haran, N. M. Urban, and K. Keller, "What is the effect of unresolved internal climate variability on climate sensitivity estimates?" *Journal of Geophysical Research Atmospheres*, vol. 118, no. 10, pp. 4348–4358, 2013.

[63] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT press, 2006.

[64] M. A. Álvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review," *Foundations and Trends in Machine Learning*, vol. 4, no. 3, pp. 195–266, 2012.

[65] E. V. Bonilla, K. M. Chai, and C. Williams, "Multi-task gaussian process prediction," in *Advances in neural information processing systems*, 2008, pp. 153–160.

[66] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, "Bayesian model averaging: a tutorial," *Statistical science*, vol. 14, no. 4, pp. 382–417, 1999.

[67] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*. Springer New York, 2009.

[68] S. T. Glavind, H. Brüske, and M. H. Faber, "On normalized fatigue crack growth modeling," in *39th International Conference on Offshore Mechanics and Arctic Engineering, OMAE*, 2020, p. V02AT02A037.

[69] H. Raiffa and R. Schlaifer, *Applied statistical decision theory*. MIT Press, 1961.

[70] J. von Neumann and O. Morgenstern, *Theory of games and economic behavior*. Princeton University Press, 1953.

[71] M. H. Faber, "On the governance of global and catastrophic risks," *International Journal of Risk Assessment and Management*, vol. 15, no. 5-6, pp. 400–416, 2011.

[72] J. Tychsen, S. Risvig, H. Fabricius Hansen, N. Ottesen Hansen, and F. Stevanato, "Summary of the impact on structural reliability of the findings of the tyra field extreme wave study 2013-2015," in *Third Offshore Structural Reliability Conference, OSRC2016, Stavanger, Norway*, 2016, pp. 1–12.

[73] J. Tychsen and M. Dixen, "Wave kinematics and hydrodynamic loads on intermediate water depth structures inferred from systematic model testing and field observations–tyra field extreme wave study 2013-15," in *Third Offshore Structural Reliability Conference*, 2016, pp. 1–10.

[74] O. R. Sørensen, H. Kofoed-Hansen, M. Rugbjerg, and L. S. Sørensen, "A third-generation spectral wave model using an unstructured finite volume technique," in *29th International Conference on Coastal Engineering*, 2004, pp. 894–906.

[75] Danish Hydraulic Institute, "MIKE21 and MIKE3 Flow Model FM, Hydrodynamics and transport Module," DHI, Tech. Rep., 01 2017.

[76] S. Saha, S. Moorthi, X. Wu, J. Wang, S. Nadiga, P. Tripp, D. Behringer, Y. T. Hou, H. Y. Chuang, M. Iredell, M. Ek, J. Meng, R. Yang, M. P. Mendez, H. Van Den Dool, Q. Zhang, W. Wang, M. Chen, and E. Becker, "The NCEP climate forecast system version 2," *J Climate*, vol. 27, no. 6, pp. 2185–2208, 2014.

[77] K. Ewans and P. Jonathan, "The effect of directionality on northern north sea extreme wave design criteria," *J Offshore Mech Arct*, vol. 130, no. 4, p. 041604, 2008.

[78] N. Friedman, M. Goldszmidt, and A. Wyner, "Data analysis with Bayesian networks: A bootstrap approach," in *Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 196–205.

[79] T. J. Koski and J. M. Noble, "A review of bayesian networks and structure learning," *Mathematica Applicanda*, vol. 40, no. 1, pp. 51–103, 2012.

[80] K. Fischer, C. Viljoen, J. Köhler, and M. H. Faber, "Optimal and acceptable reliabilities for structural design," *Structural Safety*, vol. 76, pp. 149–161, 2019.

[81] M. Schubert, Y. Wu, J. Tychsen, M. Dixen, M. H. Faber, J. D. Sørensen, and P. Jonathan, "On the distribution of maximum crest and wave height at intermediate water depths," *Ocean Engineering*, vol. 217, no. 1 December, p. 107485, 2020.

[82] H. Bredmose, M. Dixen, A. Ghadirian *et al.*, "Derisk: Accurate prediction of uls wave loads. outlook and first results," *Energy Procedia*, vol. 94, pp. 379–387, 2016.

ISSN (online): 2446-1636
ISBN (online): 978-87-7210-928-2 AALBORG UNIVERSITY PRESS