A Bayesian-Based Framework for Making Inspection and Maintenance Decisions from Data and Expert Knowledge

Queen Mary
University of London

Haoyuan Zhang

School of Electronic Engineering and Computer Science
Queen Mary University of London

This dissertation is submitted for the degree of
Doctor of Philosophy

Submitted: March 2019
Amended: July 2019

Declaration

I, Haoyuan Zhang, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged and my contribution indicated. Previously published material is also acknowledged.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Haoyuan Zhang
Submitted: March 2019
Amended: July 2019

Acknowledgements

First of all, I would like to express my appreciation to all those who have helped and supported me during the journey of my doctoral training.

I would like to thank my primary supervisor Dr William Marsh for his support throughout my time at Queen Mary. William has patiently taught me many things including the importance of holding curiosity in different fields, the art of compelling research storytelling, and the commitment to rigorous research. I have been benefited greatly from his guidance and have inspired by his expertise, hard work, and attention to detail.

Special thanks to other panel members of my research: Professor Norman Fenton and Professor Martin Neil. Their spot-on comments and valuable suggestions gave me many inspirations and positioned me on the right track of my research.

I'd also like to thank Dr Christopher Joyner for his comments on my thesis, and other members of Risk and Information Management Group for the adventures with Bayesian networks.

Finally, I would like to thank all my family members and friends, for their continuous encouragement, support and unwavering love throughout everything.

Abstract

It is estimated that more than one-third of current infrastructure maintenance expenditure is wasted through poor decision-making. To make better decisions about maintenance, there is a need to provide better predictions of asset deterioration, and further, to use this information to plan inspections and appropriate repair actions. A number of statistical modelling techniques have been proposed to predict deterioration. However, these approaches can be difficult to apply in practice, for example when the time of deterioration is only known approximately from periodic inspections. Also, these approaches lack an easy way to incorporate knowledge about the deterioration process that can readily be considered when judgements are made by experienced maintainers. Moreover, in practice, the size of available datasets on deterioration is often limited; hence there is a need to blend data with knowledge.

This thesis presents a framework for predicting deterioration and reasoning about the effects of repair using both the available data and expert knowledge that can support inspection and maintenance-related decisions. The framework uses Bayesian modelling, combining two types of Bayesian approaches: Bayesian statistical models and Bayesian Networks (BNs). Bayesian statistical models are used to estimate the parameter of statistical distributions, modelled as continuous variables. On the other hand, BNs model causal or influential relationships between (primarily) discrete variables to make predictions and can be based on elicited knowledge. This thesis builds on earlier work that combines these two forms of model, with both the continuous variables from Bayesian statistical models and the discrete variables of BNs. We refer this type of model to as a hybrid BN. The use of hybrid BNs is possible using an already existing algorithm that dynamically discretises continuous variables in a BN. BNs within the framework can be combined to model the different aspects of deterioration needed in different circumstances. The rate of deterioration can be learnt from censored deterioration data inferred from inspection records and knowledge elicited from engineers. Asset sharing similar characteristics can be grouped, and when a group contains only a few instances in the available data, data from related groups can be used to constrain the parameter learning. Deterioration through multiple condition states can be modelled. The deterioration of different components of complex structures can be combined. Finally, we model the effect of repair actions and show how to plan maintenance.

A case study using data from the US National Bridge Inventory is used to validate the deterioration prediction models. We show how real-world inspection records can be integrated with engineering knowledge to predict the deterioration. Compared with other published approaches, the proposed models show better performance, especially when the group of similar assets is small. We then apply the models to reason about inspection and maintenance-related decisions. We use case studies of maintenance practices in the GB and US to show how the models can be used to assist both operational and strategic maintenance decision making.

Many features of the proposed framework need to be adapted and combined to create a maintenance model applicable in a particular circumstance. Examples include the number of deterioration states, the decomposition of assets into components and the grouping of assets. The challenge is to create a complex and large-scale asset management system to allow a maintenance analyst to apply the framework, without needing expertise in Bayesian modelling. By representing our framework as a set of generic models using an extended form of BN – a probabilistic relational model – we show, with a simple prototype, how such a system could be realised.

Table of contents

List of figures

List of tables

List of abbreviations

ADT	Average Daily Traffic
BCMI	Bridge Condition Marking Index
BN	Bayesian Network
BPE	Bayesian Parameter Estimation
CBM	Condition-Based Maintenance
CCF	Common Cause Failure
cdf	cumulative distribution function
CPN	Coloured Petri Net
CPT	Conditional Probability Table
EMGTPA	Equivalent Million Gross Tonnes Per Annual
ETA	Event Tree Analysis
FTA	Fault Tree Analysis
MAP	Maximum A Posteriori
MBSA	Model-Based Safety Assessment
MBSE	Model-Based System Engineering
MCDM	Multi-Criteria Decision Making
MCMC	Markov chain Monte Carlo
MDA	Mean Decrease Accuracy

MDI	Mean Decrease Impurity
MLE	Maximum Likelihood Estimation
MSS	Multi-State System
MTBF	Mean Time Between Failure
MTTF	Mean Time To Failure
NBI	National Bridge Inventory
OO	Object Oriented
OOBN	Object-Oriented Bayesian Network
pdf	probability distribution function
PRM	Probabilistic Relational Model
RPS	Ranked Probability Score
SCMI	Structures Condition Marking Index
SD	Structurally Deficient
TBM	Time-Based Maintenance

Chapter 1

Introduction

As of 2017, the National Bridge Inventory (NBI) (the national database of bridges in the United States) showed that 54,560 bridges (out of a total of 615,002) were reported as structurally deficient. The associated expenditure for maintenance is considerable – for instance, the Organisation for Economic Co-operation and Development (OECD) estimated that from 2010 to 2030, the annual cost of road and rail maintenance would be approximately 220 billion and 50 billion United States Dollars respectively [134]. Furthermore, Heng et al. [68] estimated that with current practices, more than a third of infrastructure maintenance expenditure is wasted through poor decision-making. The need for an effective maintenance strategy for managing such decisions is therefore paramount.

An effective maintenance strategy for maintaining infrastructure assets requires careful planning of both inspection and maintenance activities. The inspections aim to detect any damage and identify the asset's underlying condition so that future inspections and maintenance can be scheduled. Alternatively, maintenance is performed to either repair any damage that has occurred or to improve the overall condition to help prevent any future damage. These two activities ensure the infrastructure asset operates within an acceptable level of safety and reliability. An effective maintenance strategy is therefore necessary for increasing the expected life expectancy of the structure and reduce its maintenance expenditure [51].

Inspection activity can be classified into two types: visual inspection and detail inspection. Visual inspection looks for outward signs of damage, such as missing fasteners and cracks; its decision can be supported by techniques like fault detection and fault pattern recognition (see a review from Heng et al. [68]). In contrast, detail inspection seeks to determine the conditions of the underlying parts of the infrastructure, which is often more complicated to perform and generally more expensive. For example, for a bridge with hidden critical elements that cannot be observed directly, we may need to use intrusive or non-intrusive examination methods to estimate its condition. To support detail inspection decision, we can perform a preliminary assessment by predicting the condition of the asset from its deterioration.

The process of deterioration is described by the decrease in asset condition over time, such as a bridge deteriorates from a good condition to a structurally deficient condition. At the same time, historical inspection records provide information about the asset's previous conditions over time. We can use these records to infer relevant data to model the deterioration process of an asset. By doing so, we can predict the condition of an asset by estimating how soon it is likely to deteriorate into an unacceptable level in the future. This can support the decisions on inspection and answer questions like when to inspect the asset or which asset we should inspect.

After we identified the damage or condition of an asset from inspection activity, we can decide its corresponding maintenance activity. Decisions on maintenance activity mainly comprise of the selection of repair action and the plan to perform maintenance. Repair actions can range from no action when the asset is in an acceptable state, to replacement when the asset on the edge of failure, with several intermediate actions to restore the asset to a better decay level [100]. We may want to suggest repair action given an objective, for example, minimum cost or an acceptable level of system reliability.

The choice of repair action is one of the keys to plan maintenance. Sometimes, from a life-cycle point of view, it is more cost-effective to wait for an asset to deteriorate further rather than taking repair action immediately. For example, we may have a longer life expectancy to repair an asset in a worse condition using a major repair action, than repair with minor repair action in a better state. Planning maintenance over a time horizon concerns with multiple intervention cycles, where each cycle considers the selection of repair action and the further deterioration after repair. As a result, the maintenance plan can support decisions like maintenance resource allocation. Or even, together with an optimisation technique, to provide an optimal selection of repair actions within its life cycle given an objective.

1.1 Challenges in Asset Maintenance Modelling

Managing infrastructure assets has become increasingly critical in recent years. For example, more and more bridges are deteriorated into poor condition recently due to the increase of structure age, traffic volume and more severe environmental status [94]. As a result, the corresponding maintenance cost is considerable, so as the risk of asset failure. In order to better schedule the maintenance resources, a range of different modelling techniques has been proposed to support the inspection and maintenance-related decisions making. However, in many situations, these techniques do not provide sound decision tools as they are based on impractical or simplified assumptions (see Section 2.4.4 and 3.6 below for more details). The challenges among the current practices can be summarised as:

- **Small datasets and poor-quality data** : When predicting asset deterioration, most traditional methods are data-driven (assuming to have sufficient deterioration data to learn from). However, for infrastructure assets such as bridges, the high cost of inspections means they are only performed once every few years and so, often, only limited amounts of deterioration data can be inferred [138]. Furthermore, this data usually comes with considerable uncertainty, for example, we may know deterioration happened between two consecutive inspections but are unable to decide exactly when it happened [101]. Currently, we lack a consistent way of incorporating this data.

- **Difficult to incorporate other information**: Engineers may have knowledge about the deterioration behaviours of different assets, like that a metal bridge generally deteriorates faster than a masonry bridge. But often, this kind of information is neglected in the modelling process due to the difficulty of eliciting this engineering knowledge from experts and, even when it can be elicited, incorporating that knowledge to learn the deterioration process is hard.

  Additionally, in practice, we may have many descriptive features, not only the asset material but also characteristics such as age, loading and many others. Presently, there lacks a way to select a list of important features that is indicative enough to represent the deterioration behaviour, and allows us to use them to separate assets into groups and quantify how similar the groups are. This is important because assets with similar characteristics may deteriorate in a similar way [23].

- **Reasoning within a complicated model framework**: Often, decisions recommended by many research studies rely on simplified modelling assumptions to promote model generalisability across different assets. For instance, the models only distinguish whether an asset is either working or has failed, while in practice, some assets may be rated with multiple states, where the deterioration rate between each state is not constant [60, 196]. Another example would be that the interaction between components in an asset is often simplified, but in reality, these components could interact quite differently [196, 99]. This is a common challenge for many traditional approaches: they implement the simplification to prevent their models from growing exponentially with the increase of problem complexity and becoming too complicated to manage [196]. However, the simplifications then mean it becomes impractical to apply them in some real-world applications.

In contrast, in many industrial applications, detailed models with practical assumptions have been developed. The problem is they are designed for specific assets only and therefore cannot be applied to other assets, such as those with different modelling configurations. Therefore, there is a discrepancy between approaches from research and industry which needs to be closed, achieving both practical modelling assumptions and flexibility.

1.2 Research Objectives

This thesis aims to develop a framework for asset maintenance management to address the above challenges. It plays a role as a preliminary analysis to guide the detail inspection activity and supports a variety of maintenance-related decisions in a manageable manner. To fulfil the aim to support asset inspection and maintenance decisions, this thesis considers the following research objectives:

I Develop a model that can learn asset deterioration behaviours from data and knowledge. Show how to encode data with uncertainty inferred from inspection history and how to derive engineering knowledge from experts that can be used in the model.

II Make use of the feature information to study the relationship between different asset groups. Use this to leverage the learning of asset deterioration between groups, even when there is little historical inspection record for some groups.

III Apply the deterioration model to predict the state of a given asset over time. Extend the deterioration prediction model to support more practical modelling assumptions, such as an extended description of asset states and the interaction between multiple components.

IV Use the developed deterioration prediction model in a real-world context, and compare its performance with other existing methods.

V Develop models that can reason and support a variety of inspection and maintenance-related decisions, and demonstrate the uses of them using real-world cases.

VI Show how the modelling choices could be effectively managed for maintenance modelling with various specifications.

1.3 Thesis Structure

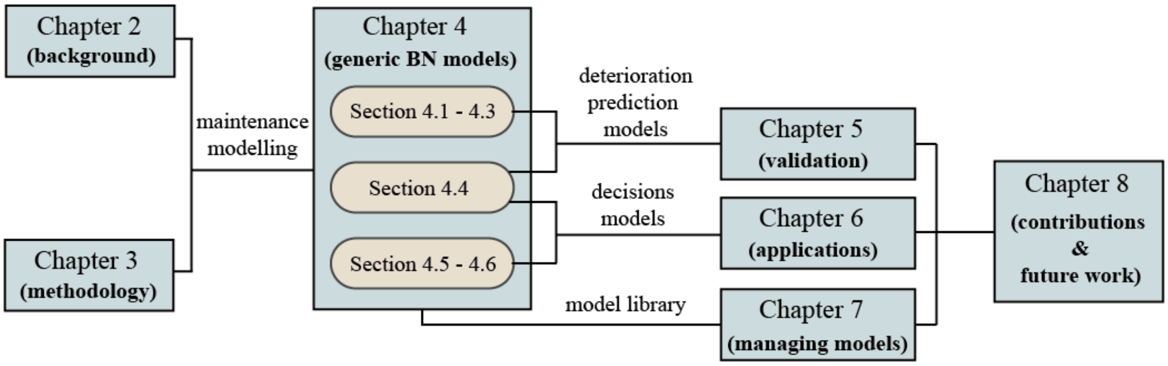


Figure 1.1 Thesis structure.

The thesis is structured as shown in Figure 1.1, where:

Chapter 2 gives an overview of the problem domain: how to model deterioration and reason related decisions. It describes the complexity in modelling asset deterioration and the variety of inspection and maintenance decisions. It also reviews the current practices for deterioration modelling and decisions support.

Chapter 3 introduces the methodology: Bayesian modelling. Three aspects of the Bayesian modelling are emphasised: i) learning parameters in a Bayesian statistical model; ii) Bayesian Networks (BNs) for decisions support; iii) how to extend them in the context of complex and large-scale system modelling. At the end of the chapter, it summaries the applications of the BNs in maintenance modelling and points out their current limitations, which are tackled in the rest of the thesis.

Chapter 4 begins the contributions of the thesis, showing a series of BNs for maintenance modelling. It covers how to learn parameters of a statistical distribution from data and knowledge. In particular, it shows how to encode censored data and how we can elicit knowledge from engineers. For assets with little deterioration data, it shows how to learn deterioration between similar assets. It also shows how to use the deterioration model to predict asset condition, and how to extend it with various assumptions. Models to support repair decisions and maintenance plans are also presented. It addresses Objectives I, II, III and partly V. This chapter leads to the validation in Chapter 5 and application in Chapter 6. The developed models are stored in the model library in Chapter 7.

Chapter 5 first uses synthesis data to validate the feasibility of applying parameter learning techniques in learning deterioration. It then uses a real case study to validate the performance of the proposed models in learning deterioration. After a background introduction of the case study, it shows how to select a subset of predictive features and use these features to group similar assets, and further, to build the BNs. The performance of the built models is measured and compared with other existing predictive techniques. This chapter addresses Objective IV.

Chapter 6 demonstrates the use of proposed methods in supporting a range of decisions raised from various maintenance-related problems. These decisions are based on real-world case studies, including asset condition prediction, reliability analysis for inspection decisions, repair decision evaluation, and maintenance planning. This chapter addresses Objective V.

Chapter 7 gives the vision to organise models proposed in Chapter 4 for maintenance problems with different specifications. A framework about how the modelling can be effectively encoded, and later managed by a decision maker who does not necessarily understand BNs, is presented. This chapter illustrates the advantage of the framework by a prototype. It shows how to build BN variants for different maintenance problems via manipulating a relational database. This chapter addresses Objective VI.

Chapter 8 summarises the contributions of the thesis toward the research objectives and discusses the future directions of this research.

1.4 Associated Publications

Part of the work in this thesis has been presented in peer-reviewed publications, as follows:

1. Zhang, H. and Marsh, D. W. R. (2016). Bayesian Network Models for Making Maintenance Decisions from Data and Expert Judgment. In *European Safety and Reliability Conference 2016 (ESREL 2016)*, pages 1056–1063. CRC Press, DOI: `10.1201/9781315374987` [This paper covers part of Chapter 4].

2. Zhang, H. and Marsh, D. W. R. (2018a). Generic Bayesian Network Models for Making Maintenance Decisions from Available Data and Expert Knowledge. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 232(5):505–523, DOI: `10.1177/1748006X17742765` [Extended from the conference paper, this paper covers part of Chapter 4 and part of Chapter 6].

3. Zhang, H. and Marsh, D. W. R. (2018b). Towards A Model-Based Asset Deterioration Framework Represented by Probabilistic Relational Models. In *European Safety and Reliability Conference 2018 (ESREL 2018)*, pages 671–679. Taylor & Francis Group, DOI: `10.1201/9781351174664-83` [This paper covers Chapter 7].

4. Zhang, H. and Marsh, D. W. R. Learning from Uncertain Data, Knowledge and Similar Groups: Individualised Multi-State Deterioration Prediction for Infrastructure Asset. *In Submission* [This paper primarily covers Chapter 5 and part of Chapter 4].

5. Zhang, H. and Marsh, D. W. R. Managing Infrastructure Asset: Bayesian-Based Models for Inspection and Maintenance Decisions Reasoning and Planning. *In Submission* [This paper primarily covers Chapter 6 and part of Chapter 4].

Chapter 2

Bridge Asset Management

Figure 2.1 Partial collapse of the Morandi bridge in Genoa, Italy on 14 August 2018 (this image is taken from [13]).

The sudden collapse of the Morandi bridge on 14 August 2018 (Figure 2.1) tragically took away 43 lives. Accidents like this add to the concern of maintainers that many bridges are in poor condition and they may be unaware which need most urgent attention. As a result, there is a need for an accurate assessment to identify bridges that could transition to a critical condition in the near future. This change of the condition of an asset, for example, from a good condition to a critical condition, is referred to as deterioration. The Morandi bridge deteriorated faster than was expected. Experts suspect the environmental factors played a role in accelerating this process in this case: the bridge is close to the sea and located in an industrial area, and sea salt, humidity and local pollutants can induce corrosion of some bridge components accelerating deterioration. Meanwhile, other factors such as possible design flaws, the heavy traffic, and the rainfall that took place during that day may also have contributed [13]. The joint influence of many factors makes the identification of assets in a critical condition as a result of deterioration very difficult.

Commonly, regular inspections are made to monitor the condition of an asset, so that if the asset deteriorates into an unacceptable condition, appropriate repairs are carried out to improve its condition. Media reports have claimed that regular inspection and repair work on the Morandi bridge was completed. However, this raises a number of questions including a) whether the frequency of inspections was suitable for this bridge given its recent condition and its rate of deterioration and b) were the repairs sufficient to ensure the bridge would operate safely until its next inspection? These questions are of concern to decision-makers in the field of maintenance. In order to reduce the risk of having accidents such as the collapse of the Morandi bridge, we would like to be able to provide an accurate deterioration prediction and use it to make suitable decisions about maintenance.

This chapter introduces the background to this challenge of managing bridge assets inspection and maintenance activities. Though the focus on this thesis is bridge assets, we believe our methodology is generic that could be applied to many types of infrastructure assets such as roads and railways. Section 2.1 introduces the concepts and current practices for bridge asset maintenance. Section 2.2 focuses on the data that is available to learn and understand the rate of decay, it introduces the deterioration time data that can be inferred from inspection records and how to manage the causes of deterioration differences between assets. Section 2.3 focuses on the modelling of deterioration, it introduces what techniques were used to estimate the deterioration rate, and what modelling assumptions are often made when applying these estimations in practice. Section 2.4 focuses on decisions support, it reviews the types of decisions and techniques usually used when making these decisions. A summary is given in Section 2.5.

2.1 Introduction to Bridge Asset Maintenance

This section introduces the key concepts for bridge asset maintenance, including asset compositions in Section 2.1.1, asset condition in Section 2.1.2, deterioration and factors that may affect its rate in Section 2.1.3, and inspection and repair activities in Section 2.1.4. A summary is given in Section 2.1.5.

2.1.1 Bridge Components

Typically, a bridge can be considered as a large system that comprised of many components [173]. As a result, the maintenance of a bridge often refers to the repair of an individual component or a group of components.

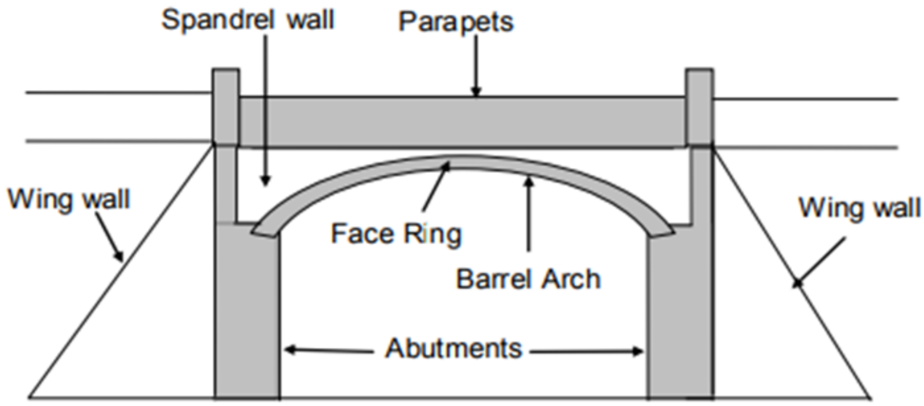


Figure 2.2 Components of a typical masonry arch bridge [144].

Figure 2.2 shows an example of a masonry arch bridge and its components described in Rafiq et al. [144]. This bridge is decomposed into six types of components. Considering the bridge as a system, these components are organised as its two subsystems: the deck subsystem and the support subsystem. A bridge deck refers to the surface of a bridge, it is used to carry the loads of the bridge (e.g. pedestrians or trains), in this example, it is made up of the parapets, the spandrel wall, the face ring and the barrel arch. In this example, the bridge support – the structure that holds up the deck – is made up of the wing wall and the abutments. In Great Britain (GB, the United Kingdom excluding Northern Ireland), Network Rail describes these subsystems (deck and support in the above example) as major elements of a bridge, and the components within each subsystem are described as minor elements [126]. This example is later used as a case study in Section 6.2.

2.1.2 Bridge Asset Condition

A means to judge the condition of an asset or its component is needed, with the overall aim of providing an indication of whether the asset or component is likely to fulfil its function. A common approach is to use a grading system, which distinguishes a fixed number of condition states. The intention is that the worse condition grades correspond to increased deterioration and therefore a greater risk of failure. However, since the evaluation of the condition may only look at visible defects, the relationship between the condition and the actual state of the asset may be complex. We describe two examples of these grading systems, one from the GB and the other from the United States of America (USA).

For bridges managed by Network Rail in GB, the condition of a bridge component is recorded when performing an inspection using a marking index called Structures Condition Marking Index (SCMI)[126](recently renamed to the Bridge Condition Marking Index (BCMI)). A BCMI score ranges from 0 to 100, where 0 represents a component in extremely poor condition, and 100 presents the element in perfect condition. Generally, a BCMI score above 80 is considered to be in good state and below 45 is in poor condition. For example, Rafiq et al. [144] modelled the condition of each component of a bridge as a three-state variable, namely, poor (BCMI range from 0 to 45), fair (BCMI range from 46 to 80) and good (BCMI range from 81 to 100) [144].

Table 2.1 Description of the grading system for bridge component in the NBI database [182]

Rate	**Condition**	**Description**
9	Excellent	-
8	Very good	No problems noted
7	Good	Some minor problems
6	Satisfactory	Structural elements show some minor deterioration
5	Fair	All primary structural elements are sound but may have minor section loss, cracking, spalling or scour
4	Poor	Advanced section loss, deterioration, spalling or scour
3	Serious	Loss of section, deterioration, spalling or scour have seriously affected primary structural components. Local failures are possible. Fatigue cracks in steel or shear cracks in concrete may be present
2	Critical	Advanced deterioration of primary structural elements. Fatigue cracks in steel or shear cracks in concrete may be present or scour may have removed substructure support. Unless closely monitored it may be necessary to close the bridge until corrective action is taken
1	Imminent failure	Major deterioration or section loss present in critical structural components or obvious vertical or horizontal movement affecting structure stability. Bridge is closed to traffic but corrective action may put back in light service
0	Failed	Out of service - beyond corrective action

Similarly, in the USA, the condition of a bridge component is also rated with a grading system. The health of a bridge is monitored through periodic bridge inspection (usually every two years) and recorded in a national database called National Bridge Inventory (NBI). Since the NBI database is publicly available and records a standard structure inspection, we use it in this thesis (Chapter 5 and 6). It archives unified information for over half a million bridges and tunnels in the US following the national inspection standard since 1992. In 2018, there are 616,096 bridges and 503 tunnels recorded in the NBI database. Recently, the Federal Highway Administration recategorized the NBI database so that from 2018, tunnels data has been separated out of the NBI database into a separated database called National Tunnel Inventory. To avoid future confusion, the phase 'NBI database' in this thesis refers to the bridge data only. The inspection evaluates the conditions of the bridge component deck, superstructure, substructure and culvert on a 9 to 0 scale with a one-point interval. In the grading system, 9 represents excellent condition and 0 represents the failed condition (see Table 2.1 for a description). In general, a condition rating of 4 or lower quantifies a bridge as structurally deficient. As a result, this bridge may require speed or load restriction to ensure safety.

These two examples describe the grading systems for condition evaluation of asset components. To evaluate the overall condition of the bridge, we can aggregate the component conditions. For example, in Rafiq et al. [144], each minor element (component) is given a weight representing its contribution to the overall bridge condition. This is done in two stages: the conditions of the minor elements are aggregated, proportional to their weights, to give the condition of major elements; these are then further aggregated proportionally to give the overall condition of the bridge. Another example is from the NBI: a bridge is considered as structurally deficient when any of its components are in a poor condition (condition rating of 4 or lower). The relationship between condition of components and condition of its asset as a whole vary between assets, this is further discussed in Section 2.3.3.

2.1.3 Deterioration Mechanisms

The deterioration process of an asset is the decrease of its functionality over time [133]. For example, the deterioration of a concrete structure can involve cracks. A crack usually starts from a point of stress concentration and propagates across the structure. The crack may expose the steel reinforcement to the air as shown in Figure 2.3, resulting in corrosion of the steel reinforcement. Expansion of the corrosion products (i.e. iron oxides) of the steel reinforcement may form further cracks in the concrete structure. Once the cracks reach a critical size, the structure fractures and separates into parts. Damages like cracks, with visible signs of deterioration, are usually easy to identify during a regular inspection.

Ryan et al. [146] has provided a comprehensive list of deterioration signs to help engineers identify the deterioration of some bridge components. But for hidden components that cannot be observed directly, their deterioration cannot be inspected from visual inspections. For example, to evaluate the deterioration of post-tensioning tendons, we need to perform expensive inspection like an intrusive inspection, such as drilling holes, so that we can determine the deterioration level inside the structure [96]. To schedule these inspection activities, we need to predict how long an asset or a component is likely to take to deteriorate to an unsafe condition.

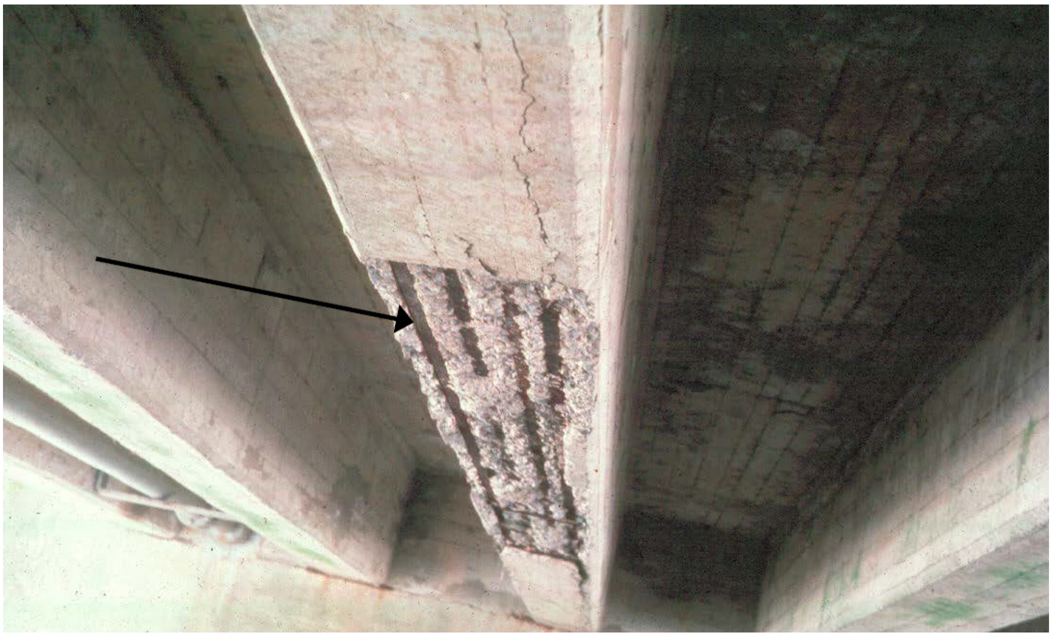


Figure 2.3 Concrete structure with cracks and tensile steel reinforcement showing [146].

Since the condition of an asset is assessed using a fixed number of condition states, the deterioration of the asset can be modelled by the rate its condition transitions from its existing state to a worse state. To estimate this rate of deterioration, we need to understand the causes of the differences between deterioration rates. For example, age plays an important role in influencing the rate of deterioration: an older bridge may deteriorate more quickly than one that is less old. Another example is the loading of the structure: a bridge with a higher traffic volume may deteriorate more rapidly than a less loaded bridge. There are many factors associated with an asset that potentially can impact its deterioration rate. For example, in the NBI database, the recorded factors range from year of built, maintenance agency, deck structure type, structure length to average daily traffic. In total, there are around a hundred features recorded for each bridge in that database. This leaves us a question about which factors are more influential on the deterioration rate, and the levels of their impacts; we address this question in Section 2.2.2.

2.1.4 Inspection Regime and Repair Policy

An on-site periodic inspection is a traditional way to collect information about asset condition, with a fixed interval between inspections [174]. For example, the maintenance regime operated by Network Rail has two types of bridge inspections: one is the annual visual examination that looks for changes in the condition of the structure as a whole since the last examination; the other one is the detailed examination, which is performed every six years to look at all parts of the structure to determine their conditions and the extent of their deterioration. In the US, bridge inspection is performed with a biennial regime. To reduce the risk of having accidents like the collapse of the Morandi bridge, we either need to increase the frequency of inspection - that means a higher cost; or predict the deterioration of the asset over time, so that we can schedule an inspection at an appropriate time for each asset. Later Section 2.4.1 reviews how these decisions are made.

The repair cost generally increases as an asset deteriorates, therefore, to reduce repair costs while keeping the asset safe, it is important to implement a suitable repair policy. Preventive maintenance performs repairs at a fixed time interval or age (time-based maintenance), regardless of the condition of the asset. This is a common policy, but it is more suitable for an asset that deteriorates at a constant rate, is less critical to safety and reliability, or has a lower maintenance cost [166]. In contrast, predictive maintenance recommends repair action based on the condition of the asset estimated from its deterioration (condition-based maintenance). This policy is more suitable for an infrastructure asset as its deterioration is often more uncertain, it is more safety critical, and its maintenance is often more expensive.

A maintenance policy should explain how to choose the best repair action given the current condition of the asset. Some repair actions can only be applied to resolve specific defects of a structure. For example, filling can only be used to structure with a sign of cracking. But in some cases, there is more than one repair option available to improve the performance of an asset. For example, for bridge deck with cracks, repair options include an epoxy overlay, filling cracks and patching.

Further, the effectiveness of different actions varies. Some actions can restore the asset's performance to as good as new, but others can only improve it slightly. Michigan Department of Transportation [116] give several examples: comparing an epoxy overlay, which can restore the condition of a deck to good condition, to patching, which can only improve the deck condition slightly. Based on their effectiveness in restoring the asset's performance, we can classify repair actions into minimal, imperfect and perfect maintenance. Minimal maintenance, such as patching, usually fixes an asset temporally to prevent it from failing, but its effect usually does not last for a long time. Imperfect maintenance, such as an epoxy overlay, restores an asset's condition to a better state. Perfect maintenance, such as replacement, returns an asset back to a state that is as good as new.

From a short-term perspective, engineers would like to know which repair action to perform as well as the anticipated effectiveness of the repair action. This is further discussed in Section 2.4.2. From a long-term perspective, we would like to plan maintenance activities, considering multiple intervention cycles, that can schedule appropriate repair actions at suitable time satisfying an overall objective, for example, minimal cost while guaranteeing a good level of reliability. This is discussed in Section 2.4.3.

2.1.5 Summary

A bridge asset comprises many components, and its condition - an indication of the performance of the asset - can be evaluated from the condition of its components. The information about the condition of an asset and its component is usually used to make condition-based decisions for both inspection and maintenance activities.

Commonly, inspection for a bridge is performed with a fixed time interval, but this regime is often less cost-effective since it neglects the variation in the rate of deterioration between different asset. When an asset is deteriorating faster, we should inspect it more frequently than is necessary for an asset that deteriorates more slowly. Similarly, taking repair action based on the condition of an asset rather than using time-based maintenance is more suited for infrastructure asset. With the deterioration prediction, we can, therefore, suggest relevant decisions for asset maintenance based on its condition.

We can estimate the rate of deterioration to predict the condition of an asset. The deterioration rate between states describes how fast the transition is and it is influenced by many factors. After a discussion of the information that can be used to learn deterioration and how to manage the factors that have impacts on deterioration in Section 2.2, we review the common techniques were used in the literature in Section 2.3. But applying these techniques to predict an asset's condition can be a complicated task - different assets may have different modelling assumptions, for example, asset maybe rated with multiple states while the transition probabilities between states are not constant, or asset with multiple components where its components may interact in various ways. This complexity is also discussed in Section 2.3. With the prediction of asset condition from deterioration, we can therefore make relevant decisions. In Section 2.4, we further review what type of decisions are usually analysed in this field and what techniques were used to suggest these decisions.

2.2 Data for Predicting Asset Deterioration

Maintainers often record their estimation of the asset's condition during each inspection, and an asset management database often contains information about the asset's features. If these inspection records and feature information are available, we can use them to infer the rate of an asset's deterioration. This section introduces inference from these inspection records and explains how to select factors that may an impact on the deterioration rate.

2.2.1 Deterioration Time Data

The majority of bridge inspection records are collected from traditional on-site inspections. These on-site inspections are often performed periodically. Generally, inferring a deterioration rate requires sufficient data derived from historical asset condition information made from inspection or maintenance [94].

The deterioration time (also called sojourn time or occupation time in some studies) of each state can be inferred from condition data and their corresponding inspection time. Since most bridges are inspected periodically, the inspection data may not reveal the actual time the structure transitioned from one condition to another: suppose, for example, that the most recent inspection shows a structure is in state 8 but was in state 9 at the previous inspection. This situation does not imply the deterioration time from state 9 to 8 is the time gap between these two inspections: the deterioration may happen anytime between these two inspections. This limitation in the information available is referred to as 'censorship', introducing uncertainty to the observed deterioration [101]. We know only that the structure deteriorated before an inspection (left censored), between two consecutive inspections (interval censored), or after the most recent inspection (right censored), rather than a specific time point.

Apart from the uncertainty in the actual time of condition changes, for some states, the number of inspection records is often small. Periodic inspection policy and slow deterioration process are two main reasons. Since an on-site inspection is conducted only every year or every few years, this activity usually only provides us with a limited number of records. For example, as described in Section 2.1.4, in GB, rail bridge visual inspection is every one year and detail inspection is every six years; while in the US, bridge inspection is performed every two years. These long inspection intervals can only infer a limited amount of deterioration data. Besides, since many infrastructure assets, such as bridges, decay slowly, transitions from one state to another do not occur often. For example, in the 26 years of inspections recorded in the NBI database, some bridges have not even deteriorated once. Finally, most of the bridges are in good or fair conditions (at state 5 or over) because maintainers often perform interventions to prevent further deterioration when assets are in poor condition. As of 2017, only 7.7% of the bridges are at state 4 or less that could produce deterioration data for lower states.

2.2.2 Features Influencing Deterioration

As discussed in Section 2.1.3, there are many features of an asset or its use that can influence its deterioration rate. It is possible to give a more accurate prediction of the condition of an individual asset by considering the specific values of the features that impact the rate of deterioration [177].

Individual structures with same features may share the same deterioration characteristics, deteriorating at a similar rate. We can use this assumption to separate the overall population into different groups, so that the deterioration rate is estimated within each group. However, each structure is often associated with many features. For example, each bridge has over a hundred features recorded in the NBI database. Considering too many features can be a disadvantage as it takes too many resources and results in slow computations. More importantly, after reaching an optimal number of features, increasing the number of features further causes a decrease in accuracy [80], for example, by overfitting. Therefore, to provide individualised deterioration predictions, one of the challenges is to reduce the feature dimension into a small subset. With a few predictive features, we can separate structures into groups so that, within each group, structures are assumed to deteriorate similarly.

A framework to give individualised deterioration predictions in this way was developed in Chang [23]. It first reduced the feature dimension of the dataset to select a small subset of features as important features, and then grouped assets with the same feature values. Each group was modelled using a Markov model to model multi-state deterioration, the transition probabilities of the Markov model were learned using logistic regression. Individualised deterioration prediction was then performed based on the group the asset belongs to. In Chapter 5, we compare this approach with our methods.

To reduce the dimension, Chang [23] first conducted a covariance analysis to remove highly correlated variables and later used a penalised regression to rank features based on their importance in deciding the deterioration time. Feature selection has been intensively studied in the feature engineering domain. Unlike feature extraction methods, such as Principal Component Analysis, that synthesise new features which may not have intuitive meaning, feature selection is more suited to the problems addressed in our work since it finds a subset of features from the candidate pool. These features are more meaningful than synthesised features, so can be interpreted by engineers.

Based on the assumption that we can remove many less relevant features without losing too much information on the response variable (e.g. the deterioration time), feature selection aims to reduce the set of features to a manageable size that is still representative enough to reflect the variability of behaviours. Plenty of techniques have been developed for feature selection, of which, filter and wrapper methods are the two most developed areas.

Filter method applies a statistical measurement to evaluate the feature space. Because of its computational efficiency, it is usually used as a preprocessing technique. Chang et al. [24] measured the correlation between variables in the NBI database. Strongly correlated variables (with a correlation over 0.90 or smaller than -0.90 as suggested by Ayyub and McCuen [10]), considered to imply redundant information, were identified. Among those highly correlated variables, they only kept the features with the highest correlations with the response variable and eliminated the rest. This method can slightly reduce the feature space size while maintaining the predicting power of the features. However, we still need further dimension reduction, because, among the remaining features, many of them do not have strong correlations with the response variable.

Wrapper methods evaluate a subset of features, where different combinations are generated, evaluated and compared with other combinations. A learning model is used to estimate a subset of features and assign a grade to each combination based on a given model metric. Various studies have been proposed to study the feature importance [181], of which, random forest is one of the most popular methods. Unlike penalised regression applied in Chang [23], random forest does not assume a linear relationship between the features and the response variable (e.g. deterioration time). Random forest is an ensemble method used to performs variable importance evaluation by weighting multiple decision trees independently developed from bagged training samples.

In the random forest, each node in the decision trees is a value of a single feature that is designed to split its descendent nodes, so eventually similar response variables end up in the same pool. Two importance measurement functions of the features proposed by Breiman [20] are commonly used in the random forest: Mean Decrease Impurity (MDI) and Mean Decrease Accuracy (MDA). MDI searches to spilt which variable would give the lowest purity (for example, Gini impurity) given the response variable. Within a tree, each feature is evaluated by how much changing the value of the feature would decrease the weighted impurity of the tree. The impurity decrease from each feature are thus averaged in the forest, and the features are ranked accordingly: the lower the decrease in the impurity the better. MDA measures the loss of accuracy. By randomly permuting the value of a single feature that matches the distribution of the samples, this algorithm computes the accuracy of the tree given the response variable. By repeating this process for each feature, the mean loss of accuracy of each feature in the whole forest is obtained and used to rank the features accordingly.

Strobl et al. [165] made a comparison between these two measurements. It reveals that MDI measurement is biased towards features with more categories. This bias is a challenging problem for the NBI dataset because it involves mixed data types, where some are binary (e.g. whether the structure is flared) and some have more than 20 categories (e.g. feature maintenance responsibility has 29 categories). Though Strobl et al. [164] later claimed that MDA measurement is also biased as it favours features highly correlated to other variables. To avoid this, when applying the MDA measurement, we can first preprocess the feature space to remove variables that are highly correlated with any of the features.

Though Chang [23] provides us with a framework to perform individualised prediction, the challenges of uncertainty in the deterioration data and learning the deterioration rate for those asset groups with limited data amount remain untouched. Since it is possible to separate assets into different groups by their feature values, we could use what we have learned about large groups to help to learn the deterioration rate of asset groups with limited data. Later in Chapter 4 we develop a model based on this idea and validate it in Chapter 5. Section 5.3.1 gives more details about how we perform feature engineering in our case study.

2.3 Techniques for Modelling Deterioration

With the data from Section 2.2, we can predict asset condition by estimating the rate of its deterioration. In this section, we first review a list of deterioration rate estimation methodologies in Section 2.3.1. We classified these methodologies into two types: point estimation techniques that predict deterioration using single-valued parameter(s), and statistical technique that describes deterioration with a probability distribution.

As explained in Section 2.1.2, the condition of an asset is usually expressed using a grading system of multiple states, for example describing condition of bridge component using index from 0 to 100 in Network Rail managed bridges and condition of bridge structures are rated from state 0 to state 9 in NBI database. The techniques described in Section 2.3.1 cover only the binary case, of working or failed. We now describe how the techniques have been extended to the more general case of multiple condition states in Section 2.3.2.

We also consider in more detail the problem already described in Section 2.1.2 of distinguishing between the states of components and the overall state of an asset. In Section 2.1.2, two simple configuration examples were described; here in Section 2.3.3, we cover more explanation about how they would be modelled that draw on standard reliability techniques for analysing systems with various configurations.

2.3.1 Estimation of Deterioration Rate

A range of techniques has been proposed to estimate the rate of asset deterioration, in this section, we distribute them into two types: a) point estimation techniques to produce the best estimate of an unknown population parameter to predict deterioration and b) techniques that describe deterioration with a statistical distribution.

Point Estimation Technique

A time-dependent stochastic process $\{X(t), t \geq 0\}$, where $X(t)$ is a collection of random variables for all $t \geq 0$, is one primary form when modelling deterioration [50]. The variable X(t) gives the state of the system – for example, working or broken – at time t. Among the stochastic processes, most studies assume deterioration to be a Markov process [12]. Markov process is a stochastic process where the future state only depends on the present state (this Markov property is very useful for modelling of multi-state asset deterioration, it is further emphasised in the next section). In deterioration modelling, the transition from one state to another state is represented by the transition probability. Followed the Markov property, the conditional probability of moving into future state S_{t+1} at time $t+1$ given the present state S_t at time t is:

$$P(X_{t+1} = S_{t+1} | X_t = S_t, \ldots, X_1 = S_1, X_0 = S_0) = P(X_{t+1} = S_{t+1} | X_t = S_t) \qquad (2.1)$$

Extensions of the Markov process, such as Brownian motion with independent normal-distributed increments and decrements, and the gamma process with independent gamma-distributed increments are also widely researched in deterioration modelling. These extensions are further discussed and compared in Van Noortwijk and Pandey [172], Frangopol et al. [50] and Gorjian et al. [58].

Point estimation is often used in these studies to estimate the transition probabilities of stochastic processes. Point estimation produces a single value that can best represent the population parameter. A range of methods can be used to fit the transition probability, such as maximum likelihood estimation for Markov process in Kallen and Van Noortwijk [75] and method of moments for gamma process in Ohadi and Micic [130].

Meanwhile, with the development of machine learning techniques in recent years, several approaches have been employed as alternatives to the stochastic process to model and predict asset deterioration with point estimators. Instead of estimating the transition probabilities like what stochastic processes do, these techniques describe the deterioration rate as an arbitrary linear or nonlinear function. For example, Winn and Burgueño [184] built an Artificial Neural Networks model for bridge deck condition prediction, which was further improved by an Ensemble Neural Networks. Similarly, Fink et al. [49] adopted a Multilayer Feedforward Neural Networks based on Multi-Valued Neurons to provide time-series aggregated predictions for railway turnout system's deterioration. These two examples describe the deterioration process in the form of a linear regression and a nonlinear regression respectively, and output the best-estimated values of deterioration given the age of assets.

Though these studies have shown remarkable performance for aggregated prediction of the global population - for example, how many assets will fail in the next 10 years; they also agreed that current approaches from point estimators suffer difficulty to provide an accurate prediction for individuals - for example, for two assets with the same age, which asset is more likely to fail. Also, in these studies, deterioration time was assumed to be precise (complete data, describing exactly how long does it takes for an asset to deteriorate) and the data size was big, whereas in practice, as discussed in Section 2.2.1, data are often censored, and for some specific type of assets, the amount of data is relatively small. Some studies had tackled the limitation of the current approaches applied in this field when dealing with data uncertainty. For example, Ferguson et al. [47] modelled the uncertain data as censored survival time within a Markov model. To evaluate the transition probabilities, they employed a non-parametric estimator called Datta-Satten estimator for hazard rate function estimation. This methodology has been applied many fields, for example, in predicting the deterioration of patient's clinical states [31], but not in reliability. Later in Section 5.4.3, we compare the performance of this method with our proposed methods. However, most other methods are still data-driven approaches and did not take deterioration uncertainty between different individuals into consideration.

Statistical Distribution Technique

Unlike point estimation technique, which describes the deterioration probability from one state to another state with a single value, a parametric statistical distribution expresses its prediction with a stochastic function that takes the uncertainty during the deterioration process into consideration [157]. With a statistical distribution, we can describe the deterioration with, for example, an interval estimation. An interval estimator consists of a range of plausible values; we can represent them in the form of confidence or credible intervals. These statistical techniques are also often applied to deterioration prediction [50]. In a lifetime distribution, the expected lifetime of an asset is derived from its likely time to failure. A set of failure time data is gathered and fitted to a statistical distribution, describing the failure probability of an asset at a given time. The lifetime distribution function (denoted as F) is:

$$F(T) = Pr(T \leqslant t) \tag{2.2}$$

where t is the time point, Pr denotes the probability and T is a random variable stands for the time to failure. This means the lifetime distribution function is the probability that the time to failure T is greater or equals to a specified time point t.

Therefore, the failure rate (denoted as λ), representing the probability of the asset has been working for a time t, but will fail with an additional time dt is:

$$\lambda(t) = \lim_{dt \to 0} \frac{Pr(t \leqslant T < t + dt)}{dt \cdot Pr(T > t)} \tag{2.3}$$

Various statistical distributions have been used to fit asset deterioration behaviour. For example, He et al. [67] and Guler et al. [63] use exponential distributions to estimate the deterioration of the railway track. Studies of bridges in Agrawal et al. [3] and railways track in Andrews [5] provide two examples showing the use of Weibull distributions to model a range of asset deterioration behaviours.

In Le [94], lifetime data of bridge components of the same type and material are grouped to fit with a series of distributions. The fit of different distributions including normal, exponential, lognormal and Weibull distributions are compared using Anderson-Darling tests. Weibull distributions have the closest fit in most cases. Given this result and its versatility in describing a range of deterioration behaviours, the Weibull distribution is adopted in this thesis.

Apart from the advantage of allowing us to represent prediction in the form of an interval estimator, another advantage of using a statistical distribution is the interpretability over its parameters. For example, in a Weibull distribution, its probability density function (pdf) over time t is:

$$f(t) = \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1} e^{-(\frac{t}{\eta})^{\beta}}, f(t) \geq 0, \beta \geq 0, \eta \geq 0 \tag{2.4}$$

and is characterised by parameter shape β and parameter scale η. For example, with a shape value that is lower than 1, we can model a failure rate that decreases with time while a shape greater than 1 describes wear-out, giving an increasing failure rate. For a given shape, increasing the scale increases the mean failure time [73]. The interpretability of the parameters gives a natural way to extract knowledge from engineers, which may reduce the need for data. A more detailed explaining about how to elicit this knowledge will be disused in Section 4.1.3.

2.3.2 Applying Deterioration Estimation with Multiple Conditions

The techniques described in Section 2.3.1 cover only the binary case, of working or failed. We now describe how the techniques have been extended to the more general case of multiple condition states. For an asset that is rated with multiple states from $1,\ldots,k$, state k is the perfect state and state 1 is the worst state. We can define the probability of structure i being in state j at time t as $p_{i,j}(t)$. Since the states are mutually exclusive, the probability distribution of the state of structure i at time t can be represented by a complete collection of its states:

$$\sum_{j=1}^{k} p_{i,j}(t) = 1 \tag{2.5}$$

Deterioration with multiple states can be modelled using a Markov chain by a sequence of states representing the condition of an asset system over time (e.g. condition rating from 0 to 7 in Agrawal et al. [3]). This use of Markov models has been studied by, for example, Cesare et al. [22] and Jiang et al. [74], who applied Markov models to predict the state distribution of bridges, and Shafahi and Hakhamaneshi [152], who used it to predict track condition rating with multiple states. These Markov models maintain the Markov property (see Equation 2.1), where the probability of a state transition depends only on the current state.

To evaluate all possible transition probabilities from one state to another state during a transition time interval t, we can build an $S \times S$ (S represents the total number of possible states) one-step transition matrix as shown in Equation 2.6. An m-step transition matrix, representing the probabilities of moving from one state to another state in m transition time interval, can be calculated by multiplying of the matrix m times with itself [50].

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & \ldots & P_{1S} \\ P_{21} & P_{22} & P_{23} & \ldots & P_{2S} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ P_{S1} & P_{S2} & P_{S3} & \ldots & P_{SS} \end{bmatrix} \tag{2.6}$$

However, the probability of a state transition in a Markov model does not relate to the time an asset has been in the current state [94]: it implies all transitions between states occur at a constant failure rate, leading to exponentially distributed failure rates. This assumption is not suitable to describe a deterioration phase (e.g. the wear-out process) where we have an increasing deterioration rate [19]. We can overcome this limitation with a semi-Markov model that assumes the residence time in a state between transitions follows a specified distribution. A semi-Markov process is a stochastic process follows two independent random processes which evolves over time.

In a n states asset, the transition time from S_i to S_j is denoted as $T_{i\to j}$, with a probability density function $f_{i\to j}(t)$ and survival function $S_{i\to j}(t)$. Since asset deterioration is incremental (without intervention, asset condition gets worse over time), $i \leqslant j$, and $i = 1,2,\ldots,n-1$ and $j = 2,\ldots,n$, from Equation 2.1 we have:

$$P(X_{t+1} = S_{t+1}|X_t = S_t) = p_{i\to j}(t) = \frac{f_{i\to j}(t)}{S_{i\to j}(t) - S_{i-1\to i}(t)} \tag{2.7}$$

This equation can be used to calculate the time-dependent transition probability from state i to j, and to form the transition probability matrix for a semi-Markov process from Equation 2.6. The use of a distribution function in the form of a semi-Markov process is illustrated in Kleiner [79]. The semi-Markov process is applied to a deterioration model, with each transition time following a different Weibull distribution. This enables the modelling of an asset with multiple states where each transition between states follows a different distribution. Droguett et al. [39] gave another example of multi-state system failure, where transitions between successive states are modelled using a semi-Markov process. They modelled several distributions, including exponential, Weibull and lognormal distribution and discussed the difference between their deterioration features.

2.3.3 Condition of a Multi-Component System

As well as wanting to predict the future condition of an asset as a single unit, we also wish to estimate the condition of an asset from that of its components. We refer to the arrange of the components of an asset with multiple components as its configuration, with various type of configurations depending on the interaction between the components. We can classify the configurations into a parallel system configuration, a series system configuration, and a bridge structure (a non-parallel or series system) configuration. In Lisnianski et al. [99], such system with multiple components where additional, the component's performance can be described using a scale with multiple states (as described in Section 2.3.2), is referred to as a Multi-State System (MSS). The notations of an MSS are:

- A system S has n components: $S = \{E_1, E_2, \ldots, E_n\}$.
- All components and the corresponding system have k states {k,..., 1}, where $k \geq 1$. The states are ordered, where k represents the perfect condition, and 1 represents the worst.
- $T_i(u)$ represents the lifetime distribution of component i in the subset state {u, ..., k}.

A system with a parallel configuration fails only if all components failed, and system with a series configuration fails when one of the components within the system failed. Parallel structure assumes the capability of the component is replaceable by another component of the same structure. Therefore, the performance of this asset depends on one available component. A series structure suggests the system performs step by step along a line of components, therefore, its performance depends on all components being available.

The lifetime of parallel structure $T_{parallel}(u)$ and series structure $T_{series}(u)$ in the subset state {u, ..., k} are given by

$$\begin{aligned} T_{parallel}(u) &= \max_{1 \le i \le n} \{T_i(u)\}, u = 1, \ldots, k \\ T_{series}(u) &= \min_{1 \le i \le n} \{T_i(u)\}, u = 1, \ldots, k \end{aligned} \tag{2.8}$$

which indicates that parallel MSS stays in conditions {u, ..., k} only if at least one of the components is among these states, and the series MSS stays in conditions {u, ..., k} only if all the components are among these states. For example, in the NBI database, a bridge is considered as in a poor condition when any of its components are in a poor condition. This can be modelled as a series MSS, meaning the lifetime of the bridge is determined by the minimum lifetime among its components.

Furthermore, an MSS may consist of a combination of parallel and series systems. We can treat it as a system with subsystems, while within each subsystem, components are arranged either in parallel or in series. Extensions to this have been proposed, for example, an m-out-of-n MSS, where at least m out of n components are functioning, and a standby system, where part of the parallel subsystem is not active all the time. Tillman et al. [168], Kuo and Prasad [87] and Kolowrocki [83] have provided us a detail introduction on these extensions.

However, some assets cannot be decomposed into a parallel or a series system since their components share the total demand or load of the system. Stoll and Garver [163] presented a typical example, where a power system fails if the full capacity of the interconnected power plants cannot meet the demand.

This kind of non-series or parallel configuration is called a bridge structure, where elements of the bridge structure are arranged spatially to share the load (demand), which implies the components are not statistically independent. Figure 2.4 shows a bridge structure example: E1 to E4 are components with the same functionality but independent of one another (e.g. abutments of a bridge), E5 is a critical component in determining the strength of the structure (e.g. a deck of a bridge) that connecting the remaining components. This represents a situation that components contribute differently to the objective function (e.g. the strength of a bridge).

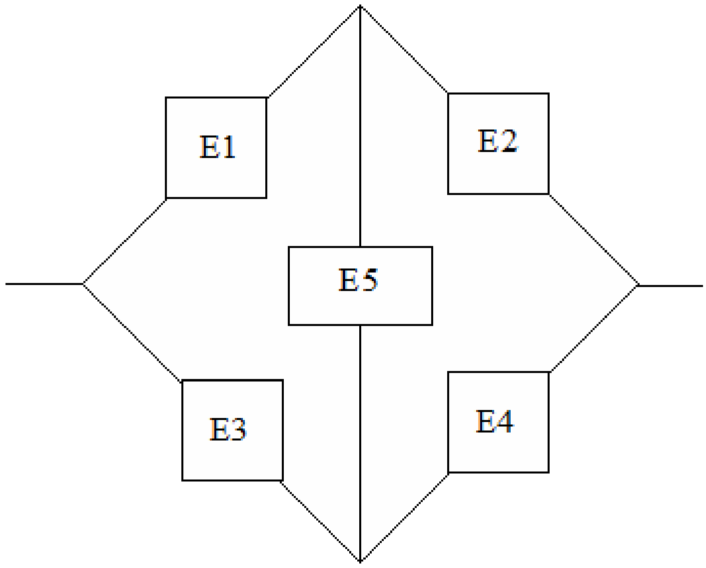


Figure 2.4 A bridge structure system configuration.

E5 is more relevant to the system performance: changing its state without changing the remaining components' states can still largely change the system functionality. Component relevancy can be therefore introduced in this type of configuration to assess the system reliability. The relevancy of a component is an associated contribution to the system performance, rather than the internal property of each component [99]. The masonry arch bridge mentioned earlier in Figure 2.2 presented in Section 2.1.1 is a typical example of a system with bridge structure configuration. Rafiq et al. [144] evaluated the condition of this bridge by its two major elements: the deck and the support. And each major element was further assessed by the conditions of its minor elements. Experts assigned weights to each element, and each weight represents the relevancy of the element in determining the deterioration of the bridge as a whole.

2.4 Making Maintenance Decisions from Asset Reliability

As an asset deteriorates, its condition gets worse, therefore, the reliability of this asset decreases. Asset reliability is defined as its ability to operate within a level of performance within a specified time [99], it is one of the most important criteria when engineers making maintenance decisions. This section describes decisions draw on asset reliability and techniques for analysing asset reliability to support these decisions making.

When managing infrastructure asset, we need to make decisions such as when to inspect which asset. We would like to use the deterioration model to predict the condition of an asset at a given time in the future to support this decision. Given a constraint on the acceptable condition, that is acceptable confidence that the deterioration is not too far advance, this prediction can guide inspection decisions such as which asset to inspect most urgently and how often the inspection is needed. This is discussed in Section 2.4.1.

When an inspection suggests a repair is needed, we have to decide which of the different repair actions to choose to maintain the asset in an acceptable condition. However, the effectiveness of different actions varies, as does the cost. To select an optimal repair decision, not only we need to consider the effect of the maintenance in restoring the condition of the asset, but also the associated cost. This is discussed in Section 2.4.2.

By understanding the rate of deterioration of an asset and the effectiveness of the available repair actions, we can plan an asset future maintenance work considering multiple intervention cycles. The challenge of maintenance planning is to use appropriate maintenance actions at suitable timing cost-effectively while satisfying a set of criteria. Section 2.4.3 discusses how these maintenance plans can be made.

To end, in Section 2.4.4, we discuss the common analysis techniques used in maintenance reliability that help engineers make these decisions.

2.4.1 Inspection Decision

The condition of an asset reflects the risk of it being at a critical state that requires intervention. As the probability of reaching an unacceptable condition rises the risk increase; we can set a threshold on the risk as the maximum allowable probability of an unacceptable condition. For example, bridges in Great Britain that are in Poor condition need major repairs, and in Very Poor condition require replacements [94] while bridges in the US at state under 5 (structurally deficient) require immediate repair [182].

After the estimation of deterioration rate, we can reason about the condition distribution of an asset given a time; this condition can be used as an indication about how reliable the asset will be. A reliability function is used to measure the probability the system did not fail before moment t. For a binary-state system, the reliability function is the survival function. T represents the time to failure; its probability density function denotes as $f(t)$, and its cumulative distribution function denotes as $F(t)$:

$$R(t) = P(T \geq t) = \int_t^\infty f(x)\,dx = 1 - F(t) \tag{2.9}$$

For a repairable system, we can characterise it by the Mean Time To Failure (MTTF) and for a non-repairable system, we can characterise it by the Mean Time Between Failure (MTBF):

$$\int_0^\infty R(t)\,dt = \int_0^\infty t f(t)\,dt \tag{2.10}$$

For example, the MTTF of a Weibull distribution μ (Equation 2.4) is:

$$\mu = \int_0^\infty t f(t)\,dt = \int_0^\infty t\frac{\beta}{\eta}(\frac{t}{\eta})^{\beta-1}e^{-(\frac{t}{\eta})^\beta}dt = \eta\Gamma\left(1+\frac{1}{\beta}\right) \tag{2.11}$$

For an asset with multiple states rated from $\{k,\dots,1\}$, where state k is the perfect state and state 1 is the worst state as defined in Section 2.3.2. Let $T(u)$ represents the time to transit to state after u. At moment t, a structure's reliability is its probability stays in state $\{k,\dots,u\}$, represented as $R(t,u)$. Conversely, the structure's unreliability (falling into an unacceptable state) is its probability of staying in state $\{1,\dots,u\}$, represented as $F(t,u)$:

$$\begin{aligned} R(t,u) &= P(T(u) \geq t) \\ F(t,u) &= P(T(u) < t) \end{aligned} \tag{2.12}$$

We can have the MTTF, representing the mean time of the system transit to the subset of unacceptable states $\{1,\dots,u\}$, where $E\cdot$ represents the expectation of the pdf: E{T(u)}

For an asset with multiple components, from Lisnianski et al. [99], we have the reliability of asset in parallel and in series configurations:

$$\begin{aligned} R_{parallel}(t,u) &= 1 - F_1(t,u) \cap F_2(t,u) \cap \dots \cap F_n(t,u) \\ R_{series}(t,u) &= R_1(t,u) \cap R_2(t,u) \cap \dots \cap R_n(t,u) \end{aligned} \tag{2.13}$$

In the case where each element's reliability is mutually exclusive (each component is independent from other components), we have the simplified version of Equation 2.13:

$$\begin{aligned} R_{parallel}(t,u) &= 1 - \prod_{i=1}^{n} F_i(t,u) \\ R_{series}(t,u) &= \prod_{i=1}^{n} R_i(t,u) \end{aligned} \tag{2.14}$$

For a bridge structure configuration (non-parallel or series system), the relevance of an element is its contribution to the overall system performance, rather than an intrinsic property of each element [99]. Therefore, it is defined in terms of the reliability performance of each element. Denoting the relevance (weight) of element i as w_i, we have:

$$R_{non}(t,\ u) \ = \ F(R_1(t,u),\ ...,\ R_n(t,u),\ w_1,\ ...,\ w_n) \tag{2.15}$$

Given a prediction about an asset's future condition, we can determine the reliability of the asset. The reliability can be used to suggest inspection decisions, including finding a suitable asset for inspection and finding a suitable time to inspect.

We can prioritise assets for inspection based on their risks of deteriorating to an unacceptable level. A simple way to do this is to rank the risk associated with each asset according to the most recent inspection result. For example, a bridge in poor condition at its latest inspection may require a more urgent intervention than a bridge showed no sign of deterioration. However, this may not always be true since it neglects the future deterioration of the asset, and therefore can lead to inaccurate risk ranking. For example, Stewart [161] gave an example of a bridge that has no sign of deterioration turned out to have a higher risk of failure in the next five years than a bridge already in a poor condition. He explained that although the bridge shows no sign of deterioration in the most recent inspection, it had a higher traffic volume than the other bridge, leading to a faster deterioration rate. The use of deterioration prediction can mitigate this by considering the further deterioration of an asset.

Stewart [161] suggested using a time-dependent reliability analysis. Structural failure is likely to occur when the statistical distribution of load exceeds the distribution of resilience (the ability to cope with the exceeded loading). In his work, the result of the prediction is a probability distribution rather than a point estimate from the most-possible prediction, we can measure the risks by the reliability for a given state threshold. The threshold is not necessarily equal to the time of a complete failure. Stewart and Rosowsky [162] have suggested that for inspection planning purpose, it is more meaningful to calculate the reliability based on the time to the initiation of the sign of corrosion, cracking or spalling, rather than the time to collapse of the structure. Thus, by comparing the probabilities of different structures reaching a defined state limit, we can adjust inspection priority accordingly.

The reliability of an asset can also be used to determine a suitable time for an asset's inspection. For critical infrastructure, the interval between successive of inspection can be a fixed time or a variable time. Traditionally, inspection is performed using a fixed (static) interval that is independent of the asset's current state. For example, as described in Section 2.1.4, Network Rail managed bridges from GB are inspected annually or every six years depending on the level of inspection complexity, and bridges in the NBI from the US are inspected every two years. This policy with static inspection time interval is also easier to execute in practice, and it is the principal policy of most current maintenance practice [174].

A dynamic inspection policy which schedules the inspection times based on the current and predicted future states could provide a more cost-effective inspection plan. Given the predicted states, the inspection time is determined using a predetermined failure threshold that ensures an asset operates within an acceptable level of reliability. For example, Wang and Liu [176] modelled different inspection levels with varying time intervals that vary based on the states of the components and a threshold level for each state. Hajipour and Taghipour [65] modelled a non-periodic inspection scheme that takes the degree of deterioration and corresponding reliability into consideration. The reliability can provide information about the probability of an asset deteriorating to a dangerous state in the future. Hence, decision makers can use this information to decide on inspections or repairs to reduce this probability to an acceptably low level.

2.4.2 Repair Decision

Maintainers perform a repair action when there is an indication (from inspection or prediction) that the asset is in a repairable condition. As described in Section 2.1.4, repair actions can be classified into three types: minimal, imperfect and perfect. Different maintenance strategies have been implemented considering these differences.

Nakagawa and Kowada [121] defined the reliability properties of an asset that undergoes a minimal maintenance strategy. They explained the use of this strategy with a system that is replaced periodically at a scheduled time, with minimal repairs between the replacement. Executing this policy may be cost-effective, and is especially suitable for a system with many identical components [141]. However, applying this policy may result in low system reliability, and it is not suitable for maintaining safety-critical infrastructure, such as bridges, where the consequence of failure is catastrophic.

Some repair models consider only perfect repair actions, such as replacement and renewal [36], as this is simpler when analysing deterioration and life cycle cost. However, this strategy would result in substantial repair cost and is usually only suitable for a structurally simple system [141].

In between minimal and perfect maintenance, imperfect maintenance is often more practical. For critical infrastructure, many repair activities do not restore the asset to a condition which is as good as new but rather to an acceptable level of performance. The use of a broader range of repair actions, including those that are imperfect, requires more complex models. Pham and Wang [141], Wang [174] and Wu and Zuo [185] summarised and discussed a comprehensive list of literature in this area.

Realistically, the repair actions available for critical infrastructure often comprise a mixture of minimal, perfect and imperfect maintenance actions. The Bridge Deck Preservation Matrix from the Michigan Department of Transportation [116] gives us an example of deck maintenance. It shows that actions like patching, which is a minimal repair, can slightly improve the condition of a deck top surface, while a deck replacement is a perfect repair and a shallow concrete overlay is an imperfect repair which can bring a deck top surface back to good condition.

Apart from repair effectiveness, some maintenance strategies also consider other constraints, such as system reliability and cost. Wang [174] summaries several types of policies for single component maintenance. Among them, age-dependent and periodic preventive maintenance policy suggest maintenance at predefined fixed age and time interval respectively. They describe the development of these simple policies, from assuming perfect maintenance by replacement at each intervention [11, 15], to considering imperfect maintenance effects [141, 175]. Despite the variations, in both age-dependent and periodic policies, maintenance is independent of the asset's history of deterioration and do not take the most recent condition into account. An alternative is a failure limit maintenance strategy [174], which considers repair when the deterioration, measured by the reliability function for an acceptable condition, reaches a predefined level.

The failure limit policy ensures asset maintenance is performed at or above an acceptable level of reliability. After the repair has been completed, the condition of the asset improves based on the effectiveness of action. Given the new condition distribution, we can measure the new reliability and its properties such as MTTF. For example, Bergman [16] proposed an optimal replacement strategy based on a state-dependent failure rate function, and Malik [106] measured it with its reliability. The range of maintenance policy options has been extended beyond those based on the effectiveness of different repairs: for example, Dohi et al. [37] suggest choosing the repair based on a repair cost limit and Dohi et al. [38] choosing the repair based on a repair time limit. Castanier et al. [21] incorporated the failure time limit policy and developed a sequential maintenance policy: instead of a fixed inspection interval, the interval was evaluated using a failure threshold. With the increase in age, most assets present a need for more frequent repairs [174] and varying the maintenance schedule interval makes more cost-effective repair decisions possible.

2.4.3 Maintenance Plan

After we understand the deterioration behaviours of an asset and the effectiveness of the different repair that may be relevant, we can schedule maintenance activities for some time in the future. How to plan these activities cost-effectively, considering multiple intervention cycles, have been studied extensively. A review of these maintenance plans can be found in Wang [174]. These studies also inspired the development among the industry. For example, the implementation of a bridge management system by the US state transportation departments know as 'AASHTOWare' Bridge Management (formerly known as Pontis) [167]. The system uses Markov models for deterioration prediction, it recommends repair works depend on the prediction, and also, it allows us to plan multiple future maintenance activities constrained by various objectives, such as cost and reliability.

Maintenance plans can help us better allocate resources and manage asset investment. For example, when building a new bridge, we would like to perform an asset life-cycle cost analysis (e.g. in Seif and Rabbani [150]) to estimate the total cost of maintenance in its expected service life. But as pointed out in Wang [174], an optimal maintenance plan needs to consider both repair cost and reliability simultaneously. In a finite planning horizon, for example, in the next 50 years, assets may require multiple interventions. Therefore, maintenance decisions need to decide suitable repair actions at a suitable time that can balance the repair cost while ensuring system reliability. This forms a Multi-Criteria Decision Making (MCDM) problem, and it is often NP-hard [61] because of the complexity of deterioration rates and system configuration.

MCDM requires an evaluation of multiple conflicting criteria to give optimised solutions, for example, repair cost and reliability are two conflicting criteria: always repair an asset can result in a high level of reliability but it also costs a lot. An exhaustive searching of all possible decision combinations is time-consuming, in the field of asset maintenance, studies usually apply heuristics to solve these optimisation problems as they can find a good solution at a reasonable computational time. For example, Tabu search was applied in Higgins [69] to find the optimal allocation of maintenance activities for rail tracks in a 4-day planning horizon. Genetic algorithm is another popular heuristic algorithm when planning maintenance, for example, Audley [9] used it to schedule repair actions for rail track, it helped him accomplish less than half of the life-cycle cost of a standard (non-optimal) maintenance plan; Le [94] used it to produce a list of optimal plans (trade-off between component conditions and life-cycle cost) to select repair options for bridge components in a 60-year planning horizon; Yang et al. [186] used it to find the optimal combination of pavement repair actions with maximum remaining life and minimum maintenance cost in a 20-year planning horizon.

2.4.4 Reliability Analysis Technique

A variety of methods have been implemented to help engineers decide maintenance-related decisions for infrastructure asset. Among them, we have traditional reliability analysis methods such as Fault Tree Analysis (FTA) and Event Tree Analysis (ETA), as well as Petri net, which gains more and more attention recently.

FTA and ETA are two commonly used logical representation techniques when modelling asset systems and inferring decisions. FTA is a top-down approach. It begins with the consequence and looks downward to search for all possible combinations of causes. ETA is a bottom-up approach. It starts with an event and looks for all possible consequence events. They both are used to make basic decisions, for example, Duan and Zhou [40] used a modified fault tree to diagnose faults, and Estes and Frangopol [43] used an event tree to decide whether to replace the deck or not. Nivolianitou et al. [128], Arunraj and Maiti [8], Mahboob [104] and Shang [154] compared these two traditional techniques in the reliability area.

Both FTA and ETA assume simple asset configurations. For example, they often assume the failures of components are independent, repairs are performed at a static time, or when making decisions, the combination of causes is deterministic [104, 6]. But for critical infrastructure, their failure interactions are usually complicated, and decisions must take account of uncertainty. Failing to model complex systems and handle uncertainty restricts the choices these techniques can suggest. Moreover, the structures of fault tree and event tree increase exponentially with the increase in, such as the number of components or states [77, 104], limiting the use of these techniques in maintenance modelling.

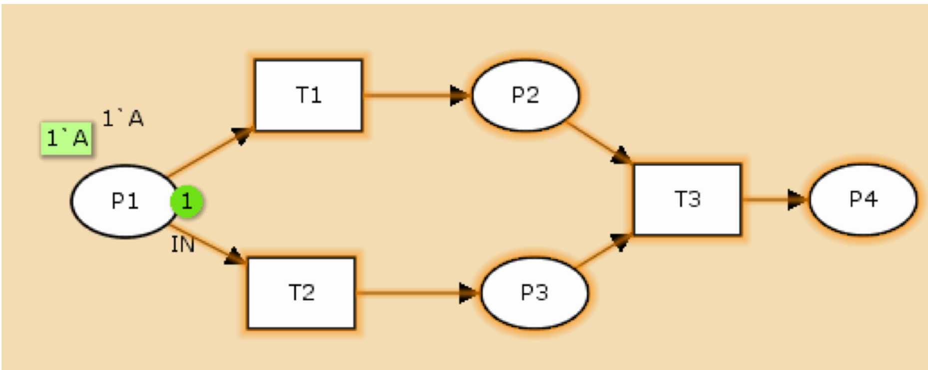


Figure 2.5 A Coloured Petri Net example.

The use of Petri nets for maintenance modelling and supporting decisions reasoning has gained attention recently [179]. In general, a Petri net consists of four fundamental elements: place, transition, arc and token. In a standard Petri net, tokens are indistinguishable - that means there is only one type of token. To overcome this, we often extend it to another type of Petri net: Coloured Petri Net (CPN) [72]. CPN allows a token to have an additional level of information - a data value called colour. An example CPN is shown in Figure 2.5. A place can hold tokens and connects to a transition through an arc. ‘P1’ in the figure is a place that holds a token with colour ‘A’. A transition (e.g. ‘T1’) may fire the tokens to another place when there are enough tokens, and the corresponding arc is satisfied (e.g. by an arc expression called a guard function). But the tokens are randomly selected when there is more than one possible transition, for example, in this figure, since no guard functions are defined, both transitions ‘T1’ and ‘T2’ are possible routes for token ‘A’ to pass through. This non-deterministic firing procedure gives Petri net the power to model reliability problems with a complex relationship between events [117].

Andrews [5] modelled the deterioration, inspection and maintenance processes of a track section with Petri nets. The model predicts track condition over time. With Monte Carlo simulations, the model can investigate the effectiveness of different maintenance strategies. This model only considers individual track modelling, and later Rama and Andrews [145] extended it to consider track as a multi-component system. Audley [9] continued the above work, developing both a single-section track state model and a multi-section track state model. Monte Carlo simulations are performed to predict the condition distribution and to investigate the number of intervention actions needed within a specific time. It also applied a genetic algorithm to find an optimal maintenance strategy.

For reliability modelling, the standard Petri net is often extended as CPN to reduce the model space. For example, when modelling multiple components, using a CPN allows multiple occurrences of a component to be analysed using a single subnet by tracking individuals using different coloured tokens [94, 95]. It can reduce the size of Petri nets and offer the flexibility to model multiple components. It has been applied in many maintenance modelling problems, for example, Le and Andrews [95] modelled several bridge components using CPNs. Each component includes the modelling of deterioration with a non-constant failure rate and the maintenance effects. They are further combined as a complete bridge model. Another example is its application in modelling bridge deterioration with a 2D-system condition scale (condition of a component is rated based on the severity extent rating in a condition matrix) in Yianni et al. [189]. These models can predict lifetime condition of components or bridges, estimate the number of maintenance actions needed, or make decisions like opportunistic maintenance [94, 95, 189]. To evaluate different maintenance strategies and to perform life-cycle cost analysis, Le [94] implemented a genetic algorithm to optimise the selection of repair actions based on different repair effectiveness and costs.

However, when combining components with different subnets into a full net, the size of CPN may still become unmanageable, especially for a system with a complex configuration [179] (e.g. configurations described in 2.3.3). For efficiency reason, Petri net is often restricted within a minimum model size in most studies [6]. We need a methodology that enables modularisation [179], for example, allows us to disassemble the model and perform analysis in a series of manageable subsections [6].

Reliability analysis in Petri nets is often performed based on Monte Carlo simulations or their variants. It can lead to inefficiency when coping with rare events or events with a small number of data [179]. Apart from that, quantitative techniques such as fault tree, event tree and Monte Carlo simulation techniques evaluate reliability using pure mathematical models, while their parameters themselves can be highly uncertain [104]. These techniques use point estimators (Section 2.3.1), which give up valuable information such as the quantiles of the parameters. We need a method that can handle this uncertainty while preserving as much information as possible.

2.5 Summary

This chapter has introduced the background to bridge asset maintenance and reviewed existing work on deterioration prediction and maintenance-related decision supports. It provides the background and the motivation for the rest of the thesis.

Section 2.1 presents maintenance-related concepts and practices implemented in most maintenance problems. Bridge asset comprises of multiple components. The condition of a bridge or its component is often discretised into several states, where each state represents a level of deterioration. The rate of deterioration can then be represented by the rate of transition between states and can be influenced by many factors, such as age and loading. This discussion leads to two reviews of deterioration prediction: one introduces the data can be used to estimate the deterioration rate, which is in Section 2.2; the other one introduces the techniques to model asset deterioration, which is in Section 2.3. With the deterioration prediction, we can make relevant maintenance decisions. In most current practice, inspection is performed at a fixed time interval, and most repair policies comprise of various repair actions with different effectiveness. Section 2.4 describes how these decisions are made and how they can be improved.

Section 2.2 focuses on the data that can be used to predict asset deterioration. It first introduces how the deterioration time data can be inferred from inspection records and points out these data are often uncertain and sometimes we only have a small amount of them. It also describes how the factors that influence the rate of deterioration and could, therefore, be used to give a prediction tailored to an individual asset. Since there are many different features, we described approaches that have been used to select the subset most related to deterioration. With these features, we can cluster assets into groups by their features, so that within each group, assets have similar deterioration behaviours.

Afterwards, in Section 2.3 we describe the modelling of deterioration. It first reviews the techniques for estimation of the deterioration rate, which includes point estimation techniques and statistical distributions. The limitations of point estimation techniques are explained: most methods cannot handle uncertainty well and often they require a large number of data to learn the transition probability. In contrast, statistical distribution gives a framework to express a stochastic function that considers the uncertainty of the deterioration process over time. Using such a function, not only gives us a choice to express prediction as an interval estimation but also provides us with a platform to express knowledge in its interpretable parameters. The rest of this section discusses two challenges for asset deterioration modelling. The first is that an asset deteriorates through a number of states from new to completely failed: some approaches assume that the transition rate is the same for all pairs of states, but others allow the rate to vary. The second challenge is handling multi-component systems, where we can have a range of different configurations, including parallel, series and bridge structure configurations.

Section 2.4 focuses on the decisions support for maintenance activities. With the prediction of the future condition of an asset from its past deterioration, we can make decisions about maintenance. To plan inspections, we can either prioritise the assets for inspections based on their risks of deteriorating to an unacceptable level or determine a suitable time for inspection. Given estimates of different maintenance actions, we can use a failure limit policy to decide which repair to perform. The model of deterioration and the effectiveness of repairs together give a maintenance strategy, so that we can plan the maintenance activities over a time horizon. Choosing the best maintenance strategy has been studied using optimisation techniques. The final part of the section discusses the techniques usually used when reasoning these decisions, it points out most current techniques suffer difficulty in modelling system with a complex configuration and handling uncertainty.

The problems and challenges that exist in deterioration prediction and decisions supports are addressed by the methodologies introduced in Chapter 3. Bayesian techniques give us a framework to describe events with credible intervals in contrast to point estimates, as well as allowing us to incorporate prior knowledge into models. With these features, in Chapter 4 we show how to use the Bayesian approach to predict deterioration from data with uncertainty. We also show how to make use of the influence factors to provide an individual prediction, and moreover, to group similar assets so that we can learn between groups. These prediction models are later applied in a case study using the NBI dataset in Chapter 5, where we compare their performance with other existing methods. To support maintenance decisions, first we show how to model multi-state and multi-component assets in Chapter 4, applying the models in Chapter 6 using real-world case studies. We show how to support a variety of maintenance decisions, including inspection priority and inspection time, suggesting repair actions and evaluating repair effectiveness. These models are organised in Chapter 7, where we show how different maintenance model can be built for different situations in a manageable manner.

Chapter 3

Bayesian Networks and Their Applications to Maintenance

This chapter introduces the Bayesian modelling methodology and its applications to maintenance. Two types of models are discussed in this chapter, one is the Bayesian Networks (BNs) that use Bayes' theorem to reason about the relationship between events (variables) using both prior knowledge and evidence (data), the other is the Bayesian statistical model that uses Bayes' theorem to estimate the parameters of a statistical model. The development of inference algorithms (Section 3.3 below) allows us to use hybrid BNs, which are BNs with both continuous and discrete variables [118]. We first introduce the two types of model mentioned above: the BN and the Bayesian statistical model. Subsequently, we use the term 'hybrid BN' uniformly.

Section 3.1 uses a simple example to show how to use Bayes' theorem to reason from evidence. BNs are introduced in Section 3.2 and used to reformulate the example. This section also shows how to assign prior probabilities and how BNs can be used to support different types of decisions. Section 3.3 describes the inference algorithms for Bayesian models, pointing out the recent development in approximate algorithms that allow us to perform inference in a model with both continuous and discrete variables. Section 3.4 discusses Bayesian statistical models, which represent the parameters of a statistical distribution as variables in the model. Section 3.5 describes how to extend Bayesian models for complex problems. Section 3.6 summaries the current applications of the Bayesian models in maintenance modelling, pointing out their potential advantages, which are the focus of the rest of the thesis.

3.1 Bayes' Theorem

Frequentist statistics interpret the probability of an event as its relative frequency in a large number of samples. In contrast, Bayesian statistics interpret probability as a subjective degree of belief under uncertainty [52], where this belief may derive from experiments or prior knowledge (the prior). The Bayesian perspective is wider, as it includes 'subjective' probabilities that could not be measured by collecting samples. With this perspective, probability can be used to reason under uncertainty coherently. When there is a new evidence (the value of an event, also called an observation), the probability of some hypothesis (modelled as a variable) is updated using the prior probability of the hypothesis and the probability of the evidence given the hypothesis (the likelihood function). The new probability that results is called the posterior probability. Bayes' theorem therefore provides us with a mathematical framework to combine prior beliefs with evidence to update our beliefs.

$$P(H|E) = \frac{P(H)P(E|H)}{P(E)} \tag{3.1}$$

Equation 3.1 is the Bayes' theorem, where E is a new observable evidence and H is a hypothesis or a model parameter that its probability is affected by E. $P(H|E)$ is the posterior probability of H given E. $P(H)$ is the prior probability of H before E is observed. $P(E|H)$ is the likelihood function of the probability observing E given H. And $P(E)$ is the marginal probability of E, it serves as a normalising constant.

Bayes' theorem shows how the prior knowledge is updated when a new piece of evidence is considered. For example, it allows us to adjust a prior belief about a bridge failure given an observation of its characteristic. In a hypothetical world, 0.1% of bridges failed in the last decade, and from historical incident reports, we know 30% of these failed bridges were overloaded for 10 years. Meanwhile, we also know that 20% of all bridges in this world are overloaded for 10 years. By using Bayes' theorem, we can measure the probability of a bridge failure given it is overloaded. Let H represents the event of the bridge is failed, E represents the event of the bridge is overloaded, from Equation 3.1, we have:

$$P(H|E) = \frac{P(H)P(E|H)}{P(E)} = \frac{0.1\% * 30\%}{20\%} = 0.15\% \tag{3.2}$$

In the example, when we presented an evidence that a bridge is overloaded for 10 years, the probability of the bridge failure rises from 0.1% to 0.15%. This process of updating the probability of variables given information is called Bayesian inference. Here we only have two variables (event E and H), but when we want to build a model with many variables and complex dependencies, we can use a graphical model called a BN, which is introduced in

Section 3.2. With an increase in the number of variables and the complexity of dependencies between variables, the difficulty of updating probabilities between variables increases sharply. To perform inference in a BN model, we need an effective inference algorithm and this is discussed in Section 3.3.

3.2 Bayesian Network

Complex interrelated problems with many probabilistic variables related using Bayes' theorem can be represented as a BN. A BN consists of a graphical structure and a number of variables. The causal or influential relationships between variables are specified by a directed graph.

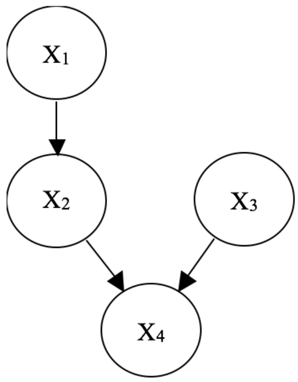


Figure 3.1 A simple Bayesian Network example.

A simple BN example is presented in Figure 3.1, it models the dependencies among four variables ($n = 4$): $X = \{X_1, \ldots, X_4\}$. X_2 depends on X_1 and X_4 depends on variables X_2 and X_3 so that we say X_1 is the parent of X_2, while X_2 and X_3 are the parents of X_4; X_1 and X_3 have no parents and X_4 has no children. The joint probability distribution of a BN is calculated using the following equation:

$$p(X) = \prod_{i=1}^{n} p(X_i | parents(X_i)) \tag{3.3}$$

Here, $p(X)$ is the joint probability of the variables in the BN model, given by the product of the conditional probability of each variable X_i given its parents. The joint distribution of this example is $p(X_1, X_2, X_3, X_4) = p(X_4|X_2, X_3)p(X_2|X_1)p(X_1)p(X_3)$. This calculation of joint distribution quickly becomes intractable as more variables and dependencies are added to the BN.

Many types of variables can be modelled in a BN, they can be classified into two types: discrete variables and continuous variables. A discrete variable has a finite range of mutually exclusive values, such as a Boolean variable has two states 'True' and 'False'; a labelled (categorical) variable has multiple states like 'Green', 'Red' and 'Yellow'; a ranked variable has multiple states with orders such as 'Low', 'Medium' and 'High' (this type of variable is further discussed later in Section 3.2.2). On the other hand, a continuous variable has an infinite range of values, for example, any values between 0 and 1. The probability distribution for such a variable can be defined by an arithmetical function or statistical distributions, such as those introduced in Section 2.3.1.

Early works on BN were mostly using discrete variables due to the computational complexity of its inference. Later Section 3.3 discusses how this limitation has been addressed by studies on inference algorithms, as a result, it allows us to build Bayesian statistical models with continuous variables as described in Section 3.4.

In this section, BNs are described using discrete variables. We can encode the relationship of a discrete variable with its parents in the form of Conditional Probability Tables (CPTs). This table contains the probabilities for each state given every possible combination of the states of its parents. For variables without any parents, the CPT is the probability distribution of that variable itself. Section 3.2.1 introduces a simple example of a BN, describing how its CPTs can be obtained from data or by elicitation from experts. The CPTs can become large, containing many parameters. Techniques to simplify the number of parameters are described in Section 3.2.2. Finally, Section 3.2.3 introduces how BNs can be used to model causal relationships between variables.

3.2.1 An Example BN: Bridge Failure

Recall the example we presented in Section 3.1 about the relationship between bridge overloading and bridge failure; in this subsection, we represent the same example as a BN. Bear in mind this is a hypothetical example for illustration purpose. We assume bridge overloading is one of the reasons for bridge failure and that this leads to an association between failure and overloading in the data. Therefore, we can model it as a parent of the bridge failure variable as shown in Figure 3.2. Both variables are modelled with Boolean nodes with two possible states: True or False. From historical record as described in Section 3.1, we know that 20% of the bridges have been overloaded over the last 10 years the CPT for the 'Overloaded Bridge' variable can be assigned as showed in the figure. The CPT of bridge failure is conditional on its parent, whether the bridge is overloaded for 10 years. Hence, there are four possible states. Since as illustrated in Section 3.1, when the bridge is overloaded, the probability of bridge failure is 0.15%, hence, bridge survival probability is

99.85%. To assign a probability for bridge failure given bridge is not overloaded, we can make use of the background information provided previously. Since the state of each variable is assumed binary, hence, $P(\neg H) = 1 - P(H)$ and $P(\neg H|E) = 1 - P(H|E)$. From Equation 3.1, we have:

$$P(H|\neg E) = \frac{P(H)P(\neg E|H)}{P(\neg E)} = \frac{P(H)(1-P(E|H))}{1-P(E)} = \frac{0.1\% * (1-70\%)}{1-20\%} = 0.0875\% \quad (3.4)$$

Overloaded Bridge	False	0.800000
	True	0.200000

	Overloaded Bridge	False	True
Bridge Failure	False	0.999125	0.998500
	True	0.000875	0.001500

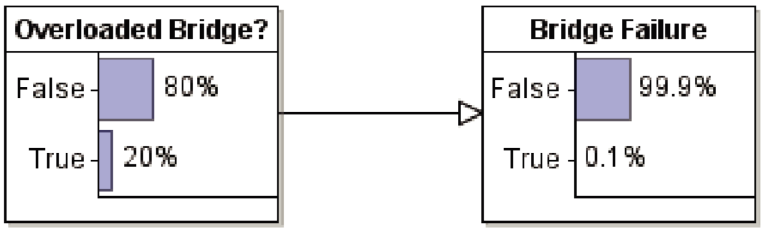


Figure 3.2 A BN example for bridge failure evaluation and its variable CPTs.

Thus, we have enough information to assign the CPT for bridge failure variable (showed in Figure 3.2). In this example model, the CPTs are derived purely from historical records about the events. But in practice, we may not have all the data needed to assign the CPTs. Furthermore, estimating parameters only from historical data limits us to describe what had happened in the past, but not things that may happen in the future. For example, if a type of bridge is newly designed, to estimate its relationship between bridge failure and loading, there are not enough historical records to construct appropriate CPTs. Instead, we can elicit them from experts. For example, the engineers may believe this new type of bridge is similar to an existing type because the two types have similar characteristics such as material and traffic volume. In this case, the engineers may have the confidence to estimate the CPTs for the new bridge type by adjusting the CPT of the earlier type.

A variety of studies have proposed different approaches to provide a reliable estimation of probability in the CPT from experts. Concerns about the accuracy of the assigned probabilities can be addressed by various probability elicitation approaches, such as verbal and numerical probability scale [86] and frequency formats [56]. Another concern was the inconsistency and bias of the probability; for this, methods such as training workshops for experts [170] and constant feedback [131] have been developed. A comprehensive guideline describing the questioning techniques appropriate for building probability assessment from experts is given by O'Hagan et al. [132].

However, as the number of state combinations of the parent nodes increases, the number of probabilities in the CPT of child node will increase, making it infeasible to elicit a probability for every possible state combination from experts or to estimate the parameters from data. The next subsection describes techniques that can be used to simplify a BN and reduce the number of values needed in its CPTs.

3.2.2 Simplification of Probability Estimation

Several techniques have been developed to reduce the workload when estimating the CPT entries, this subsection introduces three of them, which are divorcing variables, using Noisy-OR function, and using ranked node.

Divorcing Variables

A technique called divorcing [124, 127] can be applied to reduce the combinatorial explosion of values in a CPT caused by a large number of parent nodes. The divorcing technique introduces one or more intermediate nodes to act as the child nodes for some of the parent nodes. These intermediate nodes are eventually aggregated with other non-divorced parent nodes to form the target child node. By doing so, we can cut down the combinatorial space of the target child node, thus reducing the total number of probability values need to be estimated. Notes that this technique can be only applied to a situation where the effect of divorced parent nodes on the child node is independent of other parent nodes [124].

For example, consider a BN with four variables, each with three states, where variables X_1, X_2, X_3 are the parents of X_4. The resulting CPT of variable X_4 representing $P(X_4|X_1,X_2,X_3)$ has $3^4 = 81$ probability entries. Using divorcing, we introduce an intermediate node T as the child of, for example, variables X_1 and X_2. T and X_3 then become the only two parents of X_4. The original CPT breaks down into two tables representing $P(T|X_1,X_2)$ and $P(X_4|T,X_3)$. The probability entry thus reduces from 81 to $3^3 + 3^3 = 54$. The reduction becomes larger as the number of states and parent nodes increases. This technique is further developed as binary factorisation for continuous variables in Neil et al. [123], which will be further discussed and employed in this thesis later (see Section 4.3).

Noisy-OR Expression

Another common approach to simplify the elicitation process is to avoid specifying all probability entries directly by adding some assumptions about the form of the CPT. For example, Pearl [136] proposed Noisy-OR expression to encode expertise in complex CPTs. By assuming the effects of parent nodes are independent, Noisy-OR reduces the number of parameters in the CPT. A leak probability was often added as a dummy parent in an extended version of the Noisy-OR. This dummy parent is always true in the model, representing all other causes that are not included in the model. The limitation is that it can only be used for Boolean variables. Díez [34] proposed the Noisy-Max function to tackle this restriction. Recently, Noguchi et al. [129] tackled the reasoning limitation of the Noisy-OR function by perfect its reasoning with conditional inter-causal independence property.

Ranked Node

Fenton et al. [46] introduce the ranked nodes, which reduce the number of parameters needed to specify a CPT and therefore simplifies the elicitation process. A ranked node has multiple ordered states, for example, ‘Low’, ‘Medium’ and ‘High’, mapped to sub-intervals in the range 0 to 1. Encoding knowledge in this way has two primary benefits. One is experts would find it easier to express their opinions about an event using these states rather than using numerical values. The other one is it allows us to express the differences between states with orders, for example, compared to a variable that is rated as ‘Low’, a variable rated as ‘Medium’ is closer to a variable rated as ‘High’.

See an example in Figure 3.3, which has three variables: the variables ‘Quality of Equipment (C1)’ and ‘Skill of Engineer (C2)’ are the parents of ‘Maintenance Effectiveness (M)’. Each variable has five states and can be rated from ‘Very Low’ to ‘Very High’. Traditionally, we can encode these nodes as categorical variables (labelled nodes) where each has five states. The resulting CPT for node M has $5^3 = 125$ states, which would require extensive work on evaluating the probability entries. But if we model them using ranked nodes, it would significantly reduce the number of entries.

Fenton et al. [46] proposed mapping the state of a ranked node to a numerical ordinal scale goes from 0 to 1 in equal intervals, for example, the state of ‘Very Low’ is mapped to a numerical range of [0, 0.2), ‘Low’ is mapped to [0.2, 0.4) and so on so forth. Since the states have been mapped to numerical scales, we can express the CPT of the child node with a numerical statistical distribution, using a doubly Truncated Normal distribution (TNormal). TNormal, a normal distribution bounded by lower and upper limits, is applied to generate CPT of the child node. The TNormal here is denoted as TNormal $(\mu, \sigma^2, 0, 1)$, where μ is

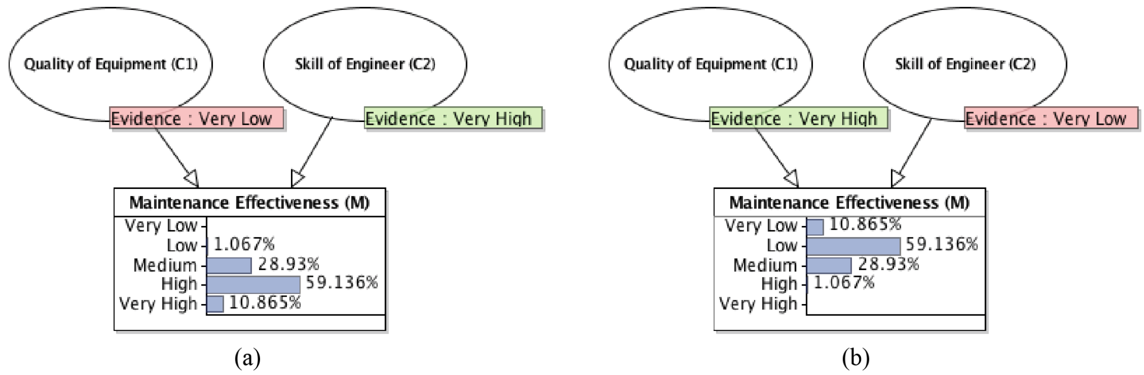


Figure 3.3 Maintenance effectiveness modelled with ranked nodes.

the mean and, σ^2 is the variance, and $[0,1]$ is the bounded finite range. One of the advanced usages of the ranked node with TNormal distribution is to use the weighted average of its parent nodes as μ to manipulate the skew of the CPT. Experts can represent their knowledge as the weights of the events and use σ^2 to represent the confidence level of this knowledge.

Figure 3.3 shows an example, where the expression for node maintenance effectiveness is:

$$\text{M} \sim \text{TNormal (wmean (0.3, C1,0.7, C2), 0.01, 0, 1)}$$

M $\sim$ TNormal $(\cdot)$ represents M follows the TNormal distribution. wmean is a shorthand for a weighted mean function. This function is equivalent to a linear model by averaging the values of variables (C1 and C2) by their weights (0.3 and 0.7). This means, C2 (with a weight of 0.7) has a higher impact on M compares to C1 (with a weight of 0.3), and the expert that assigned these weights has a confidence level of 0.01 (as the variance of the distribution). Figure 3.3, in which the CPT of M has been assigned in this way, shows the posterior of M for two different scenarios. In both scenarios one of C1 and C2 is ‘Very High’ and the other ‘Very Low’: in Figure 3.3 (a) the more highly weighted C2 is ‘Very High’ and the expected value of M is high, while in (b) the expected value of M is lower as the ‘Very High’ C1 has a lower weight. When relying on expert judgement to form a CPT, the use of ranked nodes requires many fewer parameters – in this case, just three (a weight of C1, a weight of C2 and a level of confidence) compared with the 125 probability entries needed if discrete nodes had been used. The advantage of the ranked node becomes more significant with the increase in the number of states in each node and the increase in the number of parent nodes. This technique is later applied in this thesis for evaluation of features’ impact on deterioration (Section 4.2) and condition evaluation of asset assembled by multiple components (Section 4.3).

3.2.3 Observational and Intervention BNs

One of the advantages of using BNs is the possibility of modelling causal effects between variables. With different causal representation, the reasoning process may differ. For example, inference on a model with a variable that is observed with a state ("seeing") and the same state is generated by an intervention ("doing") is different [160, 137]. These two types of modelling are named as an observation model and an intervention model respectively.

When performing the inference, observing a variable results in updated probabilities in both its causes and effects. For example, assume the overloading of a bridge can cause cracks, and cracks may lead to the bridge to fail. This can be modelled by a BN with a causal chain structure. With the inference, if we observed that there are severe cracks in a bridge deck, we can diagnose this bridge has a higher probability of being overloaded for over 10 years (cause), and predict this bridge has a higher probability of failing (effect). These inferences can be adjusted based on the causal structure of the model. Three basic causal structures are given in Figure 3.4 (top level) in the form of observational models developed by Hagmayer et al. [64], where their joint probabilities are modelled as:

- Causal chain: $P(X,Y,Z) = P(X)P(Y|X)P(Z|Y)$
- Common effect: $P(X,Y,Z) = P(X)P(Y|X,Z)P(Z)$
- Common cause: $P(X,Y,Z) = P(X)P(Y|X)P(Z|X)$

By setting the variable to the observed value (represented by shaded ovals in the figure), we can calculate other events conditional on the observed variable in the usual way. Following the bridge cracks example, with the causal chain structure model, where X represents whether the bridge is overloaded for over 10 years, Y represents whether cracks exist, and Z represents whether the bridge is going to fail. Each variable is modelled as a Boolean variable where the state of true is represented as 1, and false is 0. The probability of the bridge is overloaded for over 10 years $P(X = 1)$ given an observation that cracks exists $P(Y = 1)$, we have $P(X = 1|Y = 1)$ that is computed according to Bayes's theorem from Equation 3.1:

$$P(X = 1|Y = 1) = \frac{P(Y = 1|X = 1)P(X = 1)}{P(Y = 1|X = 1)P(X = 1) + P(Y = 1|X = 0)P(X = 0)} \tag{3.5}$$

Assume the probability of a bridge that is overloaded over the last 10 years is 0.2, $P(X = 1) = 0.2$. Since in this example the variable states are all binary, thus $P(X = 0) = 1 - 0.2 = 0.8$. Assume the probability of an overloaded bridge will cause cracks in 10 years is 0.7, $P(Y = 1|X = 1) = 0.7$, and the probability of having cracks even the bridge

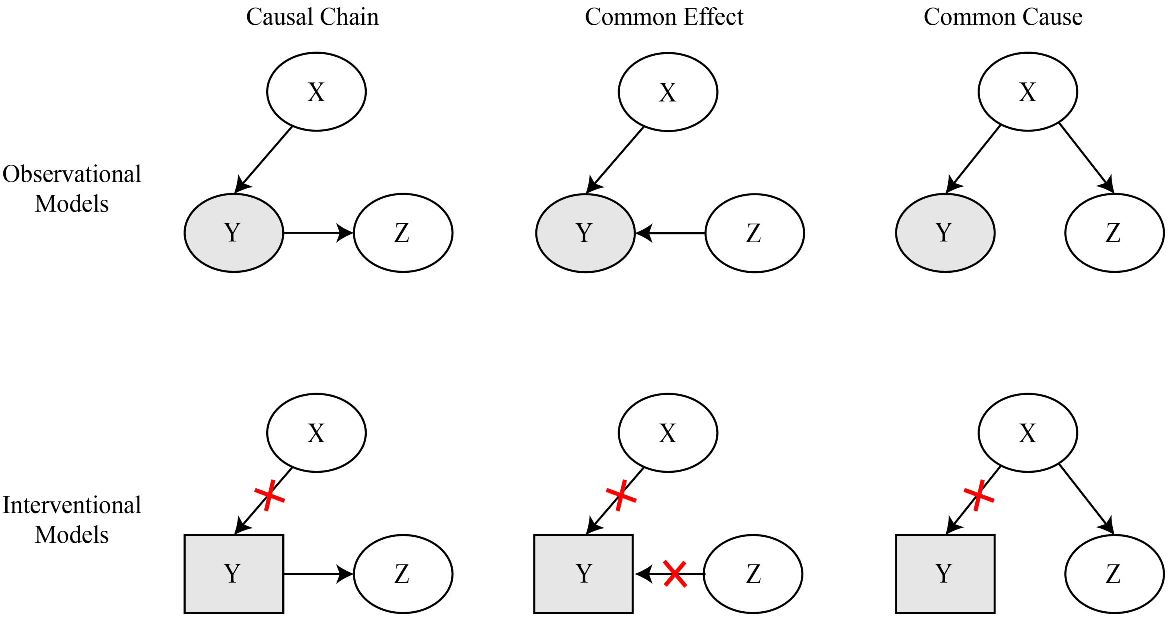


Figure 3.4 Examples of three basic causal models represented in observational and intervention modes from Hagmayer et al. [64].

is not overloaded is 0.4, $P(Y=1|X=0)=0.4$. From Equation 3.5, $P(X=1|Y=1)$ is therefore equals to 0.3. Given an observation that there are cracks, the probability of bridge is overloaded for over 10 years increases from 0.2 to 0.3. Similar procedure can be performed to infer the probability of bridge failure given that there are cracks.

In this example the variables are linked with is a causal chain structure, but the conditional probabilities will be reasoned differently given different causal structures. Hagmayer et al. [64] and Meder et al. [111] provided a detail explanation of the use of other structures. These causal structures can also be extended to model a number of other structures, such as causal chain confounder and common cause confounder structures [112, 155]. Of these, the common cause confounder structure is later modelled in this thesis (see Section 4.4).

In an intervention BN model, the intervened variable is assigned a value in the same way as an observation. However, we must remove all the links from its causes to prevent backward reasoning about the causes given the value assigned by the intervention. The corresponding intervention models (we use a square node to represent an intervention) with the three basic causal structures are shown in Figure 3.4 (bottom level) altered from their observational models (showed by red crosses).

Follows the same example of bridge failure, in the observational model, the evidence of having cracks indicates a higher probability of bridge is overloaded for 10 years, and a higher probability of bridge will fail. But if the cracks are caused by environmental issues, it does not provide evidence about a bridge is overloaded either historically or in the future.

Manipulating the cracks existence (e.g. by repair) independently of the causes of cracks disconnects the cracks from their overloading cause, that is, mended the cracks does not imply the bridge was or will be overloaded. Hence, in the intervention model, the probability of bridge failure only depends on the existence of cracks, not on its loading.

To distinguish interventions from observations, Pearl introduced the use of do-operator [137]. For example, instead of $P(X|Y=1)$ that denotes the probability of X given Y is observed with a true state, $P(X|do(Y=1))$ refers to the probability of X given that the state of Y is fixed to be true by an intervention. This operator renders a variable independent of all its causes when an intervention is performed. This process is also known as graph surgery.

For example, in Figure 3.4, assume in all observational causal structures, event Y is observed as $Y=1$, while in the intervention models, event Y is manipulated as $do(Y=1)$, we have their joint distributions:

- Observations:
  - Causal chain: $P(X,Y=1,Z)=P(X)P(Y=1|X)P(Z|Y=1)$
  - Common effect: $P(X,Y=1,Z)=P(X)P(Y=1|X,Z)P(Z)$
  - Common cause: $P(X,Y=1,Z)=P(X)P(Y=1|X)P(Z|X)$
- Interventions:
  - Causal chain: $P(X,do(Y=1),Z)=P(X)P(Z|Y=1)$
  - Common effect: $P(X,do(Y=1),Z)=P(X)P(Z)$
  - Common cause: $P(X,do(Y=1),Z)=P(X)P(Z|X)$

With the observational BN model, we can use observation of evidence to update the probability of other variables. For example, with the backward reasoning, given a safety or reliability criterion as an observation, the observational model can tell you what repair action was more likely taken in history while the intervention model can give us an estimation of the effectiveness of different repair actions. By comparing the effectiveness of different maintenance actions, we can prioritise the maintenance actions. These models are developed in this thesis and presented in Section 4.4.

3.3 Bayesian Inference

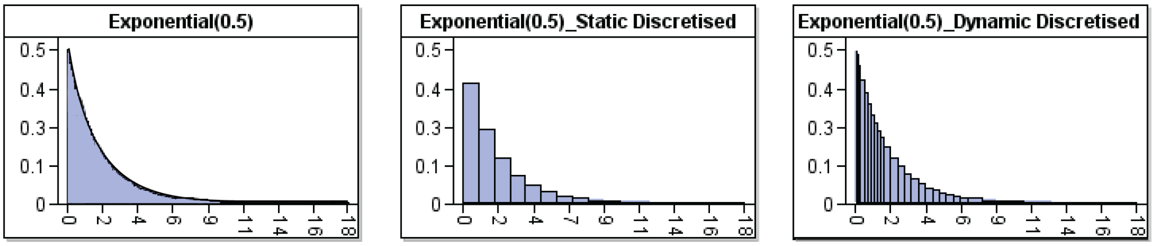
When some variables have a known state, an inference algorithm can update the probability distribution of the remaining variables, using Bayes' theorem. This process is called Bayesian inference as introduced in Section 3.1. However, Cooper [30] found the computational complexity of exact inference in a BN can be NP-hard.

Various exact inference methods are developed to tackle this challenge. For example, the variable elimination algorithm removes the non-observed non-query variables one by one by distributing the sum over the product [151]. The junction tree algorithm is another commonly used exact inference algorithm developed in the late 1980s. This algorithm performs propagation on a modified graph called a junction tree (constructed from the triangulated graph) that computes marginal probability values for variables. However, these algorithms work only with discrete variables, which is a barrier to maintenance modelling where we may need both discrete and continuous variables in a hybrid BN.

Local exact inference in hybrid BN can be executed under the assumption of conditional Gaussian distributions [153]. Early work carried out exact inference in hybrid BNs was first proposed by Lauritzen [92] and later developed by Lauritzen and Jensen [93]: the conditional distribution of a discrete variable is multinomial given its parents, while the conditional distribution of a continuous variable is configured by a linear regression model given its parents. However, this inference impose Gaussian assumption on all continuous variables limits the domain can be modelled: it is impractical for models with a mixture of discrete variables and non-standard distributions such as Weibull distributions.

Approximate inference algorithms have been studied to address this challenge. Murphy [119] discussed a list of approximate inference methods. One of the widespread approaches is Gibbs sampling, which is a Markov chain Monte Carlo (MCMC) algorithm that approximates a specified multivariate probability distribution to obtain a sequence of observations when direct sampling is difficult. Spiegelhalter et al. [159] extended this approach to include non-random variables computed from other variables deterministically. However, the approximation may not converge to represent the actual distribution when the sample amount is not large enough.

Another branch of approximate inference is discretisation. Static discretisation allows approximate inference with both continuous and discrete variables. In this algorithm, states of a continuous variable are mapped into a pre-defined finite set of discrete states. An example of the discretisation is presented in Figure 3.5. Figure 3.5 (a) shows the actual continuous distribution - an exponential distribution with a rate parameter of 0.5, and (b) tried to approximate it by discretisating the distribution into several states (showed as bars in the figure). However, the process could be tedious since the user needs to define the state intervals (each state has a value interval of 1 in this example). In addition, discretisating continuous variables into static states may lose lots of useful information if the interval is not small enough. For example, in Figure 3.5 (b), the probability of being at a value between 0 and 0.5 is much higher than the probability of being at a value between 0.5 and 1, but since the interval in this static discretisation is 1, both situations are grouped into the range of 0 to 1 with the same probability.



(a) (b) (c)

Figure 3.5 Discretisation of a continuous variable: (a) the actual exponential distribution; (b) static discretised exponential distribution; (c) dynamic discretised exponential distribution.

Inspired by the work on using non-uniform discretisation in the Bayesian model from Kozlov and Koller [84], Marquez et al. [107] use dynamic discretisation in an exact inference algorithm. Continuous variables are dynamically discretised, with narrower intervals where the probability distributions are changing most. This algorithm addresses both limitations of static discretisation: it automates the process of defining state intervals and reduces the information loss by giving narrower intervals to states with more probability changes. A corresponding example of Figure 3.5 (b) is presented in (c) using dynamic discretisation. An approximate inference algorithm that combines dynamic discretisation with propagation algorithms on junction tree is implemented in the tool AgenaRisk [2] that can deal with both continuous and discrete variables under the same BN model.

A number of open source and commercial software are available for building BN models. Kevin Murphy[1] and Constantinou [26] summarised a list of packages for this purpose. Salmerón et al. [147] further reviewed a range of algorithms that allow the inference in hybrid BNs and highlighted tools supporting inference in these types of models specifically. For example, we have:

- Conditional Gaussian assumption: gR[2], Bayes Server[3] and Hugin[4];
- MCMC: JAGS[5], Stan[6] and Edward[7];

[1] https://www.cs.ubc.ca/~murphyk/Software/bnsoft.html
[2] https://cran.r-project.org/web/views/gR.html/
[3] https://www.bayesserver.com/
[4] https://www.hugin.com/
[5] http://mcmc-jags.sourceforge.net/
[6] http://mc-stan.org/
[7] http://edwardlib.org/

- Discretisation: AgenaRisk[8] and BayesiaLab[9].

Models that allow both continuous and discrete variables in the same BN model enables us to build a statistical model in a Bayesian Network, which offers us an integrated structure to estimate the rate of deterioration and use the estimation simultaneously to reason decisions. Thanks to the support of Agena Ltd [2], the models built within this thesis were developed using AgenaRisk and its API for their flexibility in building BN models and capability of inferencing hybrid BN models. The next section discusses Bayesian statistical models that use mainly continuous variables.

3.4 Bayesian Statistical Models

As discussed in Section 2.3.1, one way to estimate how fast an asset deteriorates is to find a parametric statistical distribution that can describe the rate of its deterioration. Maximum Likelihood Estimation (MLE) aims to find the best estimate of the parameter values that have the maximum likelihood of producing the evidence (data) and is a common approach for fitting a distribution. Unlike the MLE method that determines point estimates of the parameters of a statistical distribution, in a Bayesian statistical model, the parameters of the distribution are treated as random variables. By doing so, the uncertainty of the parameters is characterised, and the information from data is captured in the posterior distribution of the parameters. Moreover, in contrast to MLE where the estimate of the parameters depends solely on the data, Bayesian estimation integrates the data with any prior knowledge about the parameters.

When estimating a parametric statistical distribution in a Bayesian statistical model, Bayes' theorem in Equation 3.1 is used as:

$$P(\theta|D) = \frac{P(\theta)P(D|\theta)}{P(D)} \tag{3.6}$$

where θ is a vector of parameters. For example, in an exponential distribution $\theta = \{\lambda\}$ and a Weibull distribution $\theta = \{\beta, \eta\}$. D denotes a vector of observations that can be used to fit the selected distribution.

To get the posterior distribution $P(\theta|D)$, we need to define $P(\theta)$, $P(D|\theta)$ and $P(D)$. $P(\theta)$ is a collection of the prior distribution of the parameters (details about how to define a prior distribution will be discussed in the next subsection). The likelihood function becomes the probability density function of the distribution parameterised by θ, for example, the pdf of a

[8] http://www.agenarisk.com/
[9] http://www.bayesia.com

Weibull distribution in Equation 2.4. By assuming the data are mutually independent and sharing the same parameter θ, the likelihood $P(D|\theta)$ is the multiplication of the likelihood function for every observation. $P(D)$ is the marginal distribution of the observations, it serves as a normalisation factor to ensure the posterior density integrate to 1. $P(D)$ can be represented as:

$$P(D) = \int P(\theta)P(D|\theta)d\theta \tag{3.7}$$

It becomes more difficult to evaluate this integral analytically especially with the increase of parameter dimensionality. Fortunately, this can be tackled with the support of numerical approximation inference algorithm as mentioned in Section 3.3. To use the posterior distribution for prediction purpose, we can evaluate the distribution of an unobserved data D_{pred} that is conditional on the observed data D, where all data follow the distribution that is parameterised by θ. Therefore, the posterior predictive distribution becomes:

$$P(D_{pred}|D) = \int P(D_{pred}, \theta|D)d\theta = \int P(D_{pred}|\theta, D)P(\theta|D)d\theta \tag{3.8}$$

Assuming D_{pred} and D are all mutually independent, we have the predicted deterioration distribution:

$$P(D_{pred}|D) = \int P(D_{pred}|\theta)P(\theta|D)d\theta \tag{3.9}$$

Sometimes, where multiple distributions belong to different groups, models can be built hierarchically so that the parameters within each group have shared prior distribution and can themselves be learned from the data within each subgroup. This type of modelling is called Bayesian hierarchical modelling.

3.4.1 Bayesian Hierarchical Modelling

A Bayesian hierarchical model can extend the Bayesian parameter estimation method to include multiple layers of information. This is achieved by creating additional parameters (called hyperparameters) as the parents of parameters (local parameters) that measure the uncertainty about the parameters themselves. In a hierarchical model, the distribution of individual within each subgroup is governed by a set of parameters, and these parameters are sampled from a set of hyperparameters that describes the characteristics of all the groups together.

Let D_i denotes individual observations from subgroup i that follow a parameter set θ_i, and assume parameters $\{\theta_1, \ldots, \theta_i, \ldots, \theta_k\}$ are parameters from the different subgroups. These parameters are all governed by a set of hyperparameters Θ. For a two-level hierarchical model, we have three stages:

- Stage 1: $D_i|\theta_i \sim P(D_i|\theta_i,\Theta)$
- Stage 2: $\theta_i|\Theta \sim P(\theta_i|\Theta)$
- Stage 3: $\Theta \sim P(\Theta)$

where the likelihood function of individual observations from subgroup i is $P(D_i|\theta_i,\Theta)$ with a prior distribution $P(\theta_i|\Theta)$, and the parameters θ_i in each subgroup i are governed by hyperparameters Θ with hyperpriors $P(\Theta)$. Assuming the individuals within each group are mutually independent, and subgroups within the overall population are also mutually independent, the posterior distribution can be expressed as proportional to:

$$P(\theta,\Theta|D) \propto P(\Theta)\prod_{i=1}^{k} P(D_i|\theta_i)P(\theta_i|\Theta) \tag{3.10}$$

Notes that the parameters θ_i here represent a collection of parameters for each subgroup i. Hence, if the likelihood function has multiple parameters, Equation in 3.10 can be further dissembled with another multiplication about the parameters.

As a result, the overall population-level parameters are jointly guided by its subgroups' parameters and the individual subgroup-level parameters are informed from all other subgroups' parameters via the estimation of the overall-level parameters [85]. This suggests we could leverage the feature of Bayesian hierarchical modelling to tackle the challenge of small data amount (Objective II) via learning between one group of assets that has little deterioration data and other groups with more deterioration data.

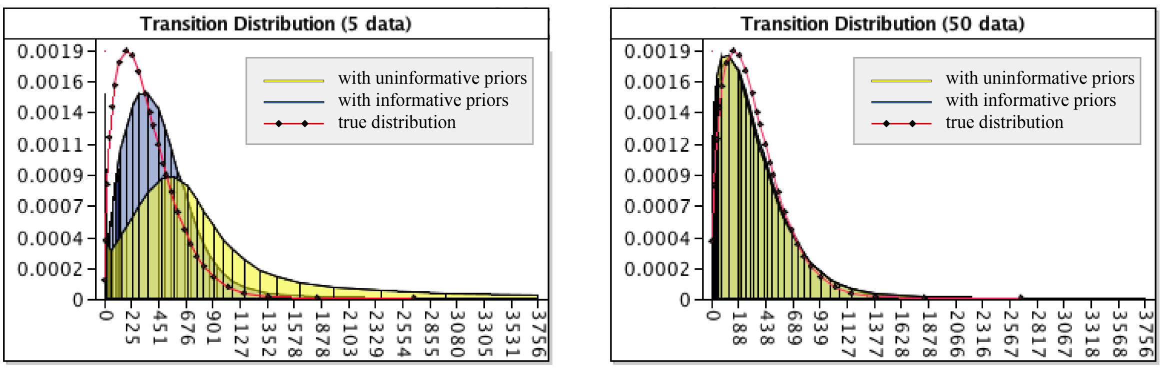
3.4.2 Prior Probability Distributions for Parameters

Estimating statistical distributions in a Bayesian statistical model requires us to quantify the prior probability distributions for any unknown parameters. This gives a natural framework to include knowledge into priors as the uncertainty specification of the parameters. These priors for parameters are modelled as continuous variables and, in this subsection, we describe the elicitation of them.

When the parameters have conjugate priors (see a list of family illustrated in Fink [48]), choosing a prior from them can give a closed-form expression for the posterior, which is advantageous in simplifying the posterior distribution calculation. However, not all distributions' parameters have conjugate priors. For example, there is no conjugate prior in a

Weibull distribution when both parameters shape and scale are unknown. Another common approach is to assign an uninformative prior when no past information is available so far. For example, by choosing a uniform distribution with extreme bounded values as the prior. With uniform priors for the parameters, the point estimate of the Bayesian method - Maximum A Posteriori (MAP) probability estimation, is identical with the result from MLE [120]. But when relevant information about the parameter is available, we can create an informative prior.

Lunn et al. [103] and Gelman et al. [52] introduce a range of methods to define an informative prior, including estimation based on data, a mixture of data and judgement, and pure expert judgement. When some historical data is available, we may obtain a prior distribution for the parameter based on an empirical estimate, for example, by matching the prior with the mean and standard deviation from the data pool. In the case when we believe the historical data is not fully representative, we may elicit knowledge from experts to discount the data weight by building a power prior [71, 103], or using a bias modelling procedure [59, 103]. Elicitation of subjective information from experts is also often used to determine a prior distribution. But subjective prior distribution about the uncertain parameter is often difficult to specify precisely. A simple approach is to ask experts directly about the central tendency and variation of the parameter as the mean and standard deviation of its prior distribution. For example, Neil et al. [125] used a triangular distribution to estimate the prior with vague ranges: a lower bound, a medium and an upper bound. The challenge is to explain and convince experts to assign these values since parameters in different distributions have different meanings and impact on their posterior distributions. Tackling this challenge is part of our objectives (Objective I) and is addressed in the next chapter.



(a) (b)

Figure 3.6 Effects of the prior distribution and data quantity in learning distribution.

To illustrate the differences when distributions are fitted with different types of priors, two examples are shown in Figure 3.6 with the corresponding configuration in Table 3.1.

Table 3.1 Fitting Weibull distributions using Bayesian parameter estimation with different priors and data amounts.

Parameter Name	With Uninformative Priors	With Informative Priors	True Value
Shape	Uniform (0.0001, 10000)	Triangular (1, 2, 4)	1.5
Scale	Uniform (0.0001, 10000)	Triangular (100, 500, 700)	400

Simulation is used to generate a set of data from a Weibull distribution with a shape value of 1.5 and a scale value of 400. The fitted distribution in yellow is governed by uninformative priors, in blue is governed by informative priors, and in red is the true distribution. From Figure 3.6 (a), we can observe that even with little data (5 data), with the help of informative priors, the posterior distribution can approximate the true distribution within a reasonable degree. This shows the advantage of choosing informative priors over uninformative priors when they are available. With the increase in sample amount (50 data), as shown in Figure 3.6 (b), both posteriors with uninformative and informative priors become almost identical to the true distribution (the yellow distribution overlaps the blue distribution). With enough sample amount (50 data in this case) to fit a distribution using the Bayesian method, the effects of choices of the prior distribution on the posteriors become minor, this is the Bernstein–von Mises theorem: with the increase of sample size, the posterior distribution for parameters is asymptotically independent of the prior distribution [171]. Conversely, if the data amount is small or only indirect information for learning the parameter is available, assigning a good prior is essential [103, 52].

3.5 Building Complex and Large-Scale Bayesian Networks

A Bayesian statistical model can summarise information comprised in the dataset and allow us to perform inference on these summarised (learned) statistics over a probabilistic framework. A BN offers us a way to propose a probabilistic model on a collection of variables, without necessarily involving data [149]. In essence, we can describe a Bayesian statistical model as a BN: the marginal distribution of a parameter is the prior; the CPT of the data is the likelihood function over the parameter; as a result, the posterior distribution is the conditional distribution of the parameter given the observed data. In the rest of the thesis, we use the term ‘hybrid BN models’ to encompass both types of models to avoid confusion.

The process of constructing a hybrid BN model structure can be demanding, especially for complex and large networks (see Section 2.3.2 and 2.3.3). In Mahoney and Laskey [105], the difficulty of using BN models for large and complex systems was highlighted. The network becomes difficult to understand when too many variables are presented in a single model [45, 26].

Often, the complexity of the model is contributed by many similar repeated fragments with the same causal (or influential) structures and variables but different entries. Object Oriented (OO) design, where each object contains a set of fixed data structures and methods interact with the data [57], gives a suitable framework to design models with many repeated structures. Early work in Laskey and Mahoney [90] implemented the idea of OO to organise the BN into various parts, where each part is a semantically meaningful unit called a fragment. Each fragment consists of a range of variables and corresponding edges. A prototype was built by arranging these fragments with a system engineering approach that follows a spiral life cycle model in Laskey and Mahoney [91]. Similar works were developed by Neil et al. [124]: inspired by Pearl [136]'s idea of assembling causal structures from a stock of building blocks, Neil et al. proposed a set of so-called idioms to serve as these reusable building blocks in a BN. These idioms represent typical types of uncertain reasoning that can be used to construct BNs based on specific cases. Adapting the concept of OO, each idiom was treated as an object, therefore can build a complex model using some simple combination rules. A software safety case was also studied using these idioms, demonstrating its capability in formulating a large-scale BN. These idioms are recurring patterns that provide an efficient and consistent guideline; with this knowledge-based guideline, these idioms allow the practitioners to understand the problem domain and Bayesian modelling principles better.

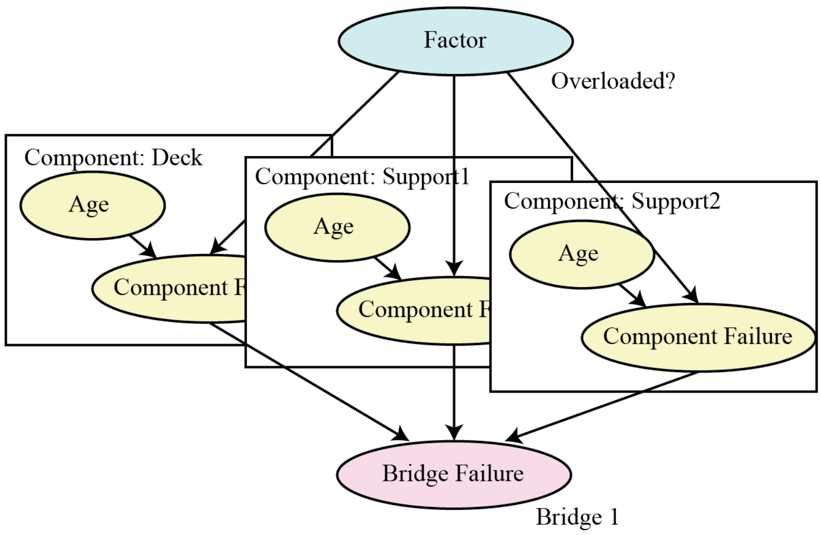


Figure 3.7 An OOBN model example about bridge failure.

At the same time, Koller and Pfeffer [82] proposed to extend the concept of OO to structure the BN parts as reusable objects and instantiate these objects to form a complete (ground) model. This type of BN is called an Object-Oriented Bayesian Network (OOBN). An object in an OOBN has attributes that can be random variables. The attributes in an object can be private and only accessible inside the object or can be inputs, or serve as the outputs of this object and become the inputs of another object managed by an external interface (Encapsulation). The object can be viewed as a stochastic function that outputs the probability distribution for each value of its inputs. The concept of class is also included in OOBN, where a model can contain multiple instances of any each class, each described using the same probabilistic model. OOBNs gives us a framework to model a generalised probabilistic model that can be reused in different contexts (Abstraction). Inheritance is also supported in an OOBN. A subclass simply modifies some of the attributes or adds new attributes from the parent class's stochastic function. These features enable us to instantiate a large BN in a well-defined way.

Assume bridge failure is determined by the failure of its components Deck, Support 1 and Support 2; component failure is determined by the age of the component and whether the bridge is overloaded (factor). We can model each component's failures using the same probabilistic model with an internal parent variable Age and an external parent Factor representing whether the bridge is overloaded that shared between all components; afterwards, all Component Failure variables are aggregated to determine the state of Bridge Failure. This example is presented in Figure 3.7 with an object class Component, which encodes the same probabilistic model for each component and this object was reused three times. However, pointed out in Pfeffer et al. [140], an OOBN is constrained in representing a fixed set of related objects only. In Figure 3.7, each Component Failure variable is fixed with two parents: its age and the factor shared between all components. While in maintenance modelling problem, the number of objects and the relationships between them could vary, for example, what if a factor has an impact on some components only instead all of them?

This challenge can be resolved using the Probabilistic Relational Model (PRM) formalism, developed by Koller [81], which extends the concept of OO in modelling probabilistic graphical models (i.e. BNs) with uncertain relations. A PRM combines probabilistic dependencies with a relational schema that describes the entities in the problem domain. The uncertainties represented in the model can include attribute uncertainty, structural uncertainty, and class uncertainty. The OO paradigm also gives us advantages in model inference: we can perform inference on specific compiled parts of the model that are queried rather than the joint distribution of a whole model where some variables may not be used in this particular case [140, 53, 114].

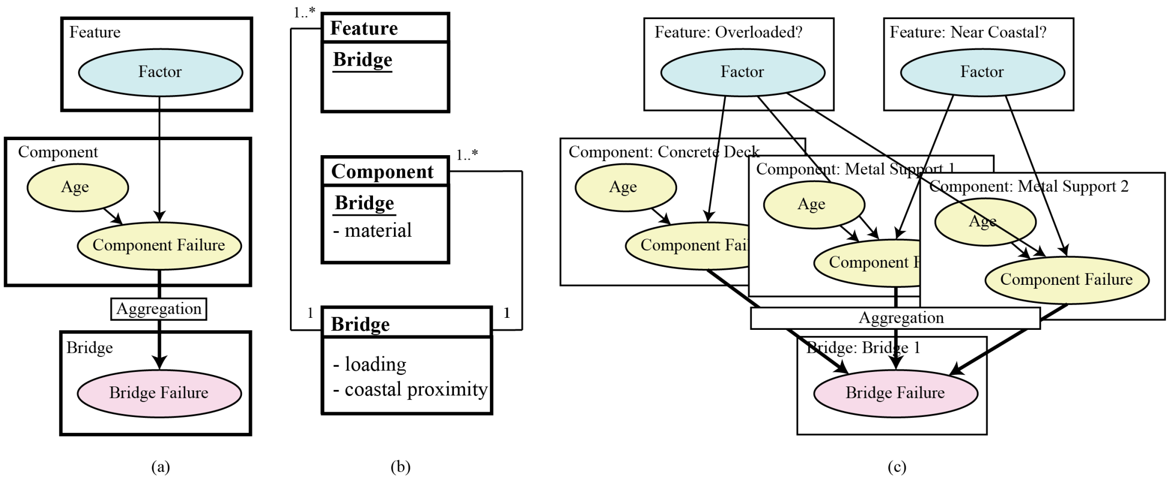


Figure 3.8 A PRM example about bridge failure: (a) probabilistic dependencies; (b) a relational schema; (c) the ground BN.

Additional to the same assumptions made in the last example, we also believe coastal proximity has an impact on component failure but only to component built with metal [187]. This example is difficult to build using OOBN as you either have to develop two object classes one for components with metal and one for others or create a dummy variable in the component object to check whether each component is made of metal and use the result to further adjust the function on component failure. Instead, PRM gives an elegant way to model this problem with two parts as shown in Figure 3.8: (a) probabilistic dependencies: every object class encodes a probabilistic model that has the same variables and dependencies; (b) a relational schema: encodes the relational structure of the problem, its instantiation is a relational database.

When instantiating the variables related to Bridge 1, the PRM first queries table Component in the relational database to see which component has a foreign key of Bridge 1, we have Deck, Support 1 and Support 2; at the same time, the PRM queries table Feature follows the same principle, we have factors about whether the bridge is overloaded and near coastal. After the instantiation of variables, the next step is the instantiation of dependencies. The relationship between Factor overloaded and component failure depends on the bridge; therefore, the dependencies are instantiated if Bridge 1 is instantiated, together, the loading value from table Bridge gives the observation on Factor overloaded. The component's material decides the relationship between Factor near-coastal and the component failure; therefore, from table Component we can see whether the component is made of metal, if yes, the dependency is instantiated and the observation on Factor near-coastal is given by the coastal proximity value from table Bridge. Variables Component Failures are later aggregated (e.g. mean, maximum or minimum) to determine the probability for Bridge Failure.

Figure 3.8 (c) presents an instantiated BN follows these procedures. Because all three components belong to Bridge 1, they all depend on Factor overloaded. Also, because the deck is made of concrete, its relationship with Factor near-coastal is not initiated, while both supports are made of metal, they both depend on Factor near-coastal. This example shows how we can use the PRM to instantiate non-fixed dependencies using the same object. Later in Chapter 7 we will explain these procedures in details, including how to instantiate a relational schema into a relational database and how to instantiate a BN follows the database and the probabilistic dependencies.

The PRM provides a separation between probabilistic models and structure relationships, gives a clear semantic in describing complex problems and an effective inference structure for the underlying models. This modelling framework is adopted in this thesis, where a number of generic BN models are developed in Chapter 4 to serve as the reusable probabilistic models that later are used as the model library in Chapter 7. To fulfil Objective VI, Chapter7 also show how to organise these models to create a maintenance model applicable in a particular circumstance according to its own structural relations.

3.6 Bayesian Network Models in Maintenance Modelling

In maintenance modelling, a range of studies has been performed using the Bayesian models for reliability modelling and analysis. In this section, we discuss the applications of BNs to deterioration modelling, inspection and maintenance decisions support as well as the current limitations, which will be tackled in this thesis.

3.6.1 Deterioration Modelling

As argued in Enright and Frangopol [41], deterioration prediction solely based on inspection records is problematic since it ignores the uncertainty in the inspection data. For example, different measurements can conclude different inspection results. But modelling with a Bayesian framework, which can combine inspection data with prior engineering knowledge, may provide a more realistic modelling for an asset. Several applications were presented to show how to update parameters of various statistical distributions via data and knowledge using the Bayesian parameter estimation framework, for example in Enright and Frangopol [41] for bridge condition prediction and in Hong and Prozzi [70] for road pavement performance prediction.

Extended use of Bayesian parameter estimation was also developed to consider more practical assumptions in asset deterioration. As described in Section 2.2.1, since inspections for critical infrastructure are often periodic, the deterioration data obtained are usually censored. This is tackled by studies by Lu [102] and Coolen [29], who have shown how to encode failure data with censorship for parameter estimation in a BN. Also, assets may be rated by multiple states. Drawing on Tsuda et al. [169], who introduced a Markov chain model to estimate multi-state failures using an MLE method, Han et al. [66] estimated the multi-state deterioration using a Bayesian estimation method modelled in the form of a Markov chain.

The ability to quantify uncertainty using expert judgement via the prior is another advantage of applying a BN. Coolen [29], Siu and Kelly [156] and Zhang and Mahadevan [195] presented examples of how to elicit priors for the parameters of statistical distributions. However, most studies focus the prior elicitation on a simple distribution like exponential or normal distribution. Not only because these parameters are easier to understand by engineers and to interpret from engineering knowledge, but also because some approaches are restricted to model simple distributions only, for example, as discussed in Section 2.3.2, a simple Markov model for multi-state deterioration assumes constant transition probability between states that follow an exponential distribution [19]. The exponential distribution is adequate in describing the early life of asset deterioration where the failure rate decreases with time but inadequate for describing the wear-out phase. In contrast, as shown by many studies such as Le [94], the Weibull distribution showed good fit in describing the lifetime distribution and is a better fit for the deterioration rate of infrastructure assets. The manipulation of the parameter shape enables Weibull distribution to specify a range of life phases in asset deterioration, including early life, useful life and wear-out life. Therefore, Weibull distribution is used in this thesis, although different distributions can be applied in our framework if necessary.

Though some works have been proposed to help experts understand the characteristics of a Weibull distribution (e.g. in Soliman et al. [158]), most of them only include generalised or simplified interpretations. A more comprehensive explanation of the parameters in a Weibull distribution for asset deterioration from various aspects is needed and is covered later in Section 4.1.3. To fulfil Objective I, we also show how to estimate the parameters of a Weibull distribution using a hybrid BN in Section 4.1.1 and to model the input data with censorship in Section 4.1.2. These steps are later extended to model multi-state deterioration in the form of a Markov model, where each transition is estimated by a unique Weibull distribution in Section 4.3.1 and components interact variously in Section 4.3.2 to fulfil Objective III.

Some asset deterioration studies extended the Bayesian parameter estimation approach with another level of parameters - hyperparameters, such as in Hong and Prozzi [70] and Andrade and Teixeira [4]. In the hierarchical BNs, assets are separated into different groups. Deterioration data within each group follows parameters of the group's distribution, while all the groups' parameters are governed by the parameters (that is, hyperparameters) of the whole population. However, most studies lack detail quantification and explanation about how to separate assets into groups, and how the parameters within each group are related. Also, they have not extended the potential use of a hierarchical BN in asset deterioration to individualise deterioration prediction by groups, which may provide a better prediction or learn between groups when some groups have fewer deterioration data. These potentials are addressed in Chapter 4 and validated in 5. In particular, in Section 4.2 we discuss how to infer parameters between similar assets or different but related assets using hierarchical BNs (Objective II). These models are further applied and validated with a real case study in Chapter 5 (Objective IV). This shows how to separate assets into groups by their features and how the deterioration of an individual asset can be predicted, notably in asset groups with limited data.

3.6.2 Inspection and Maintenance Decisions Support

Some studies have been proposed the use of BNs and causal reasoning to tackle maintenance problems. However, most early works focus on BN with only discrete variables. For example, Kang and Golay [76] reasoned about the states of a system after a maintenance action is selected using a discrete BN. Weber et al. [180] introduced a BN for diagnosis and prognosis of malfunctions in the manufacturing process to aid repair decisions. de Melo and Sanchez [33] considered factors such as the maintenance complexity and the expertise of professionals, that could induce uncertainty during the maintenance process. They presented a BN model to predict delays of a software maintenance project based on project features and experts experience. Rafiq et al. [144] developed a BN to predict bridge condition assembled from multiple components (see Section 2.3.3), but the conditions of components are given by experts rather than learned from data and knowledge. Discrete BNs are commonly used in an expert system for decisions making. To reason about the causal relationships between deterioration, learned from continuous statistical distributions as described above, and consequence of any maintenance, hybrid BNs with both discrete and continuous are most suitable [179].

With the development of inference algorithms for hybrid BNs(as discussed in Section 3.3), applications of them to reliability have been given more attention recently. Langseth and Portinale [89] discuss the properties of a modelling framework for Bayesian models with both types of variables and its application to reliability analysis. Some applications have been proposed. For example, system reliability is estimated in Marquez et al. [107] using component failure rates learned using different distributions and assembled in different configurations using a set of Boolean variables. In Han et al. [66] multiple exponential distributions were fitted in a hybrid BN for multi-state deterioration prediction and life expectancy analysis.

Meanwhile, models applied in other fields can inspire us to support a broader range of decisions. For example, the use of observational and intervention BNs (see Section 3.2.3) were recently employed to support physiological decisions [64], we can potentially apply them to support maintenance-related decisions, such as recommendation on repair action and evaluation of repair effects.

The maintenance modelling in this thesis consists of continuous variables from Bayesian statistical models combined with discrete variables commonly used in expert systems. This framework is used to create a series of decisions support models in this thesis (Objective V), including: condition prediction of multi-state assets in different system configurations (models proposed in Section 4.3 and applied in Section 6.1); inspection decisions based on asset condition prediction (applied in Section 6.2); suggested repair actions by reasoning about the anticipated repair effects (models proposed in Section 4.4 and applied in Section 6.3); maintenance planning of an asset over its life cycle including its deterioration and effects of repairs (models proposed in Section Section 4.5 and applied in Section 6.4).

3.6.3 Modelling of Complex and Large-Scale Systems

As described in Section 2.4.4, one of the challenges of maintenance modelling is the complexity in building a large-scale model need, for example, when a bridge is assembled from multiple components, and each component has its own deterioration model. As described in Section 3.5, some extensions of BNs have been proposed with advantages for organising complex models. For example, Weber and Jouffe [178] proposed a model of a complex system using an OOBN. A generic BN was developed to model the asset state. To model component with multiple states, multiple generic models were connected in the form of Markov chain using Dynamic BN (DBN, a BN that represents the evolution of state space model through multiple time slices). Each component was represented as an object in the OOBN. These components were linked in the OOBN to evaluate the system failure. Several ways to combine component failures were shown, including the logical combinations using in Fault Trees (using the representation of FTs as BNs proposed in Bobbio et al. [18]) as well as more complex ways where component failures propagate to a failure in the overall system. The modelling of dynamic OOBN shows a powerful way to represent a complex system in a compact and readable form. But, as the authors admit in their paper, further work is needed on the parameter learning of the deterioration of components. Another limitation arises from the inability (discussed in Section 3.5) to use an OOBN to model relations between varying objects.

Medina-Oliva et al. [114] tackled this limitation using a PRM and illustrated its application in dependability analysis. Medina-Oliva et al. [113] further extended the PRM to include the uncertainty from maintenance operations. But unfortunately, both of their works were built on discrete variables only. Lacking the use of continuous variables limited the functionality of the model. For example, their models cannot learn the rate of deterioration using Bayesian statistical models [178].

In this thesis, Chapter 7 develops a PRM-based framework for maintenance modelling in light of its advantages for modelling complex and large-scale system with varying relationships between objects. To fulfil the idea of OO design, inspired by the concept of fragments and idioms discussed in Section 3.5, Chapter 4 develops a set of generic BNs. These models can be used repeatedly, including models for Bayesian statistical models to learn deterioration rate and BNs for decisions support. Differing from Weber and Jouffe [178] who used DBNs to model asset condition from a sequence of model slice, where each slice represents a deterioration model between two consecutive states, Section 4.3.1 flattens all the deterioration models between different states in a standard BN. The presence of different deterioration models in a single model enables us to learn deterioration between states through the shared hyperparameters. It is useful in a situation like there are more deterioration data for some states or groups while other states or groups have less. In addition, Section 4.3.1 introduces the use of binary factorisation technique to resolve the inference complexity of the space explosion caused by the flatted model. Chapter 7 uses these generic models as a model library and describes how to use them in the framework developed as a PRM. To further fulfil Objective VI, this chapter also shows how to manage these model options from a non-statistician point of view to create maintenance models for different system specifications.

Chapter 4

Generic Bayesian Network Models for Bridge Asset Maintenance Modelling

Bridge asset maintenance modelling mainly involves the modelling of deterioration and reasoning of decisions. As discussed in Section 2.3.1, we can evaluate the rate of deterioration using a statistical distribution, which can be modelled in a Bayesian statistical model with a continuous variable. While decisions support modelling often requires discrete variables to model the relationships between events. Thanks to the development of approximate inference algorithms (see Section 3.3), for example, by combining dynamic discretisation with propagation algorithms on junction tree as implemented in software AgeanRisk, we can, therefore, perform inference in a hybrid BN with both continuous and discrete variables.

This chapter develops a range of generic Bayesian network models that can be altered or assembled to model various asset maintenance problems. It addresses Objectives I, II, III and partly V. Section 4.1 shows how to learn deterioration distribution from both data and knowledge. In particular, it shows how to encode various data types and how we could interpret engineering knowledge into mathematical parameters of a Weibull distribution. Section 4.2 presents hierarchical BNs that can be used to learn deterioration between different groups of assets. Section 4.3 shows how to use the models to predict asset condition, including asset with multiple states and asset condition is determined by conditions of its components. Section 4.4 gives two models to support the process of making repair decisions. Section 4.5 shows how to plan maintenance over a finite time horizon considering multiple intervention cycles. This chapter is further summarised in Section 4.6.

4.1 Deterioration Modelling

The expected lifetime of an asset is derived from its likely time to failure. The time to failure follows a statistical distribution. Based on the assumption made in Le [94], most infrastructures' deterioration rates (including infrastructure as a whole and its components) can be fitted with Weibull distributions. The two-parameter Weibull distribution (see Equation 2.4) is therefore applied in this thesis. This section builds a BN with parameters determined by a set of deterioration time data and prior knowledge of the parameters of the chosen distribution.

4.1.1 Learning Asset Deterioration with A Weibull Distribution

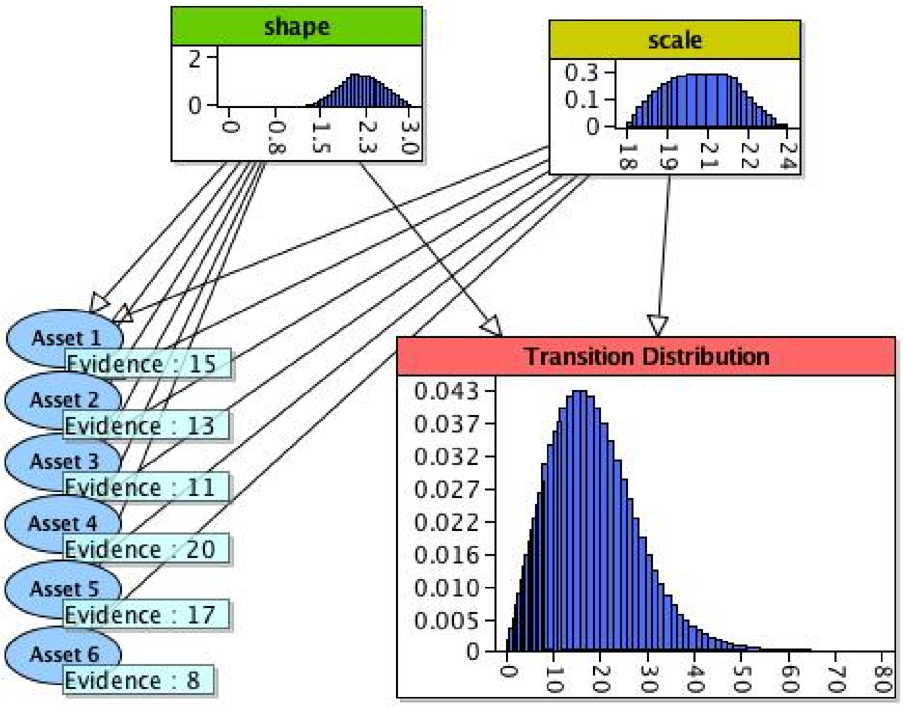


Figure 4.1 Parameter learning of a Weibull distribution.

In a Bayesian deterioration learning model, prior knowledge becomes part of the model; data on known failure times are entered, updating the distribution over the parameters. A Bayesian statistical model developed by Marquez et al. [107] can be modelled as in Figure 4.1. This is achieved by applying the Bayes' Theorem, which combines prior information of the parameters with data, to perform inference about the model. The posterior pdf of this Weibull distribution is:

$$f(\beta,\eta|\,Data) = \frac{L(Data|\beta,\eta)\varphi(\beta)\varphi(\eta)}{\iint_0^\infty L(Data|\beta,\eta)\varphi(\beta)\varphi(\eta)\,d\beta d\eta} \tag{4.1}$$

where $\varphi(\beta)$ is the prior of the parameter shape β and $\varphi(\eta)$ is the prior of parameter scale η. By observing the deterioration characteristics, experts can justify whether a type of asset has a decreasing, constant or increasing failure rate, leading to a range of possible values for the shape parameter. Similarly, it is also possible to evaluate the range of the scale parameter from the typical (mean) age of asset failure. Later Section 4.1.3 will illustrate how to elicit this knowledge in detail.

Different distributions, such as a normal distribution or a uniform distribution, could be used to express uncertainty over the range of each parameter as their prior distributions. Some experts find it easier not to specify their opinions with absolute precision but providing value intervals [148]. Following Marquez et al. [107], triangular distributions (Equation 4.2) are used in this model for its advantage in extracting expert knowledge into value intervals:

$$P(x) = \begin{cases} \frac{2(x-a)}{(b-a)(c-a)}, a \leq x \leq c \\ \frac{2(b-x)}{(b-a)(c-a)}, c < x \leq b \end{cases} \tag{4.2}$$

where $P(x)$ is the probability function of triangular distribution, a is its lower limit, b is its median and c is its upper limit. Experts assign values of a, b and c for each parameter, based on their experience.

$L(\beta,\eta)$ is the likelihood function based on Weibull distribution (equation 2.4) and Data $1,\ldots,n$:

$$L(Data|\beta,\eta) = \prod_{i=1}^{n} f(t_i|\,\beta,\eta) = \prod_{i=1}^{n} \frac{\beta}{\eta}\left(\frac{t_i}{\eta}\right)^{\beta-1} e^{-\left(\frac{t_i}{\eta}\right)^\beta} \tag{4.3}$$

Given Equation 2.4 and Equation 4.1, the posterior distribution of transition time T (node Transition Distribution) is:

$$f(T|\,Data) = \iint_0^\infty f(T|\,\beta,\eta)\,f(\beta,\eta|\,Data)\,d\beta d\eta \tag{4.4}$$

Figure 4.1 presents a BN constructed on these principles, and Table 4.1 lists its expressions. The time each asset transits from a normal to a failed state (the data) follows a Weibull distribution, and it is modelled as an observable continuous variable with an expression of Weibull distribution. This deterioration time is inferred from past transitions of assets in the same class. Take Asset 1 as an example: the observation 15 representing it takes 15 months for Asset 1 to transit from a normal state to a failed state. Because assets 1 to 6 are considered the same type of assets, we assume that they deteriorate following the same Weibull

Table 4.1 Expressions for the nodes of the BN in Figure 4.1.

Node Name	Expression
Asset 1 $\sim$ Asset 6	Weibull (shape, scale)
shape	Triangular (a1, b1, c1)
scale	Triangular (a2, b2, c2)
Transition Distribution	Weibull (shape, scale)

distribution, meaning that they share the same shape and scale. The posterior distribution of transition time from one state to another state is used to describe the deterioration process of assets in the same class. It is showed as the variable Transition Distribution in the figure, derived from the parameters learnt from data and knowledge. In the following subsections, we further introduce the modelling of data types and how to interpret parameters so that engineers may understand.

4.1.2 Modelling of Available Data Types

Often, the ideal data on the failure times of assets is not available. This section explores various uncertainty on the data likely to be available, showing how it can be used.

The transition time is the time, since the previous transition, when an asset transits from one state to another state. However, it is often hard to obtain this data: in practice, we are more likely to have data inferred from periodic inspections (and perhaps repairs) rather than data on the exact transition time (see Section 2.2.1). Four types of transition time data can be inferred from historical inspection records with uncertainty as follows. For illustration convenience, if unspecified, the inspection interval is assumed to be 12 months, and the transition T are between working and failed in this subsection:

- **Left-censored data:** the asset failed at some point before we started to inspect. For example, asset 7 failed in its first inspection after it was built, that means the transition time is less than 12 months: $T < 12$.
- **Interval-censored data:** failures happened sometime between two inspections. For example, in the fourth inspection, asset 8 did not show any signs of deterioration, but we found out it failed in the fifth inspection. Therefore, we can conclude that the asset transitioned between 48 and 60 months: $48 < T < 60$.
- **Right-censored data:** for those cases where the asset survived longer than the time available for observation. Suppose we decide to terminate the use of a working asset

(asset 9) that has been inspected sixth times and has survived for more than 72 months, hence, $T > 72$.

- **Exact-time data:** this type of data may be available when an issue is reported. For example, at 20 months, asset 10 failed suddenly, and inspection confirms this transition: $T = 20$.

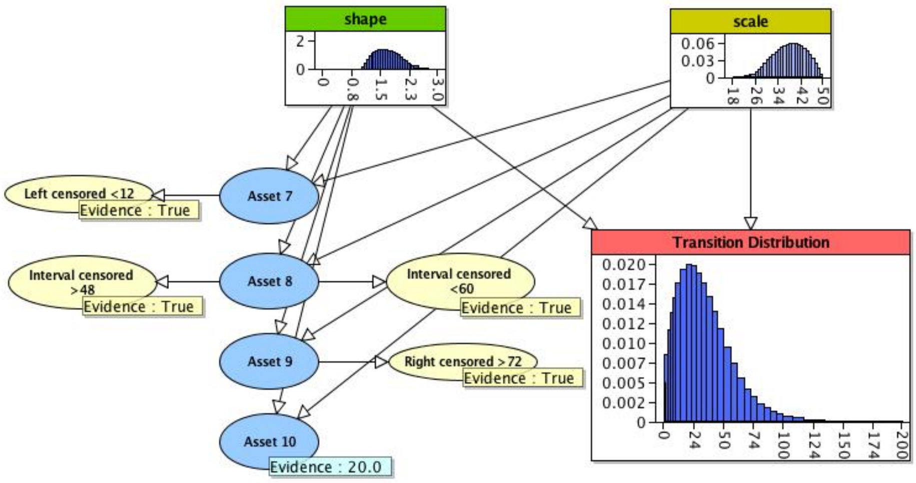


Figure 4.2 Modelling of available data types.

For exact-time data, we can directly observe the node with the failure time as shown in Figure 4.1 (asset 10). But for censored data, a Boolean variable is introduced to express a constraint on an asset's transition time. Figure 4.2 shows how to model data with censorship using the examples discussed above. To represent a left-censored data, for example, less than 12 months, the variable is expressed with a logical expression $if(T < 12$, "True", "False"), and the **True** state is observed (asset 7). Similar constraints are used for right censored data (asset 8). For interval-censored data, two Boolean nodes are built applying the same principles (asset 9).

Furthermore, for asset rated with multiple states, there is a possibility that an asset deteriorates faster than our observations. For example, suppose that at the 12-month inspection, a component remained at State 3, while in the 24-month inspection, the component was found in State 1. To use this type of information, we can enter observations for right censored data that its first transition time T_1 is greater than 12 months, and left censored data that its second T_2 is smaller than 24 months, as well as an additional constraint that the first transition time T_1 is smaller than the second one T_2, as $(\mathrm{T}_1 > 12) \wedge (\mathrm{T}_2 < 24) \wedge (\mathrm{T}_1 < \mathrm{T}_2)$.

4.1.3 Prior Knowledge Elicitation of Weibull Distribution

To use a statistical distribution to describe the rate of deterioration, prior probability distributions of parameters, such as the Weibull's shape and scale, need to be assigned. It is difficult for non-statisticians to evaluate the values of shape and scale directly, but can be made easier by understanding the characteristics of the distribution and trends of the parameters. This subsection focuses on the interpretability of the traits of parametric statistical distributions using the example of a Weibull distribution.

Shape Parameter

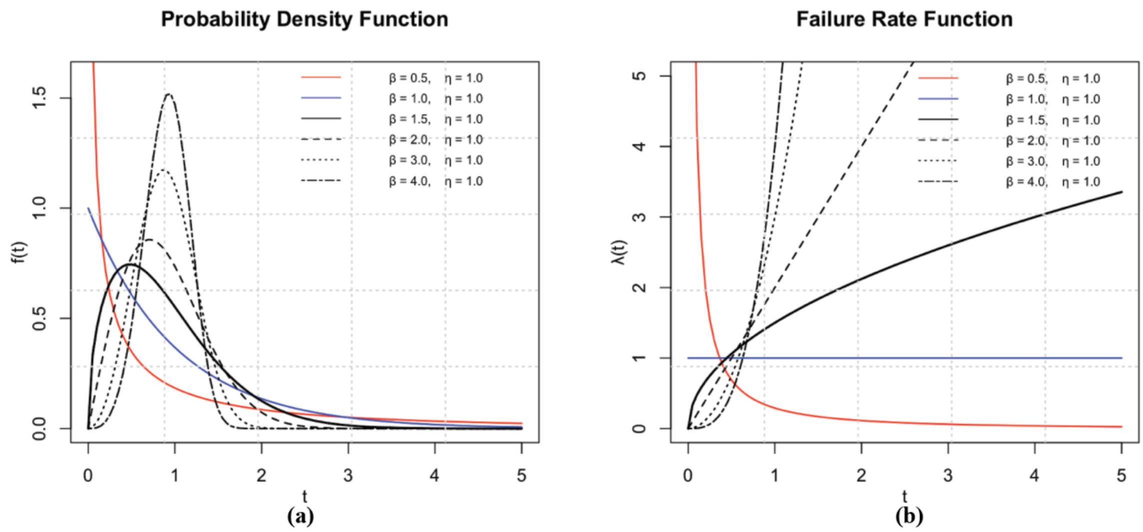


Figure 4.3 pdf and failure rate function of Weibull distributions with different shapes.

Figure 4.3 shows examples of the Weibull distributions' pdfs with different shape parameter β. Empirically, the shape parameter of Weibull distribution offers us great flexibility in modelling a variety of distribution with several physical behaviours: it becomes an exponential distribution when β equals to 1, a Rayleigh distribution when β equals to 2. Also, studies have shown that the skewness of Weibull distribution has a strong relationship with the value of β [62]. The skewness decreases with β: it has a positive skewness to the left side of the pdf when β is smaller than 3.6, and it approximates symmetrically as a normal distribution when β is near 3.6 with a skewness value approaching 0, and it skews to the right with a negative skewness when it is greater than 3.6 [110].

To understand its characteristics more intuitively, we have its cumulative distribution function (cdf), which is also known as the distribution's unreliability:

$$F\left(t|\ \beta,\eta\right)=\int_{0}^{\infty}f\left(t|\ \beta,\eta\right)dt=1-e^{-\left(\frac{t}{\eta}\right)^{\beta}} \tag{4.5}$$

While the survival function (reliability) of a distribution is the complementary of its cdf, representing the probability of the asset will survive after a given time:

$$S\left(t|\ \beta,\eta\right)=Prob\left[T>t\right]=\ 1-F\left(t|\ \beta,\eta\right)=\ e^{-\left(\frac{t}{\eta}\right)^{\beta}} \tag{4.6}$$

By representing Event A a situation where the asset will fail between a small enough interval between t and $t+\Delta t$, and Event B a situation where the asset will survive after time t, using the law of conditional probability, the probability of Event A given Event B is:

$$P\left(A|B\right)=\frac{P\left(A\cap B\right)}{P\left(B\right)}=\frac{f\left(t|\ \beta,\eta\right)dt}{S\left(t|\ \beta,\eta\right)}=\lambda\left(t|\ \beta,\eta\right)dt \tag{4.7}$$

, thus, we have the instantaneous failure rate function at any time point t:

$$\lambda\left(t|\ \beta,\eta\right)=\frac{f\left(t|\ \beta,\eta\right)}{S\left(t|\ \beta,\eta\right)}=\frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1} \tag{4.8}$$

As a natural extension of exponential distribution (when shape $\beta=1$), Weibull has a polynomial failure rate with an exponent $(\beta-1)$. The characteristics in the failure rate function are useful to help experts define the priors of the parameters. The corresponding failure rate functions of the distribution in Figure 4.3 (a) are showed in Figure 4.3 (b). A β value that is between 0 and 1 describes a decreasing failure rate over time. This often happens when an asset is in the burn-in phase that shows early degradation, which may be caused by a problematic building process or infrastructure operation. In the case where we believe the asset's failure rate would not change over time, that is, it has a constant failure rate, we can suggest β equals to 1. When describing infrastructure in a wear-out failure phase, that is, the probability of an asset failing increases over time, we can recommend the prior of β greater than 1 indicating an increasing failure rate.

Variance

By looking at Figure 4.3, we can also notice that with the increase in shape parameter β, the pdf gets narrower, which represents a smaller variance. The variance σ^2 of a Weibull distribution is:

$$\sigma^2=\ \eta^2\left[\Gamma\left(1+\frac{2}{\beta}\right)-\left(\Gamma\left(1+\frac{1}{\beta}\right)\right)^2\right] \tag{4.9}$$

validated in Abramowitz and Stegun [1], the difference between $\Gamma(1+\frac{2}{\beta})$ and $(\Gamma(1+\frac{1}{\beta}))^2$ becomes smaller with the increase in β, that is, the variance σ^2 decreases with the increase in β. When $\beta \to \infty$, the variance of the Weibull distribution approaches 0.

Scale Parameter

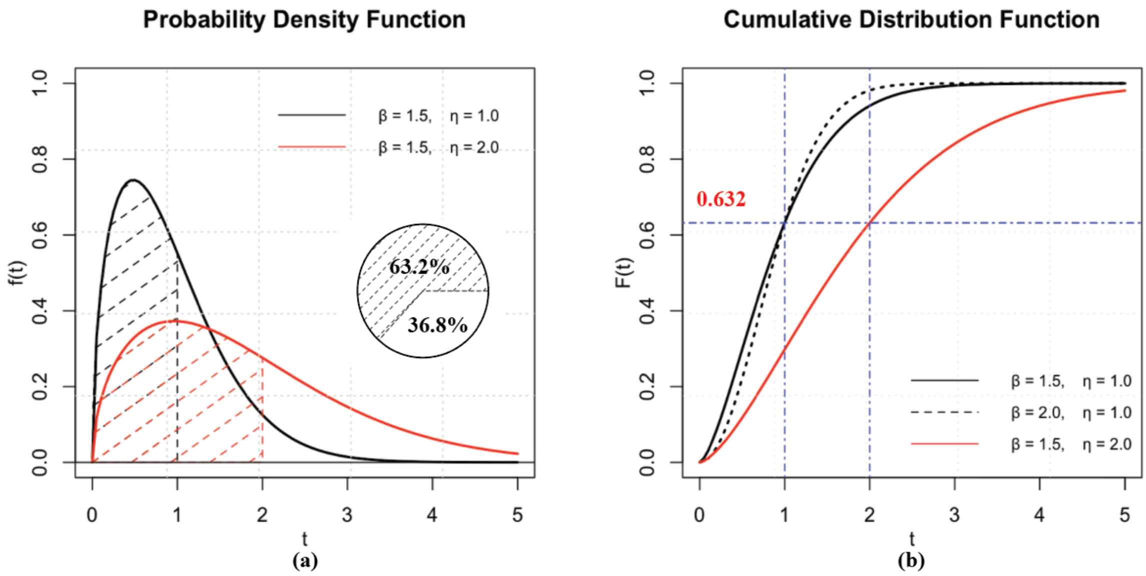


Figure 4.4 (a) pdf of Weibull distributions with different scales; (b) cdf of Weibull distributions with different shape and scale.

By holding the same shape β, increases the scale parameter η has an effect of stretching out the pdf, this can be seen in Figure 4.4 (a): with the increase in η, the range of the pdf gets wider with a lower crest. The quantiles of the distribution can explain this, given the cdf in Equation 4.5, by setting $F(tp) = p$, we have:

$$t_p = \left\{ ln \left(\frac{1}{1-p} \right) \right\}^{\frac{1}{\beta}} \eta \tag{4.10}$$

As shown by the red and black solid lines in Figure 4.4 (b), having the same shape β, it takes longer to reach the same quantile with a higher scale η. Additionally, when time to failure t equals to η, from the cdf in Equation 4.5, we have $1 - e^{-1} \approx 0.632$, that is, as shown in the shaded area in Figure 4.4 (a) and (b), regardless the value of β, when time t equals to η, 63.2% of the population will fail. Together with the skewness information of a Weibull distribution mentioned in the shape parameter, this can give experts confidence to estimate the scale parameter η given knowledge of the average failure time.

Also, given the mean of Weibull distribution μ is,

$$\mu = \eta \Gamma \left(1 + \frac{1}{\beta}\right) \tag{4.11}$$

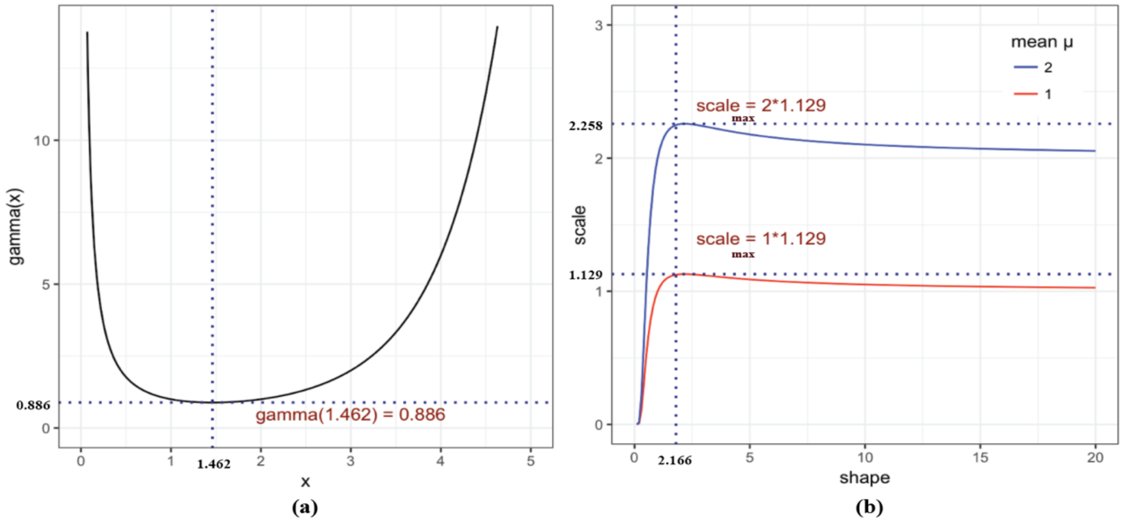


Figure 4.5 (a) gamma function; (b) mean μ of Weibull distributions.

we have, $\frac{\eta}{\mu} = \frac{1}{\Gamma(1+\frac{1}{\beta})}$. Given $\beta \geq 0$, we have $1 + \frac{1}{\beta} \geq 1$. From Figure 4.5 (a), we can see that for a gamma function $\Gamma(x)$ with $x \geq 1$, it has a minimum value at $\Gamma(1.462..) \approx 0.886$. Therefore, the maximum of $\frac{\eta}{\mu} \approx \frac{1}{0.886} \approx 1.129$, and from $1 + \frac{1}{\beta} = 1.462$, we can obtain the corresponding shape $\beta \approx 2.166$. This can be interpreted as the maximum of scale is 1.129 times of the mean of the Weibull distribution μ. This validates the examples in Figure 4.5: given a mean of the Weibull distribution μ, with the increase in shape value β, the scale η reaches its peak when the value of shape at around 2.166, which is about 1.129 times of its mean μ. And after that point, the value of η decreases slightly and gradually stabilised reaching its mean μ when $\beta \to \infty$. By knowing this, experts can form an idea of the relationship between the value of scale and mean time to failure, and provide more confidence for them to extract knowledge for the prior of scale.

Summary

Explaining the characteristics of the Weibull distribution in a way that engineers can understand is a vital process in order to quantify prior knowledge from experts. Instead of asking their opinions about the shape parameter β directly, we may interpret the question into the form of failure rate. For example, with the increase in time, does the asset get more likely to fail? If the answer is yes, we can define a prior for β that is greater than 1.

Also, we assume experts are more likely to have knowledge such as the mean failure time of this type of asset μ, which can be used to narrow down the range of β and estimate the mean of the scale parameter η. Given the μ, in the overall population distribution of this type of asset, whether the assets are more likely to fail before μ or after? If the answer is before μ, we can interpret it as a left-skewed distribution, which represents the β is ranged from 1 to 3.6. This is also validated by most studies, where most infrastructure's failure time follows a β between 1 to 3.6 (see examples in Le [94] and Nasrollahi and Washer [122]). To further narrow down β, we can increase its left bound if the experts are very confident in providing the above information. This is contributed by the information that the higher β is, the smaller variance σ^2, as discussed before.

Given the information of the μ and the skewness, and the interpretation that the value of scale parameter η represents around two thirds (63.2%) of the failure population, we can estimate the mean value of η is slightly higher than μ. Also, given the maximum of the scale η is 1.129 times of its μ, we can use it to set the upper bound of η.

4.2 Individualised Deterioration Learning from Similar Assets

In practice, we may have several groups of assets of different types, which we believe deteriorate with similar behaviour or different but related. Model in Section 4.1 is extended with another layer of parameters as a hierarchical BN to learn between groups. Since the failure times are determined by the parameters shapes and scales, we assume the distribution's parameters learnt for one type of asset share some similar deterioration characteristics with differences with other types. This section introduces how to learn between groups and individualise the deterioration learning with similar or different but a related rate of decay.

4.2.1 Assets with Similar Deterioration Rate

This subsection shows how to use the learned deterioration of a dominant group (data rich) to infer the learning of weak groups (data poor). It is suitable for a situation where assets deteriorate similarly, and experts are confident about the difference between groups. For example, it can be used for a case where there are only a few influencing factors on asset deterioration, or there is a dominant feature has been identified with a significant impact on the rate of deterioration. We do this by assuming the parameters learned for one group approximate those of the other similar types, with differences inferred by their hyperparameters.

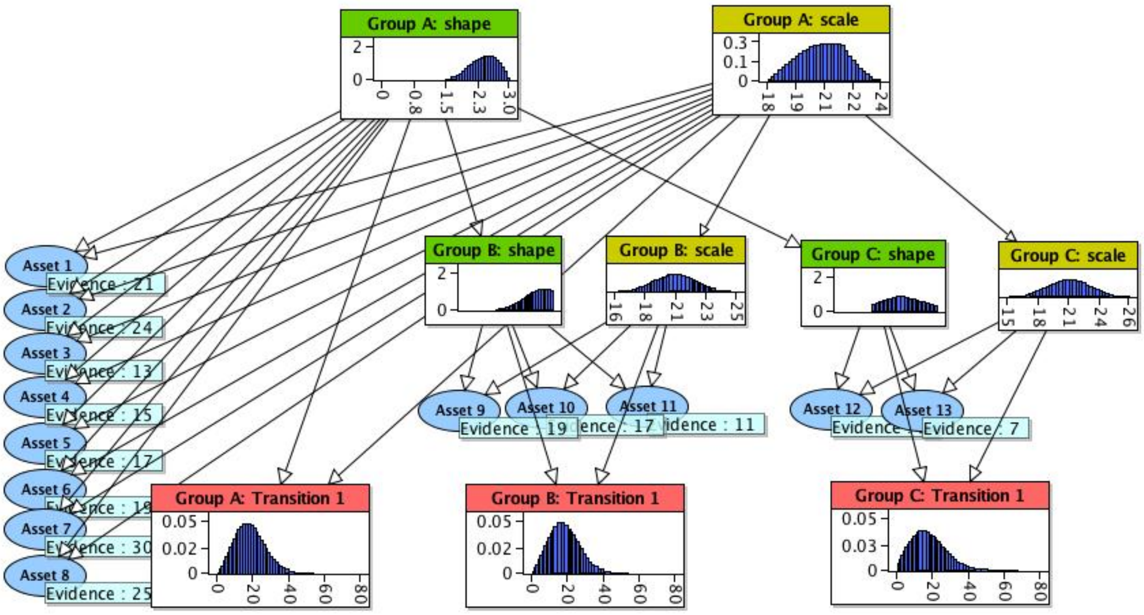


Figure 4.6 Multiple asset groups with similar deterioration.

Assume assets in Group A (assets 1 to 8 in this example), Group B (asset 9 to 11) and Group C (asset 12 and 13) have similar deterioration rates resulting from some shared characteristics (such as similar designs with different materials). Group A has more failure data (dominant group) than Group B and C (weak groups). In the case that one group has more failure data than the other groups, we may leverage the deterioration learning from one group to other groups. Figure 4.6 shows a hierarchical BN for these three groups of assets. Shapes (node Group B: shape and Group C: shape) and scales (node Group B: scale and Group C: scale) of these assets were governed by the Group A's shape (node Group A: shape) and Group A's scale (node Group A: scale) variables, whose prior probability distributions are using triangular distributions as suggested in Section 4.1.

Experienced experts may have knowledge about the typical deterioration behaviour of assets in Group A since it has more failure data to observe. Assume assets in Group A are concrete-based bridge decks, with the knowledge elicitation as discussed in Section 4.1.3, the experts can express the prior knowledge for parameter shape β and scale η. For example, for scale η in Transition 1 of a typical concrete-based bridge deck, experts estimate it follows a triangular distribution with a lower bound of 10 years, a middle of 23 years and an upper bound of 30 years. This knowledge becomes the prior of the hyperparameter in this model, which is learned from all assets in Group A.

At the same time, experts may know how similar two groups of assets are. For example, experts know about the deterioration of a concrete-based deck (Group A) is more similar to a stone-based deck (Group B) compares to a timber-based deck (Group C). This knowledge leads to a higher similarity degree (lower variance) between stone-based (Group B) and concrete-based structure (Group A). A doubly truncated normal (TNormal) distribution (its expression can be found in Fenton and Neil [45]), a normal distribution bounded by lower and upper limits is used to model the relationship between the local parameters and the global parameters (hyperparameters):

$$\text{Group B (C): shape (scale)} \sim \text{TNormal}(\mu, \sigma^2, L, U)$$

The mean μ of this distribution is the shape or scale from the rich data group, and variance σ^2 representing the degree of similarity between these two groups, which is given by experts. The lower bound L and upper bound U of the distributions are also evaluated by experts about extreme values.

Take node Group B: shape as an example: assumes it has a conditional probability distribution given by TNormal (Group A: shape, 0.5, 1, 3). Its mean is given by the distribution of node Group A: shape, which inherits the typical behaviour of the shape between assets in Group A. Its variance is 0.5 – a smaller variance means a higher similarity, representing a high degree of similarity between these two groups (Group A and B). The distribution is restricted to the region between 1 and 3, indicating it has an increasing failure rate (because the shape value is higher than 1 as discussed in Section 4.1.3) with values between 1 and 3. Similarly, for shape in Group C, experts know that a timber-based deck (Group C) may deteriorate faster than a concrete-based deck (Group A), we can assign the μ of the TNormal distribution with an additional arithmetic value to indicate this behaviour. By extracting information from experienced experts about the degree of similarity and the differences between groups, the model offers to reason parameters of a group with only a little data (Group B and C) using data from another group (Group A) that are judged to share a similar deterioration rate.

4.2.2 Asset with Different but Related Deterioration Rate

As discussed in Section 2.2.2, asset feature can be used as an indication of their deterioration characteristics. This subsection introduces the use of feature space to distinguish individual asset into groups for individualised deterioration learning. It also shows how to aggregate the feature values as an indication on the deterioration rate, which can be used to quantify the different but related deterioration between groups, and further extend the model from the last subsection to learn between groups. This model is suitable for situations when experts find it difficult to quantify the relationship of asset deterioration behaviour between groups directly by themselves.

The deterioration rate may be affected jointly by different features, like heavy loading and aggressive environment conditions (see an example in Yianni et al. [188]). Ideally, the maintainers' knowledge of these effects could be combined with statistical failure data gathered from a population where the loading and environment vary. From a decision support perspective, this will allow specific assets to be distinguished. For example, Marsh et al. [109] outlined a Bayesian architecture to integrate multiple factors, such as loading and environmental stress, to support decision but it did not show how failure data could be included.

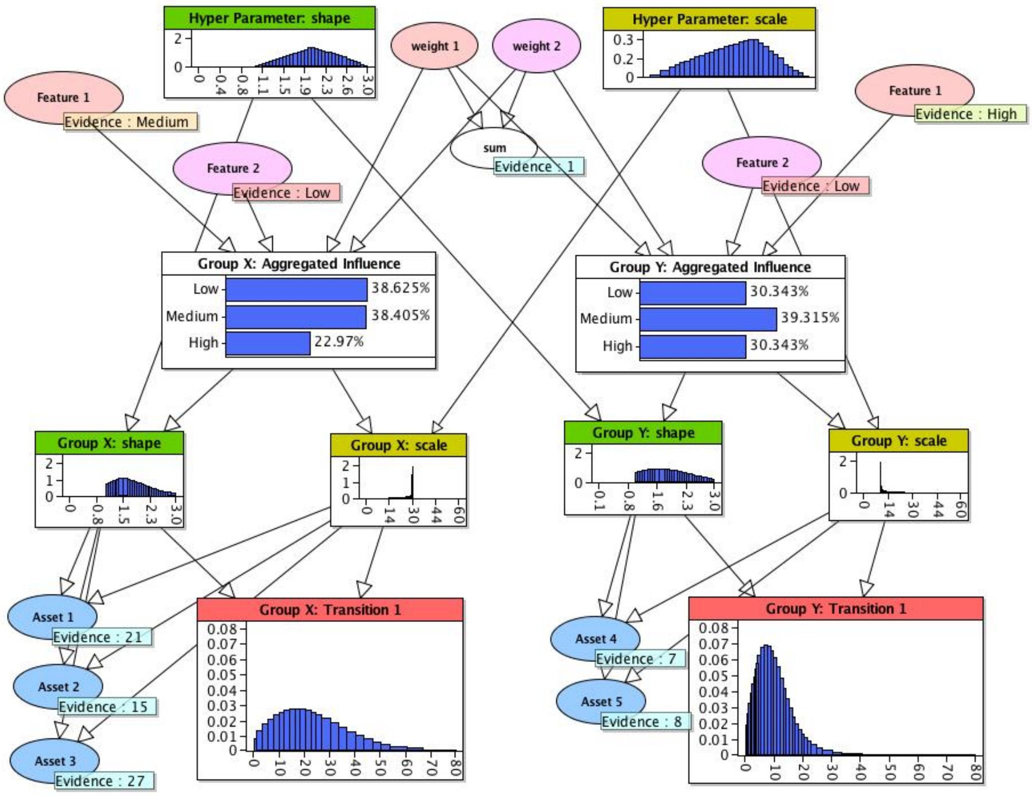


Figure 4.7 Aggregated influence on deterioration from features.

The degree of influence of each feature should be rated so that we can distinguish individual members of assets into suitable groups. Ranked node (see Section 3.2.2) is applied to model each feature. Its rating is ranging from low, medium to high. The higher degree it is rated, the faster it deteriorates, therefore, the shorter the transition time is. As shown in Figure 4.7, two features are modelled for illustration purpose (here, for example, loading as Feature 1 and environmental condition as Feature 2). An example of how to estimate their states can be adopted from Yianni et al. [188]: take loading of railways bridges as an example, track data of Equivalent Million Gross Tonnes Per Annual (EMGTPA) passes over the bridge can be used to estimate the level of loading. For loading less than 3.5 EMGTPA, they are rated as low, between 3.5 and 12 EMGTPA are classified as medium, and over 12 EMGTPA are defined as high. In Figure 4.7, a medium loading is observed for Feature 1 in Group X because the EMGTPA of the line passes over this bridge is between 3.5 EMGTPA and 12 EMGTPA. Assets with the same combination of feature values can be grouped into the same group assuming they have the same deterioration characteristics. While for assets within different groups, their feature values can be an indication of how similar their deterioration is. By doing this, we can separate assets into two groups in this example: for assets with medium rating in Feature 1 and low in Feature 2, they are grouped as Group X (asset 1 to 3); for assets with high rating in Feature 1 and medium in Feature 2, they are grouped as Group Y (asset 4 and 5).

At the same time, different features may have different strength in influencing the deterioration of assets. For example, the environmental condition may have more influence on the deterioration of a metal bridge than its service type. Experts could have knowledge about the weights of these influence factors and assign them directly. In the case where we lack this type of knowledge, we can assign a hyperparameter for each feature represents its weight (node weight 1 and weight 2). It is modelled with a prior of a uniform distribution with a lower bound of 0 and upper bound of 1. All the weights of the feature are linked together with a node that sums the weights to 1 (node sum). The weights converged and learned where there are many different groups of assets (with varying combinations of the features) with failure data. However, if we do not have many groups, due to the convergence, it is better to rate weights by experts or to give more informative priors for the weight nodes, rather than learn them. Note that in Figure 4.7 the weight nodes are created for demonstration purpose to show how they are built and linked. But in this case, it is better to assign weights directly from experts since there are only two groups.

The degree of influence factors (node: Aggregated Influence) is modelled by a TNormal distribution combined using a weighted mean (wmean, equivalent to a linear model) of the influence factors (node Feature 1 and node Feature 2), and variances σ^2 are given by experts regarding their confidence level of the weights:

$$\text{Aggregated Influence} \sim \text{TNormal}(\text{wmean}, \sigma^2, 0, 1)$$

For example, experts can assign the weight of loading with 0.3 and environmental stress with 0.7, with a variance of 0.2 as a slightly high uncertainty about these weights so that the combined influence of these factors is only slightly closer to the value of environmental stress than the value of loading. Since there are only two groups built in the example in Figure 4.7 with little failure data, the posterior of weights will not converge, resulting in both features carrying approximated equal weights. In Group X, though neither features were rated as high, there is still a 22.97% in high influence since the variance is set as 0.2 with high uncertainty.

Developed from the model in the last subsection, here a case where both groups have little data is presented. Therefore, instead of using the parameters of a dominant group as hyperparameters for weak groups, hyperparameters of the overall population that govern all the subgroups are used. These hyperparameters represent the typical deterioration behaviours of assets, for example, the typical rate of bridge deterioration. Meanwhile, the parameters within each group represent the deterioration behaviours of each subgroup adjusting by their aggregated influence resulting from features. The parameter is partitioned modelled by three TNormal distributions (since there are three states in this example: low, medium, high, each state is modelled by a TNormal distribution), with variance and the bounds are given by experts (same as the last subsection). The only difference is the mean, which is adjusted from the typical hyperparameter and aggregated influence. The mean based on the states of the aggregated influence degree: a low influence degree means a longer transition time, therefore it has a higher probability in the high-value region of the hyperparameter; in contrast, a high influence degree is mapped to the low-value region of the hyperparameter. The evaluation of regions is given by experts regarding how easy the assets can be influenced by external factors.

4.3 Asset Condition Prediction from Learned Distributions

The BNs of the previous sections cover the distribution of transition time learned about its deterioration. We can extend it to predict the state of an asset based on a future inspection time, allowing decisions about, for example, inspection and maintenance reasoning.

Figure 4.8 shows how the prediction works. We denote Inspection Time as T, Transition Time from Working to Fail as $T_{Working \to Fail}$, the predicted state node is modelled by a Boolean node with a logical expression: $if(T < T_{Working \to Fail}$, "Working", "Fail"). Assume the asset is currently working, this example aims to predict its condition (either working or fail) 24 months later. With a transition distribution learned from failure data (models from the last section and Equation 4.4), as shown in the figure, 43.213% of its pdf is smaller than 24 months (showed as red area). It means there is a 43.213% probability this asset will fail (transit to another state). Hence, 56.787% will survive (does not transit).

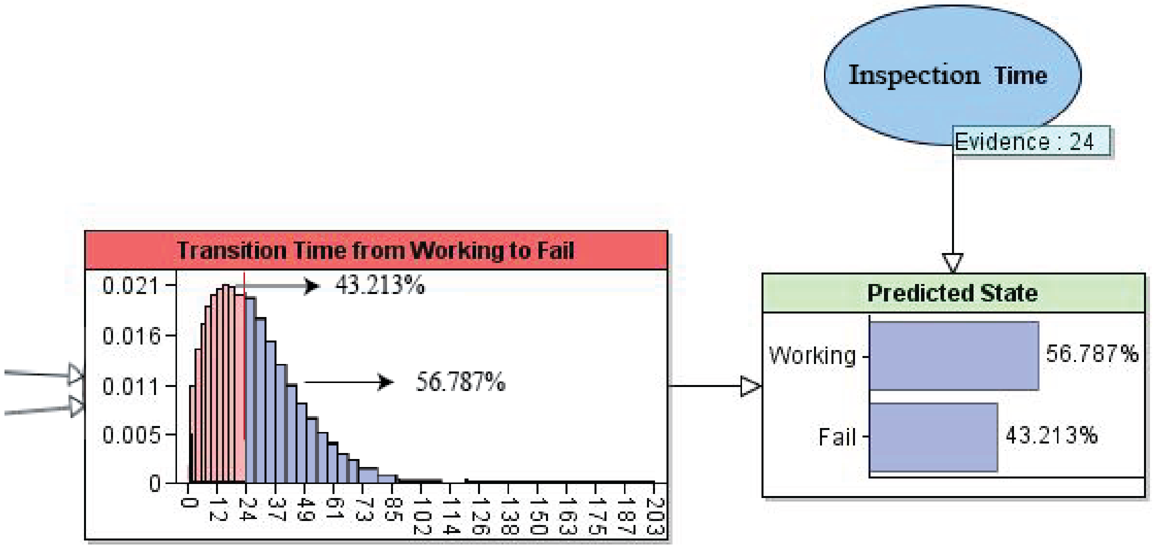


Figure 4.8 Condition prediction of a binary state asset after 24 months.

This model is further developed in the following subsections to predict the condition of an asset that is rated by multiple states and to predict the condition of an asset that is affected by its multiple components.

4.3.1 Asset with Multiple States

Distinguish ageing processes of assets directly only working from (hard) failure is not enough for making decisions about a variety of maintenance actions. Most critical infrastructures, their inspection data and repair decisions are made based on a grading system representing different levels of deterioration (see Section 2.3.2). For which, to predict their conditions, we need to model deterioration between multiple states.

To model the deterioration of assets through several states, the model in Figure 4.8 is extended with multiple states showed in Figure 4.9. Here we model the condition of an asset with a three-state scale using a categorical variable (named labelled node in AgenaRisk [2]), with the transition from Good to Fair ($T_{Good \to Fair}$), Fair to Poor ($T_{Fair \to Poor}$). Each transition is modelled by a separate parameter learning model. We assume deterioration progresses with the sequence of the rating system, that is, the transition from Good to Poor must go through the transition of Fair. Hence, we denote Inspection Time as T, in this example, the expression for the predicted state node becomes $if(T < T_{Good \to Fair}$, "Good", $if(T < T_{Fair \to Poor}$, "Fair", "Poor")).

Assuming the starting state of this asset is a Good condition, the query from Inspection Time will first visit the Transition Time from Good to Fair. In the pdf of this node, only

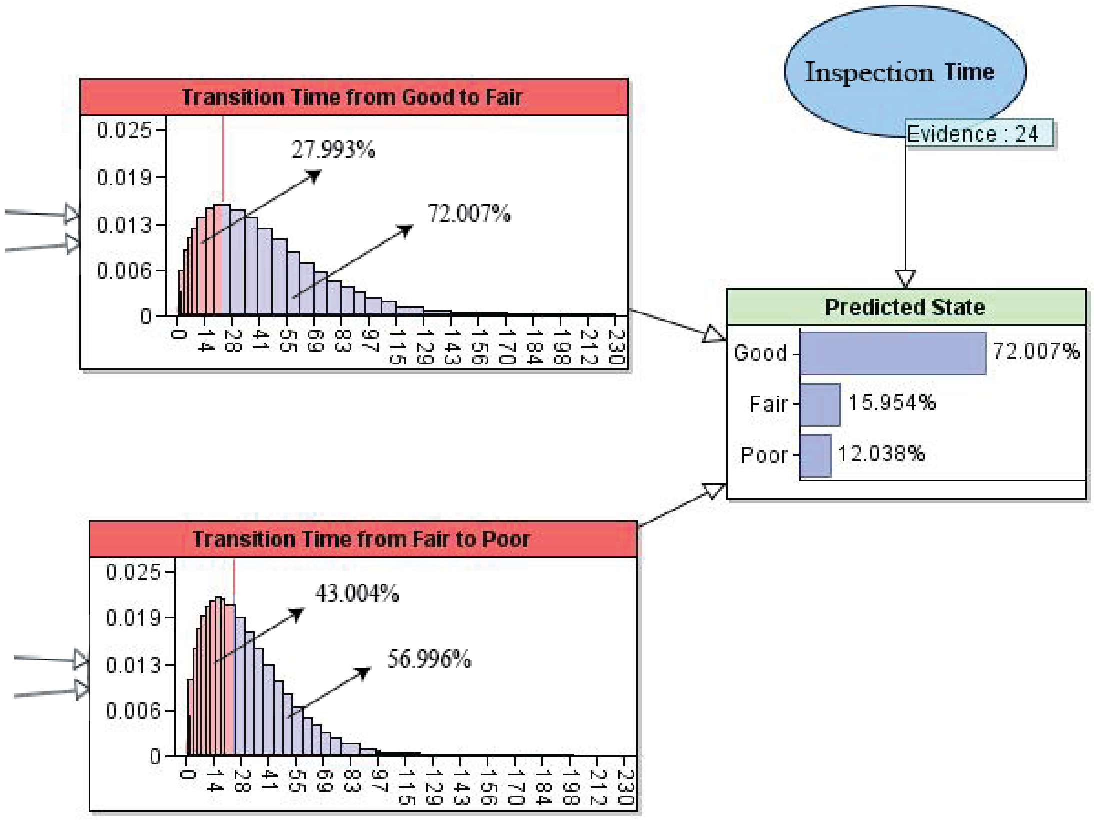


Figure 4.9 Condition prediction of a multi-state asset after 24 months.

27.993% of the area of this distribution is smaller than 24 months, that means, 72.007% of this asset will still stay at Good state. For those transit to Fair, only 43.004% will further transit to the Poor state showed in Transition Time from Fair to Poor, that is, 27.993%*43.004% = 12.038%. Therefore, we have the deterioration prediction distribution as shown in node Predicted State.

This is a simple example with only three states, but with the increase of the number of states, the number of parent nodes for Predicted State node increases correspondingly. In a BN, a node that has too many parents results in a very large Conditional Probability Table (CPT), which leads to computational complexity for its inference. Especially for continuous variables like transition time distribution, this complexity increases sharply since dynamic discretisation inference algorithm will discretise each distribution into dozens of states. If, say 30 states were discretised for each transition, if an asset is graded by a 10-point scale like the NBI dataset (see Weseman [182]), the predicted state would have 9 transition parents that require a CPT of 30^9 entries.

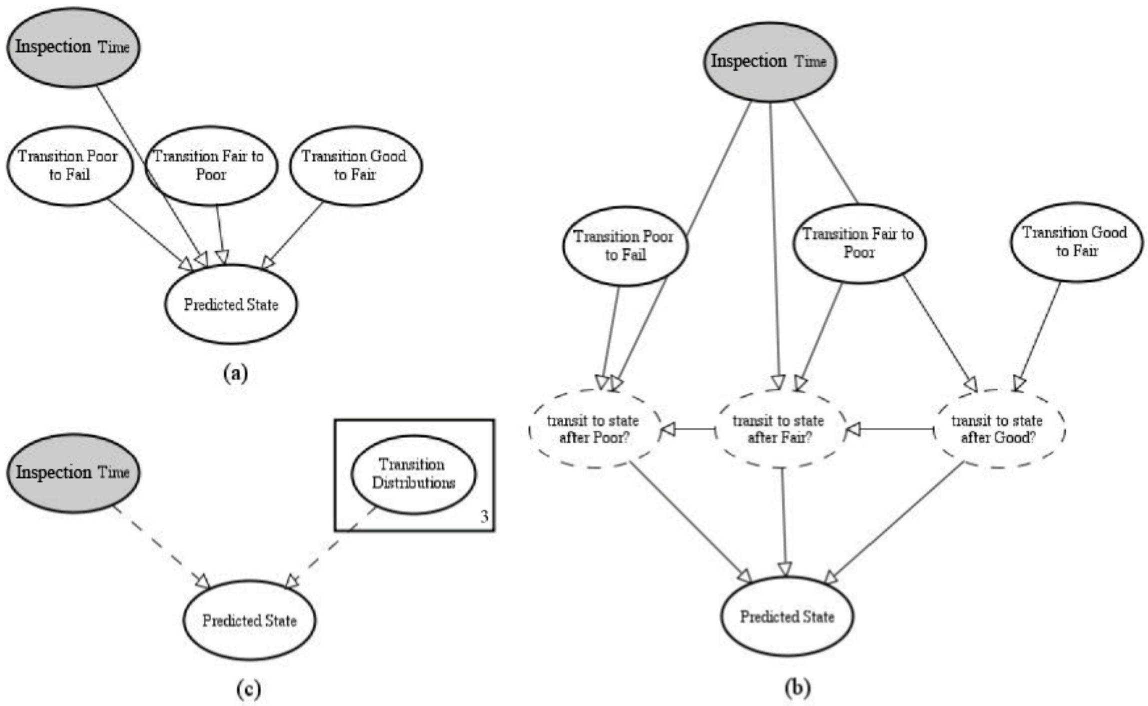


Figure 4.10 Condition prediction of a multi-state asset: (a)model with aggregation; (b)binary factorised model; (c)plate model.

To reduce the inference complexity and consequently speed up the calculation, the binary factorisation modelling technique proposed in Neil et al. [123] is adopted in this thesis. This technique converts the aggregation structure into a binary tree structure (for example, from Figure 4.10 (a) to (b)) that aims to ensure all nodes in the BN have no more than two continuous nodes as parents. Among the variables in the model, the inspection time node and all transition distribution nodes are continuous variables. Therefore, for the first transition, we build a temporal Boolean node (showed in dotted line) to query if the asset transits to worse states given the inspection time (same function as the predicted state node in a binary system as in Figure 4.8). The following temporal Boolean nodes have the same structure, only with an additional link from the previous temporal node in the form of a Markov model. It will transit to worse states only if the previous state is transited to the current state and current transition is smaller than the inspection time. For example, we denote node transit to state after Good as t_{Good}, node Transition Fair to Poor as $T_{Fair \to Poor}$, Intervention time as T, the second temporal transition node is true when $(t_{Good} = True) \wedge (T > T_{Fair \to Poor})$. At last, all the temporal nodes are linked to the predicted state node. By doing so, it creates a much smaller CPT than the one with the aggregation structure: these temporal nodes are discrete variables, and each temporal node only has two states. We denote the rating system with n

states, the binary factorised model reduces the size of an aggregated structure from $30^{(n-1)}$ to $2^{(n-1)}$. In the example in Figure 4.10, it reduces from 30^3 to 2^3, and for the NBI example, it can reduce the space from 30^9 to 2^9.

This modelling technique is employed in the rest of the thesis. Figure 4.10 (c) gives an example of how to represent the model in (b) using a modified plate model to make the models more readable: a square indicates that there are multiple variables inside the square with the same structure. The index shows the number of variables with the same structure. The temporal nodes are omitted but represented by the dotted lines, indicating the model is binary factorised.

4.3.2 Assets Assembled by Multiple Components

The condition of an asset with multiple components can be evaluated in various ways depending on the type of interaction between components, including in parallel, series, bridge structure, and their variants (see Section 2.3.3). Dynamic Fault Trees (DFTs) is one of the most notable frameworks in modelling a flexible system configuration [108]. Here, we represent the DFTs in the form of event-based Bayesian structure to model parallel and series system, and bridge structure is modelled with a similar structure but with a specific function. Each transition distribution of each elementary component of the system is represented by a continuous random variable. Each transition of the system is connected by several components' corresponding transitions and characterised by an arithmetic function. The state of the asset given a time instance is modelled with a discrete random variable using the models we described in the last subsection. Since the resulting BN is a hybrid BN with continuous variables, we use dynamic discretisation for its inference.

To specify the joint distribution of the BN, we need to assign the marginal (prior) probability density functions of the basic events (transition distributions of components) and functions of intermediate events (transition distributions of asset system). In the case where deterioration distribution between components is statistically independent, we can define them by their marginal probability distributions. By using the models described in Section 4.1 and 4.2, we can learn these transition distributions from deterioration data and knowledge. The function of an intermediate event is characterised by its parents and its fault tree construct. These functions are illustrated with the examples showed in Figure 4.11.

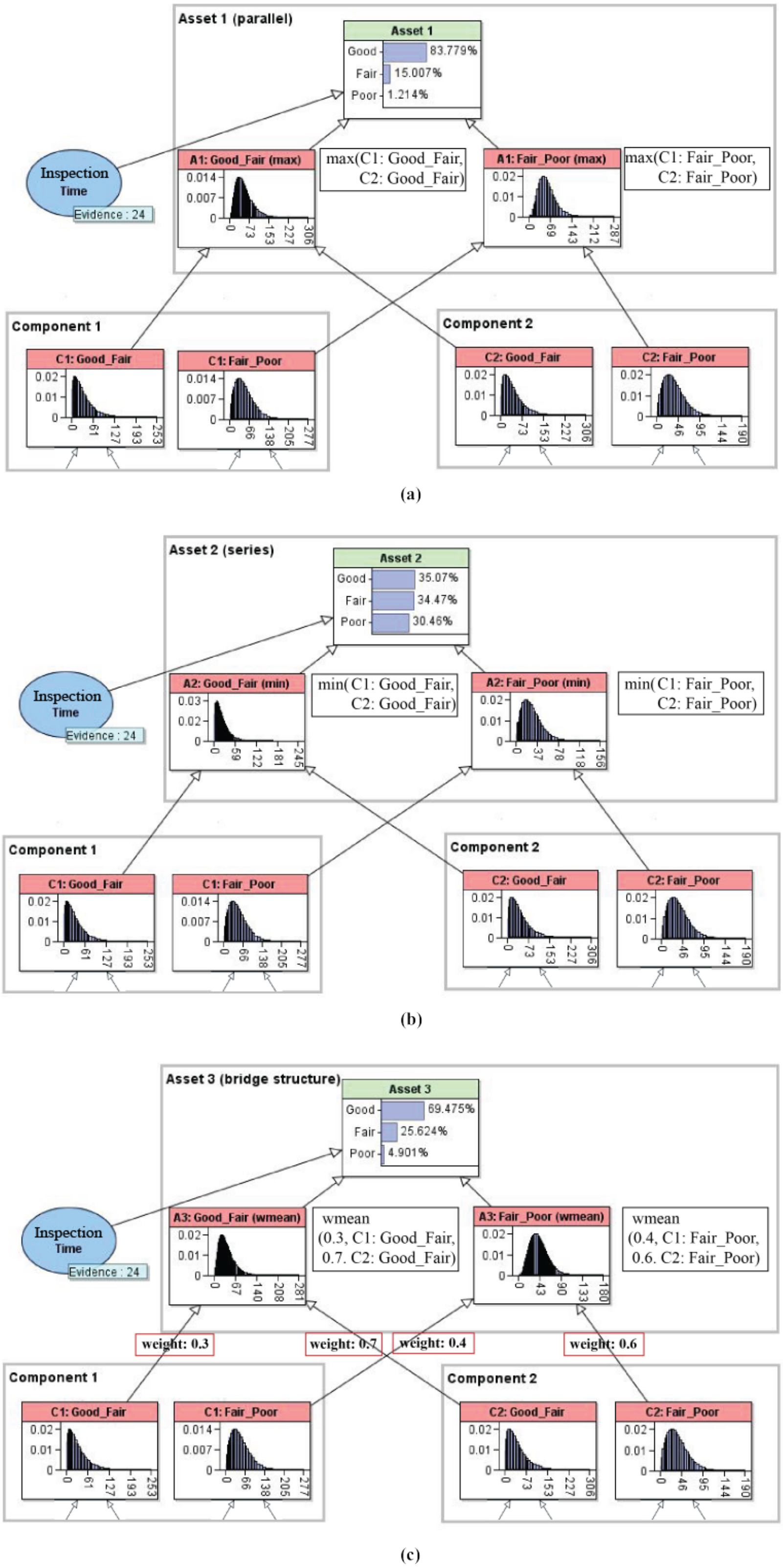


Figure 4.11 Multi-state asset system configuration: (a)asset assembled in parallel; (b)asset assembled in series; (c)asset assembled with bridge structure.

Consider Asset 1, 2 and 3 (A1, A2 and A3) all consisting of two components, Component 1 (C1) and Component 2 (C2), arranged in parallel, series and with a bridge structure respectively. Transition distributions of all components are learned from the deterioration learning models mentioned before. The system configuration models are depicted in Figure 4.11 for asset rated by multiple states (a three states example is showed: Good, Fair and Poor).

In Figure 4.11(a), the components are assembled in parallel. AND gates are used to describe the relationship between input and output event. AND gate is commonly used in the parallel system, representing the event fails if all the components in the system fail. We denote the continuous transition distribution of Component C_i as T_{C_i}, we have $T_{AND} = max\{T_{C_i}\}$. Each transition of A1 is constructed by an AND gate that connects the corresponding transition from C1 and C2. We also include a categorical variable developed from the last subsection that connects the transition distributions of A1 and the inspection time, representing the state of A1 given an inspection time.

Similarly, in Figure 4.11(b), the components are assembled in series and can be described by OR gates. OR gate is used to represent a situation where the event fails when at least one of the components fails. It can be denoted as $T_{OR} = min\{T_{C_i}\}$. Each transition of A2 is constructed by an OR gate that connects the corresponding transition from C1 and C2 and evaluated by a categorical variable.

Slightly differently, in Figure 4.11(c), the components are assembled with a bridge structure and evaluated by an aggregation function of its components with weights rather than using the fault tree gates. Each component in the bridge structure system is assumed to have a designated weight contributing towards its failure. And further, aggregated, an example of an aggregated weighted mean (wmean) is discussed here. We denote the weight of Component C_i as w_{C_i}, where $i = 1, \ldots, n$, we have the weighted mean transition distribution $\overline{T}$ of this system

$$\overline{T} = \frac{\sum_{i=1}^{n} w_{C_i} T_{C_i}}{\sum_{i=1}^{n} w_{C_i}} \tag{4.12}$$

Each transition of A3 is constructed by a weighted mean function that connects the corresponding transition from C1 and C2. In this example, experts believe that contributing to the transition between Good to Fair of A3 as a system, C1 has a weight of 0.3 and C2 has a weight of 0.7; for the transition between Fair to Poor, C1 has a weight of 0.4 and C2 has a weight of 0.6. These system transitions are further evaluated by a categorical variable to predict the asset condition.

Sometimes, decision makers are more interested in evaluating the state of the system by the state of its components, rather than the transition distributions. For example, in the US, one criterion in determining whether a bridge is structurally deficient is a condition rating of 4 or less for any of its structures, including deck, superstructure, substructure and culvert (if it exists). We can model them by considering each of these structures as a subsystem, and they are assembled in parallel.

In a simple case where assets are rated by binary states (for example, working or fail), we can model them as traditional fault tree with a Boolean node in the BN. The Boolean node is constructed with a comparative statement to express the logic gates and can be used to perform static fault tree analysis. For example, for a binary-state asset that is evaluated by the states of two components C1 and C2 in parallel, we can define: Asset $\sim$ $if(C1$ ==“Failed” && C2 ==“Failed”, “Failed”, “Working”). Here, C1 == “Failed” means C1 is in “Failed” state. This expression means if both events, event C1 is in “Failed” state, and event C2 is in “Failed” state, are true, then the result is in “Failed” state, otherwise in “Working” state. With BN tools like AgenaRisk [2], this expression for the Boolean operator can be automatically transformed into a corresponding CPT for inference in the BN. The joint probability distribution of the BN that has a collection of independent cause variables Component $\underline{C}$ consists of Component C_i, where $i = 1, 2, \ldots, n$ as the parents of the asset’s state Y:

$$p(\underline{C}, Y) = p(Y|\underline{C}) \prod_{i=1}^{n} p(C_i). \tag{4.13}$$

Therefore, together with the prior probability of the state of components (learned from the condition prediction discussed in the last subsection), after the inference of the BN, we can get the probability of a binary-state asset’s states assembled in parallel, in series or with a bridge structure. Thought we could extend the if-else statements from the Boolean nodes in binary-state systems to form a collection of logic expressions (nested if-else statements) to evaluate the state of a multi-state asset with components assembled in parallel or series, the process for defining these logic expressions becomes tedious with the increase in the state number.

Instead, for a multi-state asset that is evaluated by multi-state components, functions such as maximum, minimum or weighted mean can be used to describe the relationships between states of components and asset. To do that, we need to consider representing the states with numerical scales rather than merely using categorical states. Here, their states are modelled with a set of ordinal continuous intervals. The discrete variable (state of components and assets) is mapped onto a continuous scale that is bounded (from 0 to 1) and monotonically ordered. The reason for expressing them on an ordinal scale is that in most asset rating scales, the states are ordered. For example, in the NBI, rating 9 represents a better condition compares to rating 8. Given the number of asset states, we can discretise the numerical scale accordingly. Hence, the binary-state system becomes one of the special cases with an interval of 0.5. Figure 4.12 presents a three-state scale example. Therefore, the interval for each state is 0.333. Thus, **Good** state belongs to the interval of [1, 0.667) and **Fair** state belongs to the interval of [0.667, 0.333) and so forth.

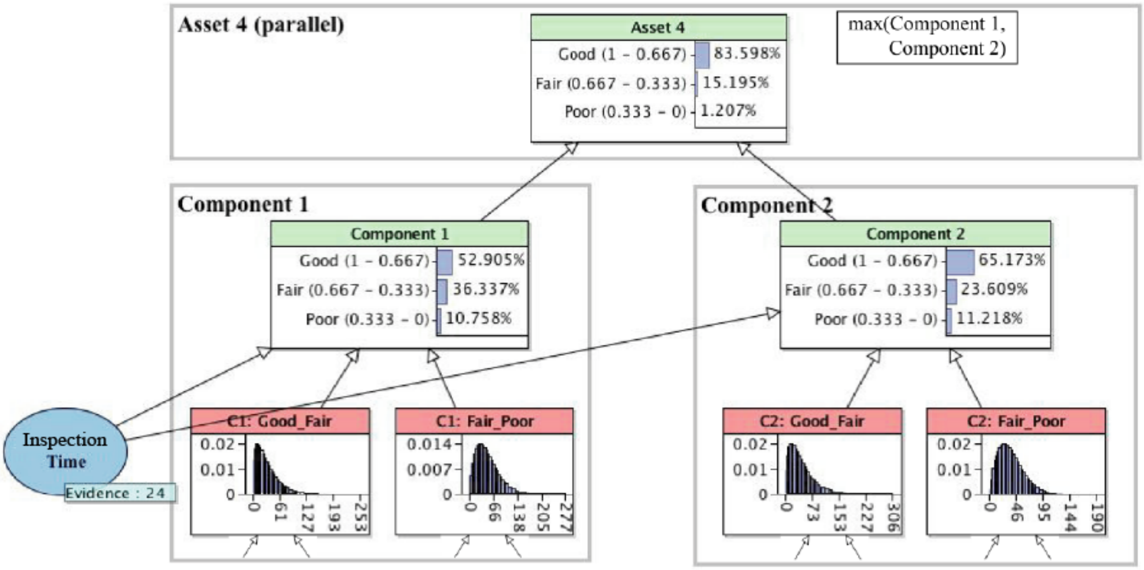


Figure 4.12 Multi-state asset system assembled by parallel components.

Table 4.2 CPT for an asset with two components that are assembled in parallel.

	Component 1	Poor			Fair			Good		
	Component 2	Poor	Fair	Good	Poor	Fair	Good	Poor	Fair	Good
	Poor (0 - 0.333)	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Asset 1	Fair (0.333 - 0.667)	0.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0
	Good (0.667 - 1)	0.0	0.0	1.0	0.0	0.0	1.0	1.0	1.0	1.0

Therefore, with these numerically scaled states, we can use maximum and minimum or weighted mean functions. Since the states, for example, Good to Poor, are mapped to ordered numerical bounds from high to low, and their states are mutually exclusive, the probability of asset under maximum, minimum or weighted mean function becomes the maximum, minimum or weighted mean state scale of their components and later re-marginalised. Figure 4.12 presents an example of two components assembled in parallel. The CPT generated from the maximum function for Asset 4 is given in Table 4.2. The probability of the asset follows the sequence of logic: if either component is in Good state, the asset is in Good state; or if either component is in Fair state, the asset is in Fair state; otherwise, the asset is in Poor state.

Additionally, for a bridge structure system configuration, the quantification of the contribution of each cause to the effect (credibility of state of the component to the state of an asset) often involves uncertainty. To consider the uncertainty, we can extend the state node that is discretized into several ordinal continuous intervals to a state node that is characterised by a TNormal distribution using these ordinal continuous intervals. This function is also implemented in the ranked nodes developed from Fenton et al. [46] and has been implemented in software like AgenaRisk [2] as discussed in Section 3.2.2. Therefore, we have:

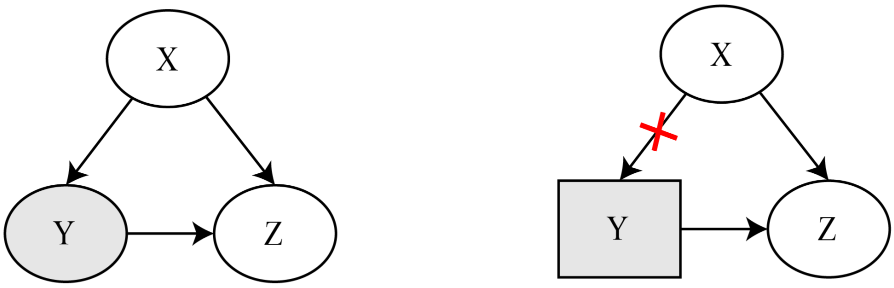
$$\text{Asset State} \sim \text{TNormal}(\mu, \sigma^2, 0, 1)$$

where μ is the weighted mean function of the state of components. σ^2 is the variance representing the certainty from experts about assigning the weights of components: ranging from 0 to 1, where 0 represents entirely certain about the weights, and 1 represents utterly uncertain about the weights. The lower bound of 0 and upper bound of 1 are applied to avoid probability from expanding into infinity ranges but instead regularise the resulting distribution within the finite range of 0 to 1. Hence, in order to include the uncertainty about assigning the weights, for bridge structure, instead of using a function like wmean (0.4, Component 1, 0.6, Component 2), we can use function TNormal (wmean (0.4, Component 1, 0.6, Component 2), 0.2, 0, 1). It not only includes the wmean function as the μ, but also consists of the uncertainty about the weights of components as σ^2, which is 0.2, a relatively high level of uncertainty in this example.

In summary, this subsection presents two types of modelling approaches to evaluate the state of an asset that is assembled by multiple components. Depending on the requirement of the applications, we can either estimate the state of an asset from its components' deterioration distributions, or its components' states. Especially for a bridge structure configuration, this subsection also shows how to extend our model to include expert knowledge. This knowledge reflects the expert's certainty level about assigning the weights of components, which represents the contribution of the state of each component to the state of the asset as a system. Though only three types of system configurations are developed: parallel, series and bridge structure, they can be combined or extended to model more complex configurations, for example, using the AND and OR gates together to model a k-out-of-n system or extend AND gate with a logic expression to form a Priority AND gate to model system redundancy as suggested in Marquez et al. [108].

4.4 Repair Decisions

When the state of an asset has been identified, either from on-site inspection or from prediction, we would like to decide what repair action to perform. This section develops two types of supports for making these repair decisions: one for suggesting maintenance actions from the historical frequency of repair action usage, which is built upon a type of BN called an observational model; one for evaluation of repair effectiveness, which is built upon a type of BN called an intervention model.



(a) Common Cause Confound Observational Model

(b) Common Cause Confound Interventional Model

Figure 4.13 Extensions of common cause observational and intervention models with confounders.

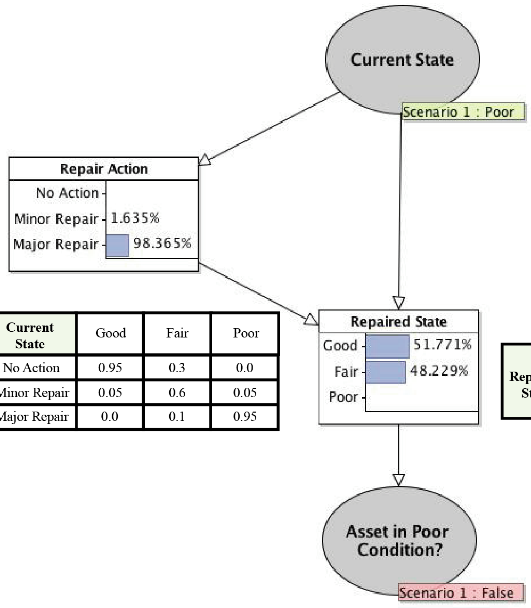
These models are developed using an extension from one of the three basic causal structures introduced in Section 3.2.3: a common cause with confounder [112, 155], it represents a situation that an event is causally affecting both the cause and the effect as shown in Figure 4.13. It can be used to model the effectiveness of repair action given its current condition. X represents the current state, Y represents the selection of repair action, and Z represents the repaired state. Each variable is modelled as a discrete labelled variable with multiple states or options. The joint probability distribution of this model can be represented as $P(X,Y,Z) = P(X)P(Y|X)P(Z|X,Y)$.

In the observation model (Figure 4.13 (a)), the passive observation of taking a positive repair action indicates a higher probability of the current state is in a worse state, and a higher probability of the repaired state is in a better state. We can use this model to backwardly reason the recommendation on repair action when given a safety or reliability criterion as an observation.

While in the intervention model (Figure 4.13 (b)), the decision on repair action is independent of the previous state. The decisions could base on other external reasons, for example, financial reason or opportunistic maintenance with other assets. Thus, the observation on repair action does not provide evidence of the previous state (hence, the link is removed) but only have an impact on the repaired state. So that, we can answer a question such as: 'Given the current state, if repair action X is taken, what is the probability of this asset in state Y?'. By comparing the effectiveness of different actions, we can prioritise the maintenance actions.

4.4.1 Observational Model: Historical Frequency of Repair Action

Given an observed condition of an asset, maintainers may want to know under the same scenario, what action would other maintainers take. We can build an observational model like Figure 4.14. In this model, an asset is rated by a 3-point scale: Good, Fair and Poor. Three repair actions are included for illustration purpose. Repair action is the child variable of the observed asset state since historically, the decision on repair action depends on the state of the asset. These two nodes together become the parent nodes of the repaired state. We also include a constraint node to evaluate if the repaired state is still in Poor condition.



Current State	
Good	0.333
Fair	0.333
Poor	0.333

Repair Action \ Current State	Good	Fair	Poor
No Action	0.95	0.3	0.0
Minor Repair	0.05	0.6	0.05
Major Repair	0.0	0.1	0.95

Repair Action	No Action			Minor Repair			Major Repair		
Current State	Good	Fair	Poor	Good	Fair	Poor	Good	Fair	Poor
Repaired State: Good	1.0	0.0	0.0	1.0	0.8	0.0	1.0	0.99	0.5
Repaired State: Fair	0.0	1.0	0.0	0.0	0.2	0.3	0.0	0.01	0.45
Repaired State: Poor	0.0	0.0	1.0	0.0	0.0	0.7	0.0	0.0	0.05

Asset in Poor Condition? \ Repaired State	Good	Fair	Poor
False	1.0	1.0	0.0
True	0.0	0.0	1.0

Figure 4.14 Observational model for repair action suggestion from historical frequency.

The CPT of the current state variable is uniformly distributed, but in most cases, this node is observed. In this example, it is observed with a Poor condition (100% probability in Poor condition). The CPT of repair action depends on the observed state. Its entries are adopted from the historical frequency of repair action usage or expert judgement. For example, when the asset is inspected with a Fair condition, in the repair history, 30% of maintainers took no action, 60% used minor repair, and 10% applied major repair. A similar procedure is applied to specify the CPT of the repaired state about maintenance effectiveness. This example assumes the available actions are imperfect maintenance, and different actions have different effects in restoring asset state depending on its current state. For example, in history, assets repaired by major repair actions, 99% of them restored asset in Fair condition to Good condition while 1% didn't have any effect; 50% of them successfully restored asset in Poor condition to Good condition, and 45% to Fair condition but 5% failed to improve its current state. The constraint node is an observable Boolean node with a logical expression if(Repaired State == "Poor", "True", "False"), and its equivalent CPT is shown in the figure. The False state is observed in this node to prevent the repaired state from being in Poor condition.

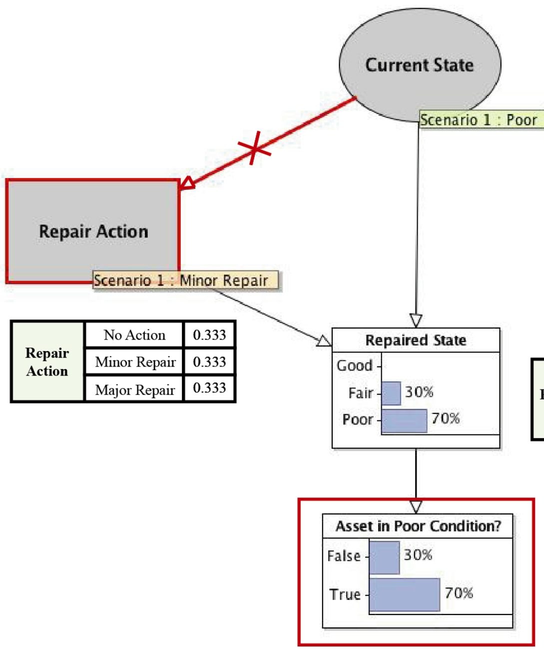
By observing the state of the asset and the constraint to prevent asset from being in an unreliable condition, with this observational model we can infer the repair action frequency that is taken in history or from expert experience. In Figure 4.14, for an asset that is currently in Poor condition, to avoid its repaired condition still in Poor condition, in history, 1.635% of maintainers took minor repair action and 98.365% repaired with major repair.

In this example, we modelled a simplified relationship between repair actions and asset state. The model can be extended to describe more complex maintenance assumptions, such as some repair actions are only applicable to restore asset in a specific range of state, or repair actions depend on the states of multiple components. We can also include more constraints, such as costs and expected service time, to provide a reference for making repair decisions under a more practical scenario. These complex decisions are later illustrated in Chapter 6 with real-world cases.

4.4.2 Intervention Model: Effectiveness of Maintenance

An intervention model can be used to answer questions such as what the condition distribution of the asset is if a specific repair action was taken. The distribution implies imperfect maintenance that not all actions can restore asset condition to a particular state, but a collection of states probabilistically.

Figure 4.15 presents an example of how to transform an observational model (Figure 4.14) into an intervention model for repair action evaluation. Followed the suggestions from Constantinou et al. [28], in the intervention model, the link from the cause should be removed. In the observational model, this link is used to explain the observation for repair intervention; while in the intervention model, we need to remove it to avoid inferring posterior probability for cause (current state node) when performing intervention (observed). Though technically this does not have an impact on the example in Figure 4.15 because both current state node and repair action node are observed. It is still necessary to remove this correlation because the current state could be predicted instead of being observed (see an example later given in maintenance planning), with the link the model will wrongly reason backwards and have an impact on the current state. Furthermore, the constraint node becomes unobservable; it acts as an indication of the intervention impact.



Current State		
	Good	0.333
	Fair	0.333
	Poor	0.333

Repair Action		
	No Action	0.333
	Minor Repair	0.333
	Major Repair	0.333

	Repair Action	No Action			Minor Repair			Major Repair		
	Current State	Good	Fair	Poor	Good	Fair	Poor	Good	Fair	Poor
Repaired State	Good	1.0	0.0	0.0	1.0	0.8	0.0	1.0	0.99	0.5
	Fair	0.0	1.0	0.0	0.0	0.2	0.3	0.0	0.01	0.45
	Poor	0.0	0.0	1.0	0.0	0.0	0.7	0.0	0.0	0.05

	Repaired State	Good	Fair	Poor
Asset in Poor Condition?	False	1.0	1.0	0.0
	True	0.0	0.0	1.0

Figure 4.15 Converting an observational model to an intervention model for repair effectiveness.

Most CPTs for the transformed intervention model remain the same as those in the observation model, except for the CPT of the repair action. The repair action node becomes independent from current state node and is observed. Hence, like the current state node, its CPT is uniformly distributed. Both the observational model and intervention model provide natural incorporation to express the repair history data through the CPTs, which allows us to reason the repair effectiveness from data. Additionally, thanks to the nature of the BN, even without these data, we can also formulate these CPTs based on expert judgement [28].

Figure 4.15 shows an example of estimating the impact of an intervention. For an asset that is in Poor condition, performing a minor repair only has a 30% probability restoring the asset to Fair condition yet 70% probability staying in Poor condition. Hence, in this case, we may suggest using major repair action instead.

This subsection presents a simplified intervention model to estimate maintenance effectiveness. Same as the observational model, it is also possible to extend this model to formulate more complex repair decisions, which are illustrated in Chapter 6 with real-world examples. Also, by considering the intervention model as a model object, we can form multiple objects to model multiple repair decision cycles. Together with the condition prediction model from Section 4.3, we can plan the maintenance over some time, and further, perform analysis such as life cycle analysis as discussed in the following section.

4.5 Maintenance Planning

Intervention model from the last section gives us a tool to evaluate the repair effectiveness but does not explicate decisions should be taken to optimise the use of resources automatically. An adequate approach is needed to formalise the process of making these decisions, especially with the increase in complexity in decisions making of many problems, such as a situation where future decisions depend on past decisions [45]. In this section, the repair decision process is modelled as an object (fragment) that contains the intervention model together with the condition prediction model. By modelling multiple objects that are arranged sequentially, it allows us to model decision processes over a finite time. A life-cycle cost analysis is used as an example to show its potential use when considering multiple sequentially organised decisions. We also point out the possibility to plan optimal maintenance decisions when working together with an optimisation technique.

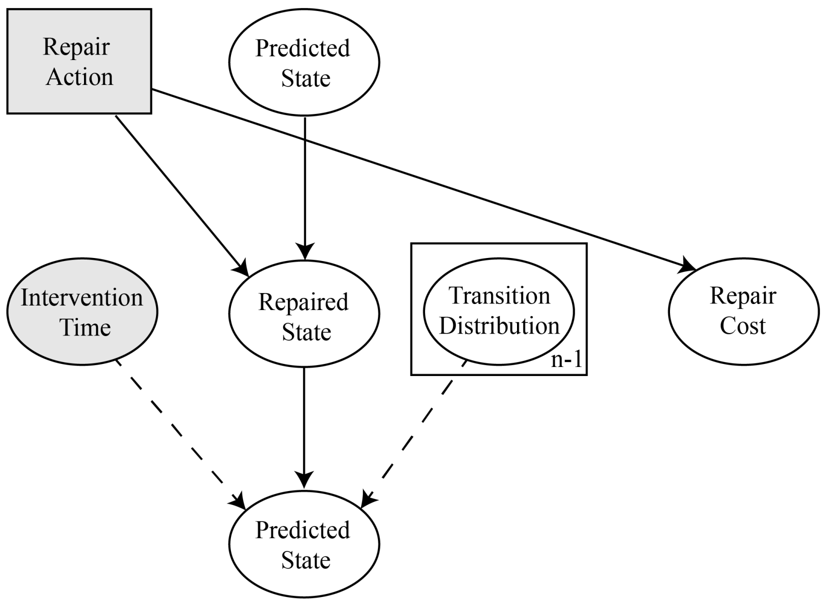


Figure 4.16 An object for intervention process modelling.

Figure 4.16 gives an example of combining the intervention model from Section 4.4.2 with the prediction model from Section 4.3 as an object. Note that instead of evaluating whether the repaired state is in Poor condition, in this example we measure the repair cost using an arithmetic function that depends on the selected repair action, which is later used to analyse the life-cycle cost. The repaired state becomes one of the parents of predicted state, which can be used to measure the further deterioration of the asset after maintenance. Different from the model in Figure 4.10, where an asset is assumed to be in a specific condition (hard evidence, for example, 100% in Good condition); the repair action cannot guarantee perfect maintenance. Hence, the repaired state is distributed probabilistically (soft evidence). The previous binary factorisation, therefore, becomes a special case when the repaired state is 100% in Good condition. In fact, this special case only applies to the situation for replacement in most cases. While for imperfect maintenance, to deal with various probabilities about various repaired states, we need to perform multiple levels of binary factorisations.

Assume asset can be rated among $\{n, \ldots, 1\}$ states, where state n is the perfect state, and state 1 is the worst state, we have $n-1$ transitions. By dividing the factorisation into $n-1$ blocks, where each block represents a factorisation process that start with one of the possible states. Given an intervention time, for state i, we perform a binary factorisation with the Markov property (for example, transition to state after $i-1$ only depends on the transition to state after i) that evaluate whether the asset will deteriorate to state $\{i, \ldots, 1\}$. We model this process by a collection of temporal Boolean nodes from 'transit to state after i?', to 'transit to state 1?'. In the block for state i, all temporal Boolean nodes are aggregated to form a temporal predicted state node representing the state distribution if the repaired state is in state i. The repaired state node contains a message about the probability distribution of each state. Together with the temporal predicted state nodes, they are aggregated to form an ultimate predicted state node representing the state distribution of this asset.

To define the CPT for the ultimate predicted state node, the predicted probability in each state needs to be determined first. For repaired state and ultimate predicted state, we have collections of state distributions represented as $P_{repair,i}$ and $P_{predict,j}$ respectively, where $\sum_{i=1}^{n} P_{repair,i} = 1$ and $\sum_{j=1}^{n} P_{predict,j} = 1$. For the temporal predicted state that starts with repaired state i, its probability of deteriorate to state j is represented as $p_{i,j}$, where $\sum_{j=1}^{i} p_{i,j} = 1$, and $i \geq j$. Therefore, the probability of the ultimate predicted state in state j is:

$$P_{predict,j} = \sum_{i=j}^{n} p_{i,j} P_{repair,i} \tag{4.14}$$

The prediction part from Figure 4.16 is unfolded in Figure 4.17 for an asset that is rated by four states: Good, Fair, Poor and Fair, hence, three transitions (the index $n-1=3$). For the repaired state that is in Good condition, followed the procedure in Section 4.3, its

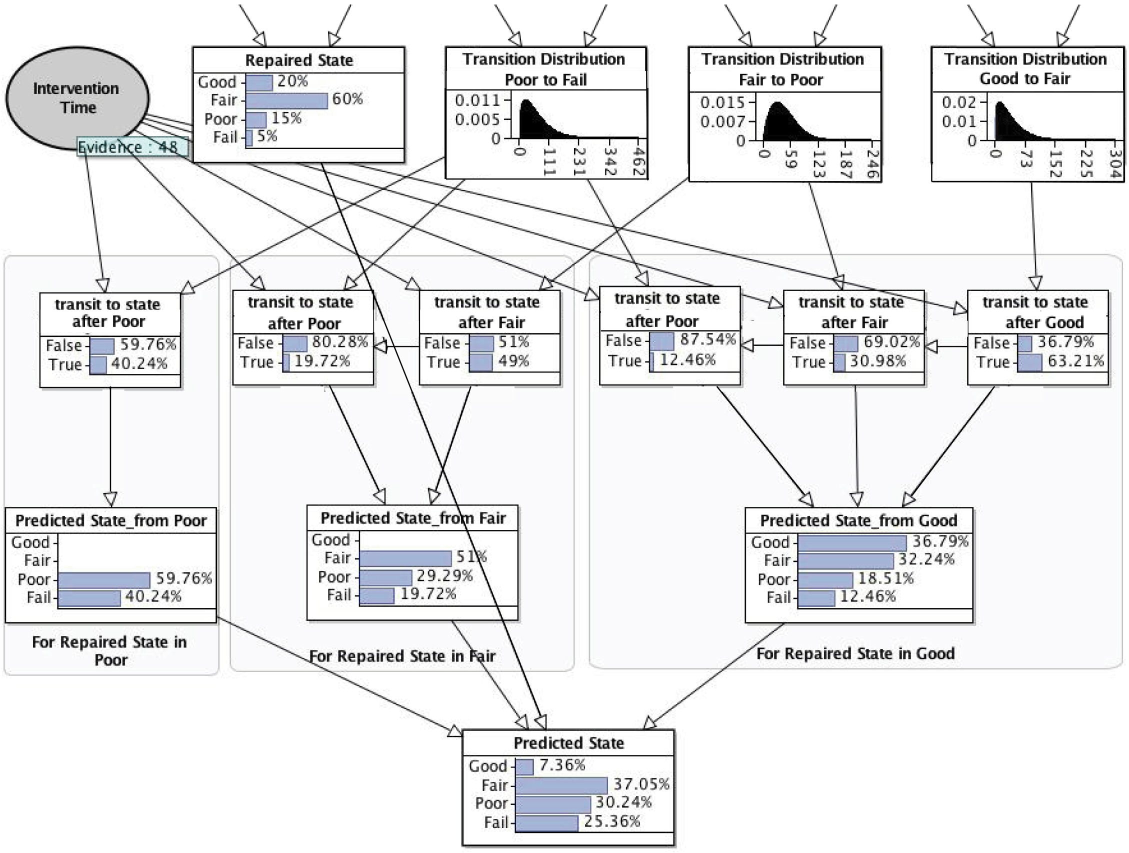


Figure 4.17 Multiple binary factorisations for the repaired state distributed probabilistically due to imperfect maintenance.

deterioration is binary factorised with three temporal nodes about its step-wise transitions given the observation of intervention time. These temporal nodes are gathered together to evaluate the further deterioration of asset under the circumstance that the repaired state is in Good condition. Similar structures are performed to evaluate the deterioration of the asset state in Fair state and Poor state respectively. A collection of temporal predicted states for asset starts with various repaired states is linked, together with the probability distribution of the repaired state to infer the ultimate predicted state. For example, to evaluate the probability of this asset in Poor Condition, Equation 4.14 is employed, where $j = 2$ and $n = 4$. We need to combine three situations, where each situation is multiplied by the probability of it being in that repaired state: in 48 months, the probability of repaired state in Good state deteriorates to Poor state, the probability of repaired state in Fair state deteriorates to Poor state, and the probability of repaired state in Poor state stays in Poor state. Hence, we have $18.51\% * 20\% + 29.29\% * 60\% + 59.76\% * 15\% = 30.24\%$.

For critical infrastructure, inspection and maintenance are often performed periodically. Therefore, to model a lifetime maintenance process, we can model multiple intervention processes that are either carried out with a fixed time interval of a dynamic time interval. Each intervention process is represented as an object that consists of the decision on repair action and its further deterioration till the next intervention. By extending the model from Figure 4.16, these multiple intervention processes are modelled as multiple BN objects that are arranged sequentially as shown in Figure 4.18. In each object, the posterior of predicted state becomes the prior of the predicted state of its next object (next intervention). The repair action includes no action, repair or replacement of the next object. It is determined by the state of the asset's further deterioration (for example, the most probable state or a state is above a certain percentage) since the last intervention. The repair cost is cumulatively calculated in the next intervention depending on its repair action. By performing multiple cycles of this model, we can estimate the life cycle cost of this asset that considers both repair decisions and further deterioration.

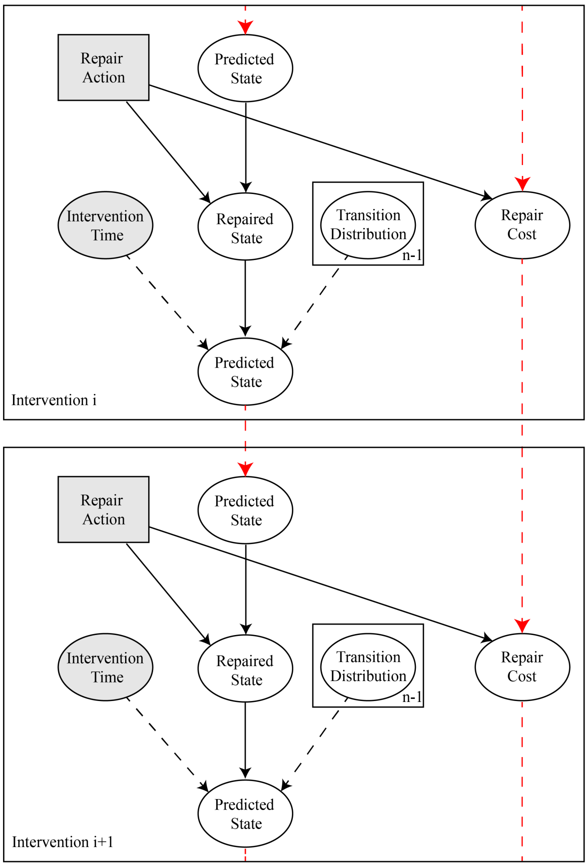


Figure 4.18 Modelling the process of multiple interventions as multiple BN objects that are sequentially organised.

This model can be used to estimate the life cycle of asset maintenance and identify potential risk and expenditure. To further utilise this model, we can incorporate a maintenance strategy to select optimal repair action and to decide intervention time interval. For example, by setting constraints to activate repair actions (e.g. some repair actions can be only used when the asset is in a particular state range), with objectives like minimum overall repair cost in the whole life cycle or ensuring asset state is above a certain level for safety reason, this forms a mathematical optimisation problem. It can be tackled by a range of techniques, such as heuristic algorithms, to decide which repair action to select at what time. Together with our model, we can perform analysis like optimal life cycle cost analysis to give more valuable suggestion when planning maintenance activities.

4.6 Summary

4.6.1 Summary of Models

In this chapter, we have developed several generic BNs with both continuous and discrete variables for asset modelling, including:

1. *A BN that can learn the rate of asset deterioration from both data and expert judgement.* In particular, we show how to model deterioration data with censorship, and how to elicit expert knowledge to assign priors for Weibull distribution's parameters (Objective I).

2. *Two hierarchical BNs for individualised asset deterioration prediction.* One can learn from similar assets when some assets have more deterioration data, but some have less. The other one can learn from different assets when most of the assets have little deterioration data, but at the same time, assets can be separated into different groups by their features. Features are chosen by their influence on the deterioration rates (Objective II).

3. *A BN that predicts an asset's condition at a given time using the learned transition distribution.* Initially for single condition state, this model is first extended to allow an asset to be rated with multiple conditions. A second extension of the model is to use the deterioration of an asset's components to assess its overall state. Different ways to model the contribution of the states of components to the overall state are explored (Objective III).

4. *Two repair decision BNs.* One can suggest repair action based on the historical frequency of actions being taken for assets in a similar given state, while the other predicts the effect of a selected repair action (Objective V).

5. *A model of the effect of a repair and the subsequent deterioration of an asset.* This model uses the earlier repair model as an object so that, with multiple sequential objects, we can model the life cycle of an asset through several cycles of repair and further deterioration (Objective V).

The models are generic in that each shows how to satisfy a particular modelling problem and decisions support need and can be combined and adapted to a specific maintenance situation. The models of many asset states use continuous variables to model the time of the transition from one state to the next, and this leads to high computational demand. To reduce this computational complexity, we adopt the binary factorisation technique developed from Neil et al. [123]. This technique first creates a Boolean variable for each state to represent the probability of the transition from this state at a given time, before combining these variables into the overall probability distribution over the possible condition states. This ensures that the models are feasible to compute.

4.6.2 Summary of the Use of Data and Knowledge

These BN models combine the available data with expert knowledge; we summarise the use of them below:

1. Available data:
   - **Censored data**: the exact time of an asset transitions from one state to another state is not always available. We introduce the modelling of censored data in Section 4.1.2 so that records from the periodic inspection can be used in place of exact transition times.
   - **Deterioration data from other groups**: some assets (e.g. new bridge types) may have only a little deterioration data. Therefore, we pool deterioration data with related asset types that are different but have related ageing processes as shown in Section 4.2.
2. Expert knowledge:
   - **Deterioration characteristics**: these models provide the framework to include knowledge from experts as the priors of a statistical distribution's parameters.

The assignment of the priors (hyper priors) is made possible by understanding the characteristics of each parameter. Triangular distributions are used in these models to estimate ranges for the Weibull distribution's shapes and scales as shown in Sections 4.1.3 and 4.2.

- **Deterioration similarity between groups**: knowledge that, for example, a concrete structure deteriorates more slowly than a timber structure is well-known by experts. An experienced engineer can estimate the degree to which two groups of assets will have similar deterioration. TNormal distributions are used in Section 4.2 to adjust the ranges of the parameters according to the group similarity.
- **Weighting of asset features:** experts may know that loading may have a higher impact on some assets' deterioration than coastal proximity. This type of knowledge can be used to distinguish individual assets by modelling the values of these features and aggregating them as the influence on the deterioration rate as shown in Section 4.2.2.
- **Contribution of components' states to bridge state**: Some components are more important in determining the condition of a bridge than others. This knowledge is expressed as a weight associated with each component, and they are further aggregated to represent the condition of a bridge as shown in Section 4.3.2.
- **Repair effectiveness**: Engineers may have experience about the effectiveness of different repair actions on restoring asset condition. This knowledge is expressed as a CPT about the probabilities of repaired condition conditional on the repair actions and current condition as shown in Section 4.4.

The use of appropriate deterioration model depends on the availability of data and knowledge. Also, how much data we need for each model varies between problems and the quality of data itself. From personal experience, to fit a Weibull distribution with complete data, we can approximate the true distribution with 50 observations using uniform priors or with around 20 observations using good priors. But in reality, we may only have censored data, as a result, the model often requires much larger data quantity for the distribution to converge. To choose a deterioration model, we can follow the sequence of: i) when there is a lot of deterioration data, we can use the model from Section 4.1.1 even with uniform priors, otherwise, ii) when we have some data and the engineers are confident in assigning priors for the parameters, we can use the same model with informative priors, otherwise, iii) when some assets have more data while their related assets have less, meanwhile the engineers can tell how similar the deterioration rates are between these asset groups, we can apply the model from Section 4.2.1 to use the learning from data-rich groups to infer the learning of data-poor groups, otherwise, iv) when the engineers cannot tell the similarity between groups, we can use model from Section 4.2.2 to learn the differences between groups using feature information.

The following chapters show how we can use and extend these generic models in the context of bridge maintenance. Chapter 5 validates the deterioration prediction models, at first using synthetic data and then in a real case study. Finally, it assesses the performance of the predictions in comparison to other available approaches. Chapter 6 shows how to use the decision models to make real-world decisions. All the later chapters build on the generic models introduced in this chapter; in particular, Chapter 7 presents a way to assemble them for different modelling requirements in a way that does not require detailed expertise in Bayesian modelling techniques.

Chapter 5

Deterioration Prediction Validation

Chapter 4 proposes a number of BNs, among them, the Bayesian statistical model can be used to learn the rate of deterioration from both data and expert knowledge (Section 4.1). This model can be extended to learn from other groups so that we can tackle a situation where some groups have little data for deterioration learning, but others have more (Section 4.2). We can apply this model to predict whether an asset will transit from one state to another state given a time (Section 4.3). For asset rated by multiple states, we can assemble multiple deterioration models in the form of a Markov model to predict multi-state deterioration (Section 4.3.1). This chapter aims to validate the use of these models (Objective IV). It first validates the hypothesises of these models using synthesis data in Section 5.1. National Bridge Inventory database is later used as the case study to show how they can be used for a real-world problem. We first introduce the background and challenge of this database in Section 5.2, then show how to build the prediction models using this database in Section 5.3. Section 5.4 validates the prediction performance of the models from various aspects. A summary is given in Section 5.5.

5.1 Validation Using Synthesis Data

Before building models for deterioration prediction on a real case study, we need to validate the hypothesis that we can approximate the deterioration distribution by learning their parameters from data and knowledge. A set of synthesis data that follows a Weibull distribution with a shape β equals to 2 and a scale η equals to 300 is first generated. These data are learned to fit a Weibull distribution using the Bayesian Parameter Estimation (BPE) method suggested in Section 4.1. We compare it with MLE, one of the most popular methods in distribution fitting (see Section 3.4). The BPE is set with a prior of a Triangular distribution

(1, 3, 20) for its shape β and a prior of a Triangular distribution (100, 350, 500) for its scale η.

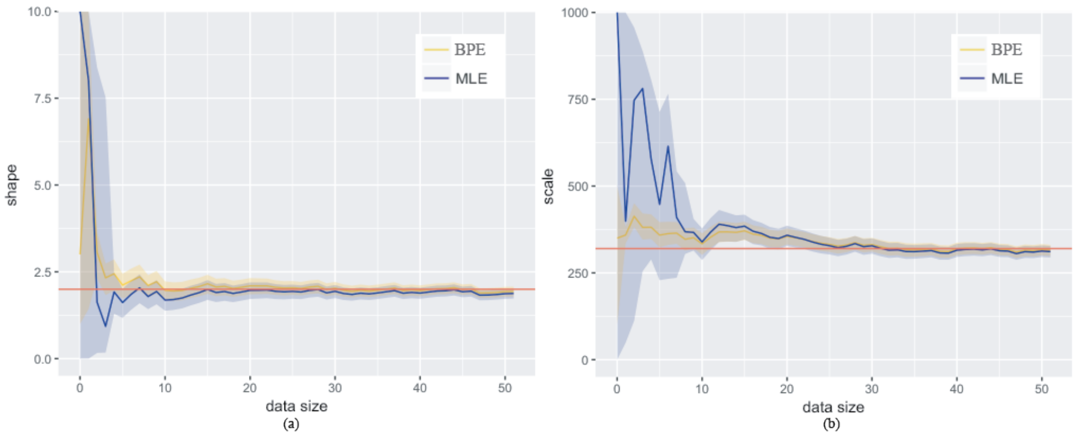


Figure 5.1 Parameter learning of a Weibull distribution with the increase in data amount: (a) shape β parameter; (b) scale η parameter.

With the increase in the number of data, the learned parameters are shown in Figure 5.1 - (a) for shape β parameter and (b) for scale η parameter. Among them, the red lines are the actual values, where the shape is 2, and the scale is 300. The yellow lines are the means of the learned parameter using BPE, and blue lines are the means of the learned parameter using MLE. The corresponding ribbons are their percentiles - from 5% lower percentiles to 95% upper percentiles.

We can see that with the increase in the amount of data, both parameters gradually approximate their actual values with narrower percentiles in both methods. When it reaches 50 data samples, the parameters are nearly identical with the actual values. We can also see this behaviour in Figure 5.2 (b), the posterior pdf from both methods presented a good fit with the actual distribution. It supports the hypothesis that with enough data, it is possible to approximate the distribution by learning parameters from the data.

Meanwhile, compared to parameters estimated by MLE, both parameters in Figure 5.1 learned using BPE show closer distances to the actual values and narrower percentiles when the amount of data is small. Figure 5.2 (a) presents an example of the posterior pdf learned with little data. Since MLE cannot be executed when there is too little data to form the estimation, for the fitting of MLE we started with five samples. With five data samples, MLE shows a different distribution compares to the actual distribution. We suspect this is because the shape parameter from MLE was wrongly estimated with a mean smaller than 1. While for BPE, due to the prior of shape was estimated with a lower limit of 1 by experts, it shows a satisfactory distribution even with zero data sample. With the increase in data amount,

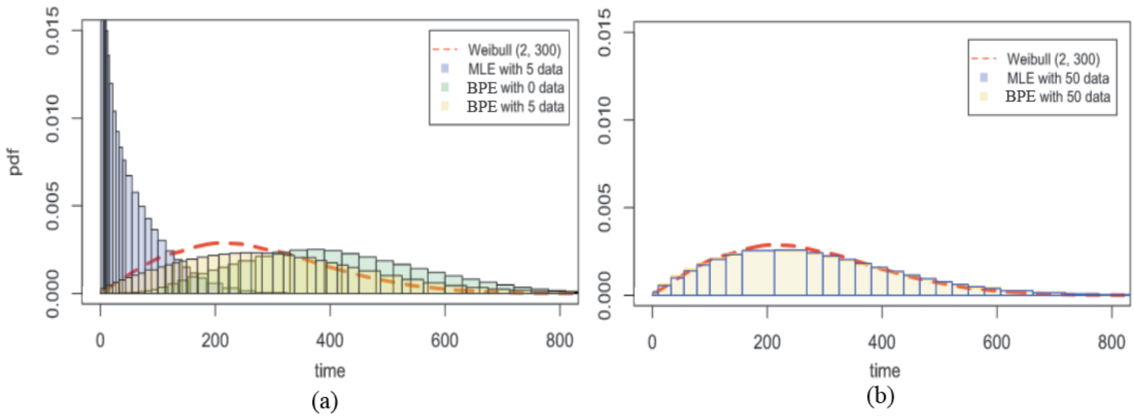


Figure 5.2 Fitted distributions: (a) with little data; (b) with 50 data.

for example, five data samples, the fitted distribution from BPE is improved with a closer distance to the actual distribution.

To quantify the distance between two distributions, Kullback–Leibler (K-L) divergence is also measured. Let i represents a time point, $p(x)$ and $p(y)$ represent the learned distribution and actual distribution respectively, we have:

$$D_{KL}[p(x) \parallel p(y)] = \sum_{i}^{N} p(x_i) \log \frac{p(x_i)}{p(y_i)} \tag{5.1}$$

The smaller the K-L divergence is, the closer the fitted distribution is to the real distribution. By randomly select samples from the learned distribution and compare with its corresponding samples from the real distribution, with a large enough sample amount, we can approximate the K-L divergence of two continuous distributions.

Comparing with the real (synthesis) Weibull distribution, distribution fitted with five data using MLE has a K-L divergence of over 8. It indicates the fitted distribution has a considerable distance with the real distribution. In contrast, fitted distributions using BPE present a satisfying performance. Even in the case where zero data is considered, by purely learning the parameters from prior knowledge, the learned distribution only has a 0.99 K-L divergence with the real distribution. It improves to 0.13 with five data input and 0.0075 with 50 data. Though MLE can also present a good fit when enough data is considered (with 50 data, its K-L divergence is 0.0101), this measurement agrees with the visual presentation in Figure 5.2 - when there is not enough data to fit a distribution, BPE can give a satisfying performance in the light of prior knowledge.

Nevertheless, in practice, it is unlikely to have this kind of ‘perfect data’ like the synthesis data. Apart from the uncertainty among data (censorship as discussed later), the data we have may not strictly follow a statistical distribution, but a combination of several distributions.

Figure 5.3 gives an example of the data we may have, which distributes like the distribution coloured in red. It is actually a mixture of data from distributions of Group 1, 2 and 3. Even in the case where we can approximate the distribution of the mixed group, its prediction (called aggregated prediction) from this learned distribution may not adequately represent the situation of an individual case.

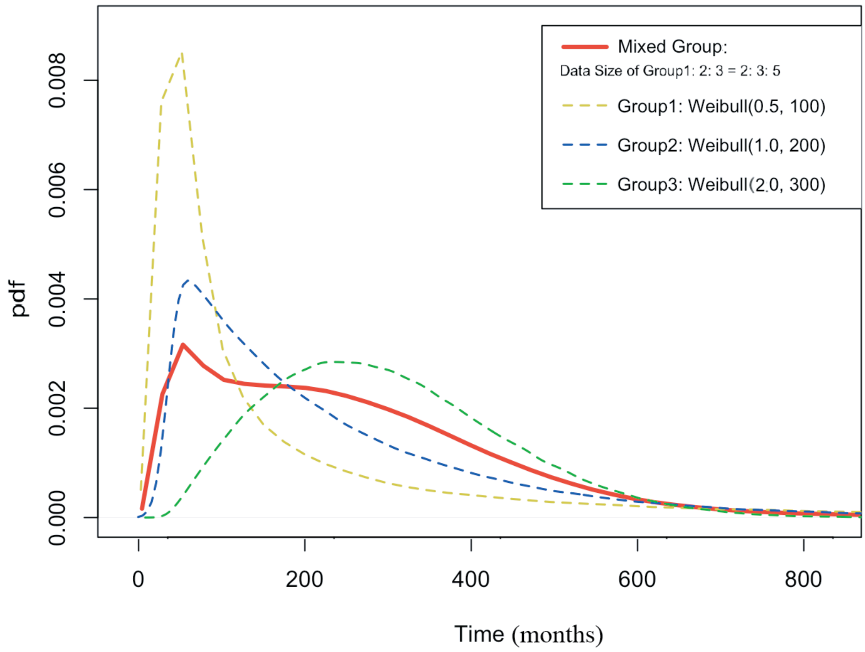


Figure 5.3 Three groups of distributions and a distribution mixed with these three groups.

For example, to predict a data point that belongs to Group 3 (coloured in green) at time 400 months, if we use the aggregated prediction from the mixed group distribution (coloured in red), we end up with a cumulative probability of around 0.92 which is quite different from its actual cumulative probability 0.83. This is resulting from deterioration in other groups generally have a longer deterioration time (longer tails in the distribution) compared to Group 3.

One of the solutions to resolve this problem is to individualise the data origins into separate groups. As discussed in Section 2.2.2, for infrastructure deterioration, by assuming their features (e.g. loading and environment) have an impact on their deterioration rate, it is possible to separate assets deteriorate similarly into groups. By doing so, we may provide a more accurate prediction using the individualised group distribution rather than the aggregated prediction. The purpose of the above discussion is to validate the hypotheses:

- We can approximate a distribution by learning their parameters from data.

- When data is not large enough, Bayesian parameter estimation gives a better distribution fitting with the help of prior knowledge about the parameters.
- Compare to an aggregated prediction, by separating assets into groups, it is possible to provide a better prediction by individualised their distributions into groups.

These hypotheses support the potential applicability of the methodologies. The following sections validate the use of the methodologies with a real case study from the National Bridge Inventory. Unfortunately, the validation metrics above cannot be applied to measure the deterioration prediction performance of real cases: unlike synthesis data where we know its actual distribution to compare with, we usually do not know the actual distribution of a real-world problem. Also, it gets more difficult since, in practice, deterioration data is most likely to be uncertain. For example, they could be censored as discussed in Section 2.2.1 and later showed in Section 5.2.1. Since in the NBI database, assets are rated by multiple states, in the following sections we consider the predictions as a multi-class classification problem: we validate the performance of different methods by predicting which state the asset will deteriorate to given time. We separate deterioration data into a training set and a testing set proportionally and apply two metrics (later discussed in Section 5.4.1) to measure the performance in condition prediction.

5.2 National Bridge Inventory

Mandated by the Federal Highway Administration to ensure the safety of public transportation, the NBI database archives unified information of bridges in the United States. It includes structure specifications, such as structure type; operational condition, such as Average Daily Traffic (ADT); and inspection data, such as structural condition.

The structural health of bridge is monitored through bridge inspection, which includes condition evaluation of deck, superstructure, substructure and culverts on an S9 to S0 scale, where S9 represents an excellent condition, and S0 represents a failed condition. In general, a condition rating of 4 or lower quantifies a bridge as structurally deficient. As a result, a bridge may require speed or load limit to ensure safety.

Inspection data on the structure condition provides us with a valuable source for the deterioration prediction model. However, the database contains only a small window of time frame. Only in recent decades, the NBI was enforced to unify bridge inspection data in accordance with the national bridge inspection standards. The publicly available data contains record ranging from year 1992 to 2017 by far. Since inspection is often conducted biennially (some higher risk bridges may require more frequent inspection), the average amount of inspection data generated from each bridge is very limited. Also, some records are missing due to bridge suspension or miscoding, and some records are repair data, which is not related to deterioration learning in our case. It leaves us a limited amount of data to use, where most of them are also noisy and imbalanced.

5.2.1 Deterioration Time Data

Since most bridge inspections are periodic, that is, conducted with a time interval rather than real-time inspection, the inspection time may not reveal the actual deterioration time of the structure. For example, a structure may be rated as State 8 (S8) in this inspection but rated as State 9 (S9) in the last inspection. We cannot directly infer the deterioration time from State 9 to 8 is the time gap between these two inspections because the deterioration may happen anytime between these two inspections.

To encode this uncertainty so that we can use inspection dates to infer deterioration time, we need to introduce censorship when representing the deterioration time [101]. Censored data is commonly used to represent incomplete observations. The lack of certainty of an event happened precisely at a time point can be measured by a value range using censorship. Except for cases like the accidental failure of a bridge, where we can infer complete failure data from noticeable signs, in the usual circumstances, we may have more confident in saying the structure deteriorated between two consecutive inspections (interval censored), after the most recent inspection (right censored), or before the first inspection (left censored), rather than a specific time point.

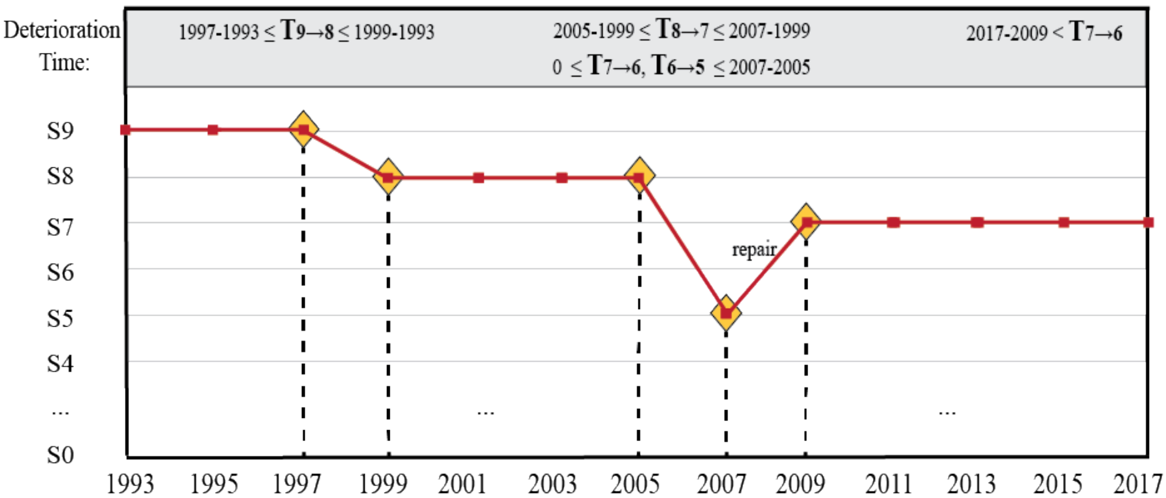


Figure 5.4 Censored data of deck condition from a bridge's inspection records.

Figure 5.4 presents an example of censored deterioration time data for a deck condition generated from the NBI database. In this example, represented by red squares, there are 13 inspection records, from year 1993 the deck was rated as S9 to year 2017 rated as S7. Five deterioration time are generated representing the time length of a structure stays in one state before it transits to another state. Four of them are interval censored, and one is right censored.

In 1997, the inspection showed the deck was in S9 while in 1999 it became S8. We are unsure about the exact time point of when this deterioration happened, but sure about this deterioration occurred between these two inspections. Therefore, inferred from two consecutive biennially inspections (from year 1993 to 1999), we can represent the length of this deck staying at S9 before it transits to S8 $T_{9\to8}$ is greater than four years but less than six years. In year 2007, the inspection result shows that the deck suddenly transit to S5. Except for accident like a train-strike caused sudden failure, we assume natural deterioration is decremental by one state at a time. Therefore, we can infer $T_{8\to7}$ is greater than six years but less than eight years, while $T_{7\to6}$ and $T_{6\to5}$ happen between year 2005 and 2007, hence, less than two years. After repair, the most recent inspection shows that this deck is still in S7, we can also infer the deterioration time for $T_{7\to6}$ is greater than eight years (right censored).

This example also reveals another concern about the amount of usable data we can infer from each asset. In this example, 13 inspection records only generate 5 transition time data. This situation gets worse since most bridges do not have a complete inspection history, and most of them have a long length of occupation time that do not generate any deterioration data. In the next subsection, we discuss another concern about the data inferred from the inspection records: the amount of data is imbalanced.

5.2.2 Imbalance Inspection Records

As shown in Figure 5.3, deterioration behaviours of assets may be different when they belong to different groups. We can use asset features to decide which group the asset belongs to, here we use feature bridge structure type as an example. The primary structure type of each bridge in the NBI database is determined by the structure material and the predominant structural design. A normalised heat map of the mapping between structure material and predominant structural in the NBI database is displayed in Figure 5.5, where the darker colour represents a higher frequency of bridge with these feature values.

The most common combinations are concrete and steel-based structures across different structure designs. 21.45% of them is made of steel and designed as stringer or multi-beam or girder. However, there is only 2.63% when it is designed as a stringer or multi-beam or girder but made of concrete and 0.03% when it is made of steel but designed as a slab.

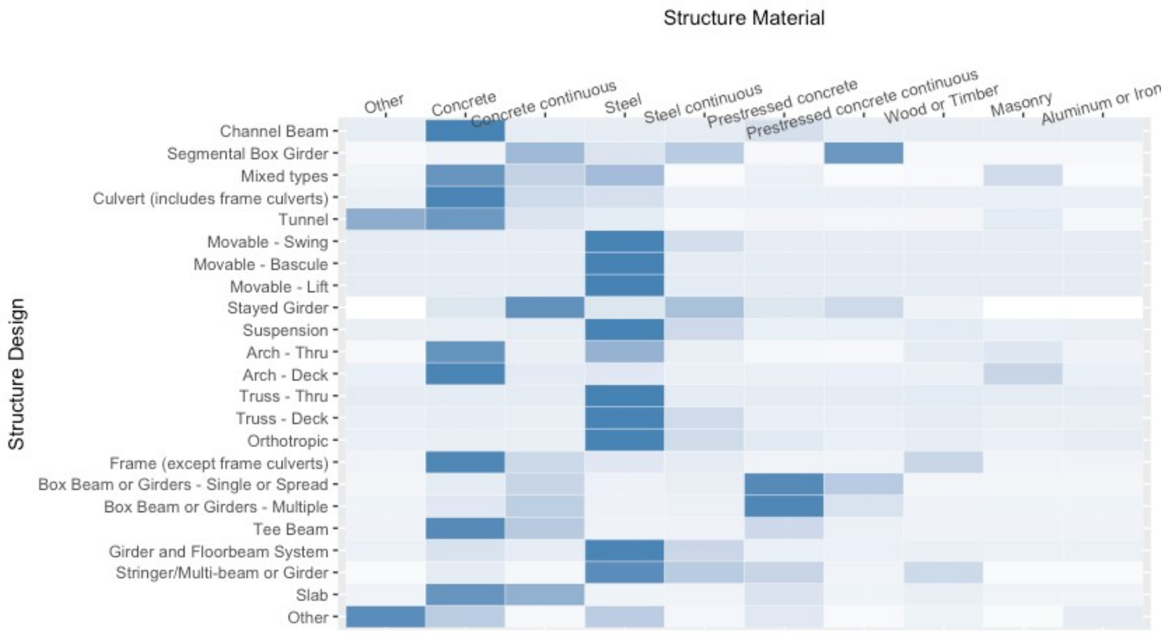


Figure 5.5 Heat map of NBI bridge structure type.

This imbalance of bridge type distribution rises a challenge in predicting the deterioration of those less common structure types: we may not have sufficient data to learn about their deterioration behaviours. While simply pooling all structure types together as one group may lead to inaccurate prediction, especially when structure type has a high influence on deteriorate rate. For example, typically, a steel structure deteriorates significantly faster than a masonry structure.

Another imbalance of the database is the condition distributions. Here defines S9, S8 and S7 as Good condition, S6 and S5 as Fair condition and the rest as Structurally Deficient (SD). Most structures in the NBI are in condition rating over 5 and few of them in condition rating 4 or lower. One of the reasons for this imbalance is that maintenance works were performed: for structures flagged as SD, as instructed by maintenance guideline, immediate rehabilitation work is often performed to prevent further deterioration. Hence, very few structures have records with poor condition. The age of bridges and length of record history are another two main reasons that may skew the distribution. For critical infrastructure like bridges, they usually have a long life span in their deterioration. While most of the bridges in the NBI database are relatively young - seldom of them may reach SD by far. Additionally, the NBI database was only enforced since 1992 with a relatively small time window to represent the statistics fully.

Figure 5.6 shows an example from Federal Highway Administration [44], the NBI bridge condition distribution grouped by ages in year 2017. We can see most of the bridges are

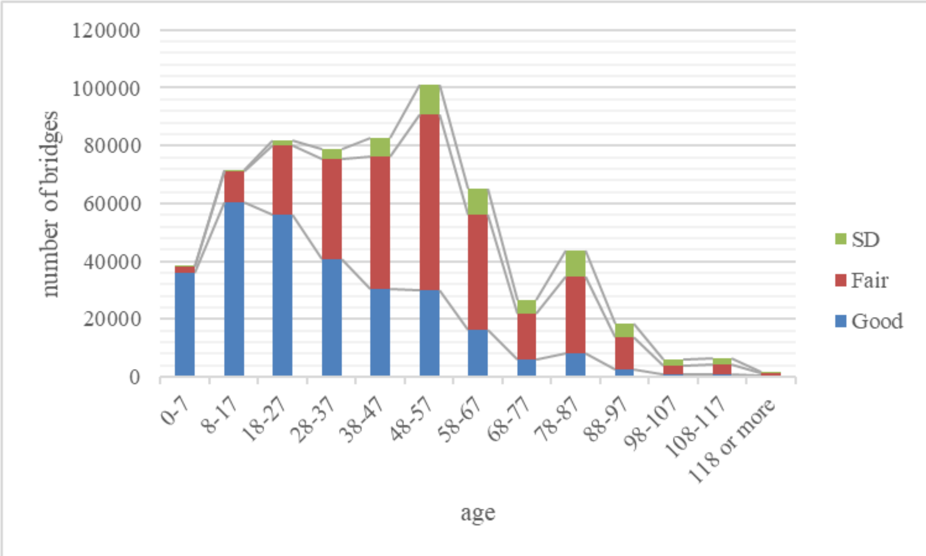


Figure 5.6 NBI bridge condition distribution by age in year 2017.

in Good or Fair condition. For older bridges, for example, aged between 78 to 87, around 20% of them are in SD, while for young bridges, for example, aged between 8 to 17 years old, only 0.68% of them are in SD. However, even so, this does not guarantee we can infer deterioration data from them since most of them may be repaired right after the inspection. Learning deterioration of these states becomes a challenge.

5.3 Building Deterioration Prediction Model

The NBI database encodes over a hundred features for each bridge. Taking too many features into consideration is inefficient and sometimes may result in overfitting [84]. Therefore, before building a deterioration prediction model, this section first reduces the dimensionality of the dataset to a small number of features that is representative enough in deciding the deterioration rate. The selected features are used to classify bridges into groups. This process is illustrated with the training data about deck structure in Wyoming in the NBI database. For each transition, a deterioration model that can learn parameters from other groups is built. They are further assembled to predict deck structure condition that is rated by multiple states.

5.3.1 Dimension Reduction in NBI Dataset

Apart from the features recorded in the NBI database, feature age, an indirect feature is also generated in our study due to its popularity in predicting deterioration [122, 23]. For a bridge that was reconstructed, age is inferred by the reconstruction year to the inspection year, while the rest of the bridges is inferred by the year of built to the inspection year.

Firstly, an exploratory data analysis is performed to reduce the quantity of the features. For example, features with no association to deterioration time, such as structure number; and features where over 95% of their population are missing, such as critical feature inspection date, are removed.

Secondly, a preliminary correlation matrix, of which Pearson correlations are used for continuous variables, and polychoric correlations are used for categorical variables, is developed to measure the statistical relationships between pairwise variables. This results in some strongly correlated variables, for example, feature maintenance responsibility and feature owner of the structure has a 0.94 correlation. One of them is considered redundant information. The latter feature is removed from the feature candidate due to feature maintenance responsibility has a higher correlation with the deterioration time. This process results in a candidate pool of over 40 features.

Lastly, to further reduce the dimensionality of the dataset, feature selection is performed using an R package called Boruta [88], which is built on a modified random forest (see Section 2.2.2) to evaluate the feature importance. First, Boruta adds and shuffles the value of duplicated features from the original features to remove their correlation with the response variable deterioration time. We called these new features as shadow attributes, and they are combined with the original feature space. A random forest classifier is then performed, and MDA measurement is used in our case. The bias caused by MDA metric is avoided thanks to the pre-processing procedure of removing highly correlated variables mentioned earlier. Additional Z scores among those shadow attributes are computed by dividing the mean of accuracy loss by its standard deviation in order to take accuracy loss fluctuation into account. Features with significantly high Z scores are tagged as important and vice versa. This process is repeated until all attributes are tagged. Details of this algorithm can be found in Kursa and Rudnicki [88].

Figure 5.7 shows a boxplot example using the NBI database of the deck structure in Wyoming. Rankings of features are separated by the Z score, which is coloured by blue box indicated as shadow attribute. Green boxes represent accepted features that have higher importance values, that is, are more predictive in evaluating deterioration time. Yellow boxes are tentative features that are medium significant, which are taken into a backup plan when needed. Red boxes are rejected or insignificant features, they are removed from the feature candidate pool. The labels on the left side are the features from the NBI database, and the numbers associated are their item ID. Detailed introduction of each feature can be found

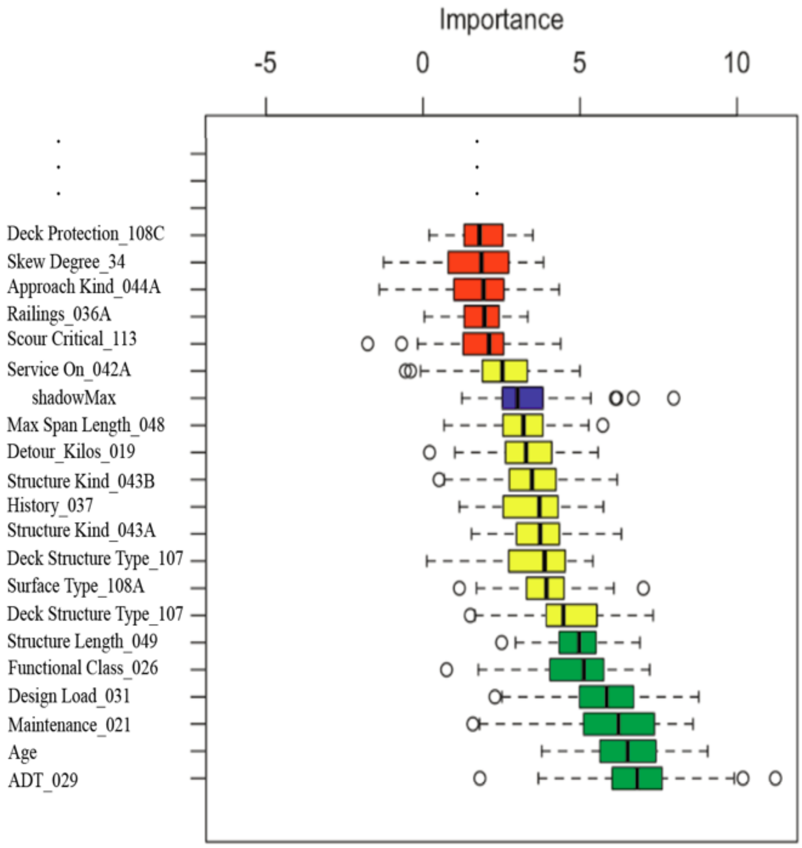


Figure 5.7 Feature selection for deck structure in Wyoming from the NBI database (lower the more important).

in the NBI manual [182] using the item IDs. A brief introduction of each accepted feature follows the sequence of its ranking is:

- **ADT**: average daily traffic, records the most recent average daily vehicles traffic volume on the structure.
- **Age**: an indirect feature generated from Item 27 built year, Item 106 reconstruction year and Item 90 inspection date, represents the current age of the structure.
- **Maintenance**: the maintenance agency that is responsible for the structure. Notes that different agencies may have different inspection training and regulations.
- **Design load**: the designed live load of the structure.
- **Functional class**: the functional classification of structure, for example, whether it is designated in a rural area or urban.
- **Structure length**: the length of the roadway supported by the structure.

5.3.2 Assigning Feature Levels

After the important features are identified, the possible values of each feature are categorised into several levels representing their indication of deterioration time. Three levels, from low, medium to high are used here.

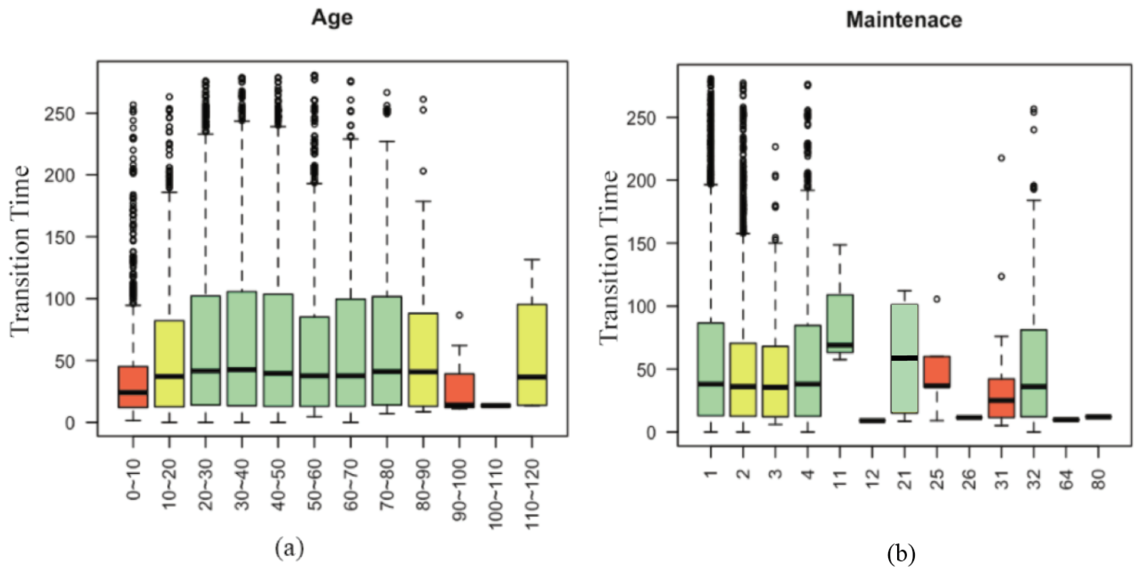


Figure 5.8 Assigning feature levels: (a)feature age; (b)feature maintenance.

Figure 5.8 presents two examples to assign feature levels. In the figure, the red bar represents level low, that structure has a higher chance to deteriorate faster; the yellow bar represents level medium; the green bar represents level high. Since feature age is a continuous variable, it is first discretised into several bins with a 10-year interval. By plotting against the deterioration time within the training dataset, we can see different age categories have different deterioration time distributions. For example, for age between 0 to 10, though several outliers having long deterioration time, most of them deteriorate within 50 months. Therefore, it is rated as low. Feature maintenance is a categorical variable. We can see that for example, a structure that is maintained by agency 1 - state highway agency, normally has a longer deterioration time, it is rated as high.

5.3.3 Learning Transition Distribution Between Groups

After the level of each feature is assigned, bridges can be separated into groups by their feature levels. For each transition, we applied the hierarchical BN developed in Section 4.2 to learn the transition distributions between different groups as shown in Figure 5.9. Notes that for demonstration simplicity, Figure 5.9 only shows the top two important features, ADT and Age. Later in Section 5.4.2 we will discuss how many features we should consider.

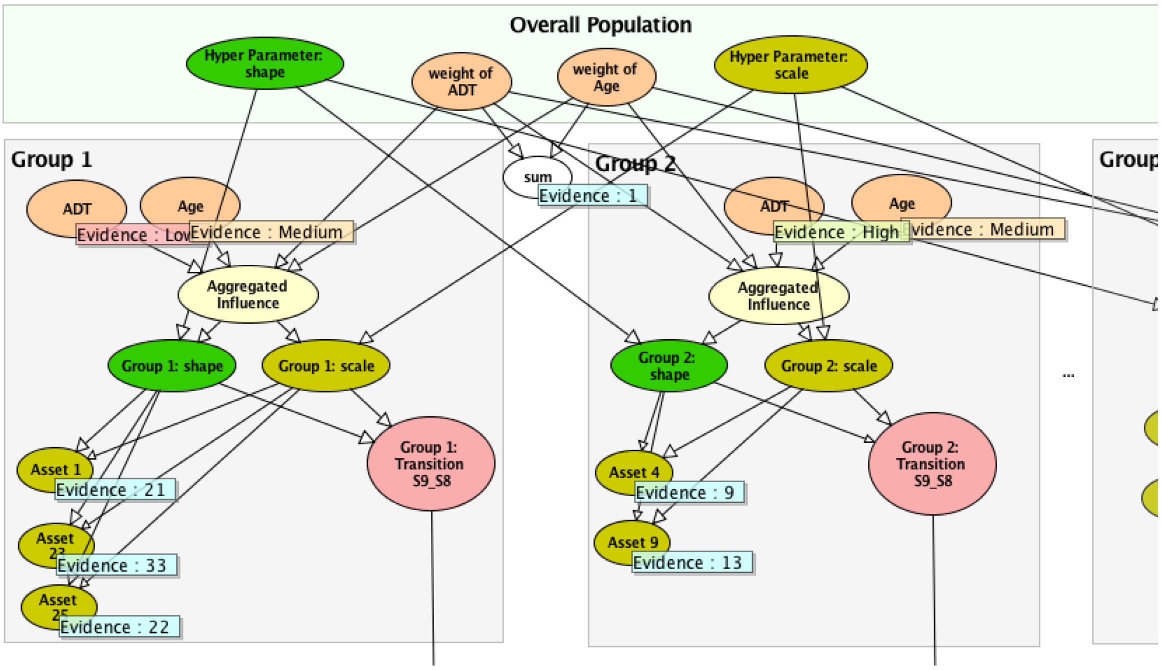


Figure 5.9 Hierarchical BN to learn between groups.

In this model, instead of eliciting each group's priors $\varphi(\beta)$ and $\varphi(\eta)$ in Equation 4.1 from experts, the priors of the local parameters are learned from hyperparameters together with an indicator called Aggregated Influence resulting from each group's features. The Aggregated Influence is essentially a linear indicator that aggregates all the relevant features together. It encodes the feature influence into low, medium and high three levels using a weighted mean of the feature levels with a variance representing the certainty of the aggregation. Instead of assigning the weights and the joint influence of each feature purely from experts, which would require sophisticated experience from experts, they are learned as hyperparameters as well. The weight of each feature is modelled by a uniform distribution from 0 to 1, where all the weights are summed to 1 with a constraint node. Each parameter is partitioned by the level of aggregated influence and its corresponding hyperparameter, where for example, a low-level influence indicates a faster deterioration. The instruction about how to elicit priors for a Weibull distribution introduced in Section 4.1.3 is adopted. We elicit the priors for the hyperparameters and their differences on the deterioration rates between different levels of influences and corresponding variances.

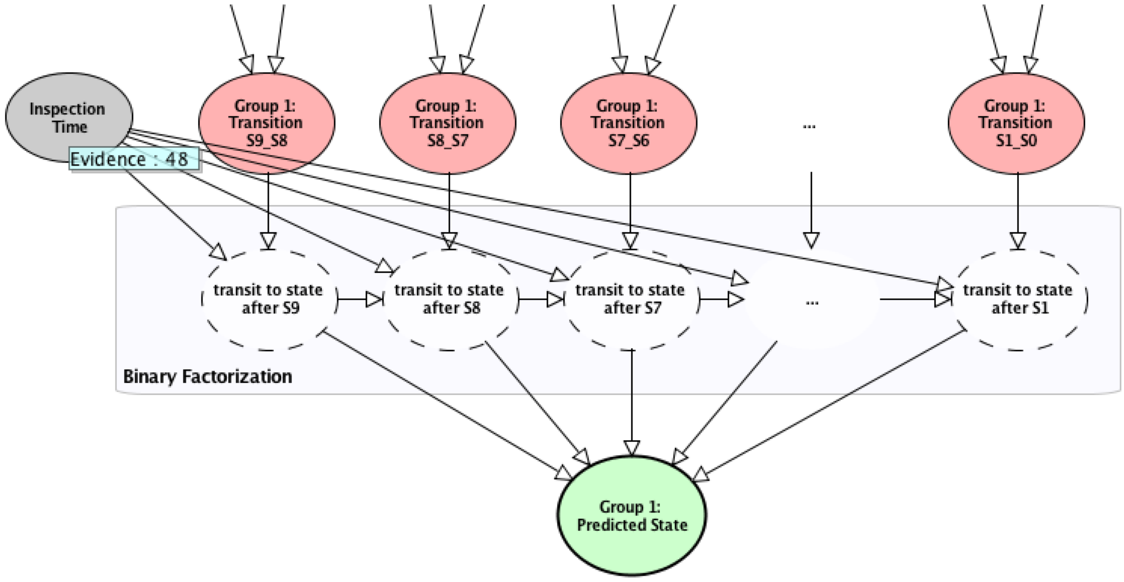


Figure 5.10 Condition prediction of assets in Group 1.

After all the deterioration models for all transitions are built, we assemble them using the model proposed in Section 4.3.1. Figure 5.10 shows an example of a deck from Group 1 that was previously inspected with a condition of S9. The model is used to predict its condition after 48 months since the last inspection. Several intermediate nodes are created to binary factorised them in order to reduce the inference complexity as described in Section 4.3.1. For each deck, we query the related transition models based on the group it belongs to. We enter its observation of the inspection time to produce individual condition prediction. The next section discusses the prediction performance in various aspects.

5.4 Predicting Deterioration and Its Validation

This section introduces how to measure the performance on the prediction, and investigates the performance of our approach from different perspectives.

5.4.1 Measurement Metrics

Accuracy rate is one of the most common measurement metrics for prediction performance, and it evaluates the fraction of the classification. For example, given a structure in year 2010 is in S7, we want to predict its condition in year 2017: if we predict it will deteriorate to S6 and it matches the actual inspection result in year 2017, we have an accuracy rate of 100%. By repeating this process for all the test datasets, we can have an average accuracy rate of our approach.

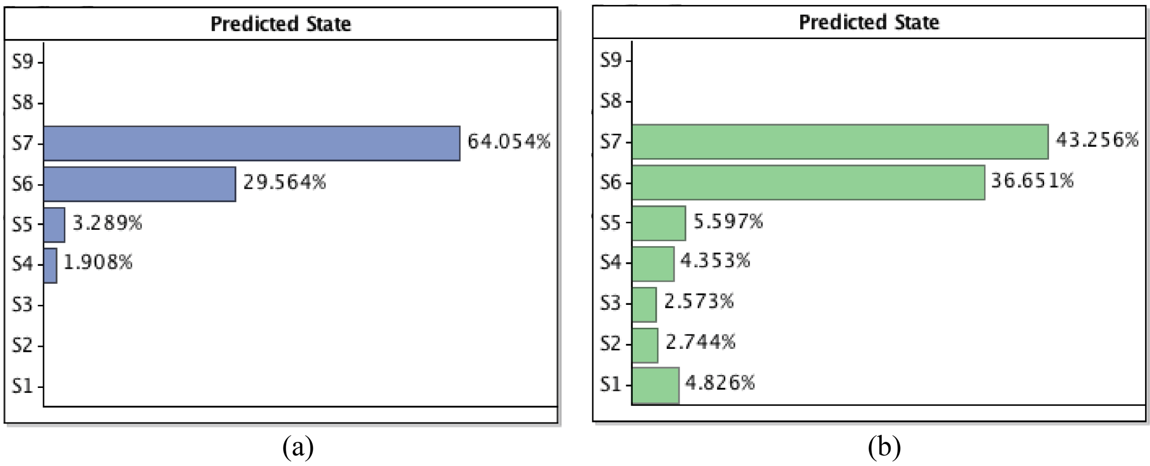


Figure 5.11 Two predicted condition distributions.

However, the inspection result itself is often uncertain. In most cases, inspection is carried out visually, though inspectors are trained with standard inspection manuals, the inspection result is still covered with a range of noise. The noise comes from, such as the inspection tools they used, environmental factors, or human subjectivity. An experiment was carried out by Phares et al. [142], which shows that for the same structure, 68% of the ratings given by different inspectors fell into a one-point interval range differences, and 95% fell into a two-point interval range differences. Therefore, in practice, when predicting the deterioration of an asset, other than a single estimation of the condition, we can also predict the condition of an asset problematically with multiple states.

Given a query, the developed BN predicts and outputs a probabilistic distribution over a discrete range of possible outcomes. Figure 5.11 presents two predictions for a deck's condition distribution after 45 months given its initial state is at S7. Though after 45 months, the actual inspection result of this structure is in S6, that is, both predictions are wrong if we only consider the accuracy rate by selecting the most likely state. However, compare to result in Figure 5.11(a), (b) gives a better prediction with a closer distance to S6. Ranked Probability Score (RPS) is chosen to measure this distance as it can measure the performance of multi-state probabilistic prediction with orders.

RPS is an extension of the Brier score for multi-class classification that also takes the distance between different possible outcomes into account. Assume there are K categorical events, the cumulative observation X_m and prediction Y_m can be defined by a vector of the observation's probability components and a vector of the prediction's probability components respectively:

$$X_m = \sum_{k=1}^{m} x_k,\ Y_m = \sum_{k=1}^{m} y_k\ , m = 1,\ \ldots,\ K \tag{5.2}$$

the RPS is the sum of the squared difference between the components of these two vectors

$$RPS = \frac{1}{J} \sum_{k=1}^{J} (X_m - Y_m)^2, J = K - 1 \tag{5.3}$$

A perfect prediction, for example, event k, would assign all the probability to x_k ($x_k = 1$), so the difference would be 0 with an RPS = 0, while the worst score is $K-1$ due to the accumulation [183]. That is, the smaller the RPS we have, the better the prediction we made. In the examples in Figure 5.11, (a) has an RPS of 0.052 while (b) has a score of 0.033. Hence, (b) is a better prediction. This calculation yields the RPS for a single event. To evaluate the performance of a prediction for a collection of events n, the average RPS can be defined as

$$R\bar{P}S = \frac{1}{n} \sum_{i=1}^{n} RPS_i \tag{5.4}$$

5.4.2 Selection of Feature Number

If all relevant features are used to distinguish the asset into different groups, then there is likely to be insufficient historical data to fit every transition distribution, even with 'data rich' group. Therefore, groups need to be defined by the features that are most important to the deterioration time but within a limited amount. After the feature selection in Section 5.3.1, six accepted features are considered as important. However, each feature is quantified into three levels depending on the feature values. Considering six features would result in $3^6 = 243$ groups, which is not only expensive for prior elicitation, but also expensive to perform inference on it. In this subsection, we investigate how many features we should consider in our models.

Deck condition started with State 7 (S7) between year 1992 and 2010 in Wyoming is studied in this subsection as it has the largest amount of data. We use them to learn the deterioration distribution for Transition 7 (T7, whether a deck in S7 will deteriorate into another state). To mitigate the influence of prior knowledge on the prediction performance, all the priors in this section are provided uninformatively with a uniform distribution from 0 to 20 in the shape parameter and a uniform distribution from 0 to 500 in the scale parameter. To investigate the performance of our model with the increase in feature number (from zero feature to six features), a set of experiments considers prediction of whether a deck in S7 will deteriorate into another state in the next four years (from 2011 to 2014) is performed.

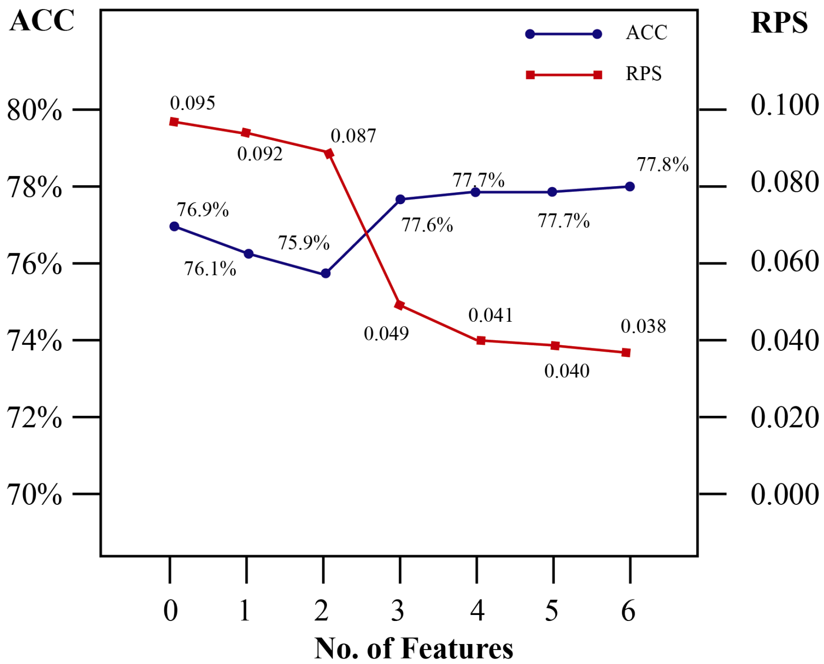


Figure 5.12 Accuracy rate (higher better) and RPS (lower better) performance with the increase in feature amount.

Figure 5.12 presents the result of the experiments. The blue line measures the average accuracy rate by the most likely state (the higher, the better) and the red line measures the average RPS (the lower, the better). Apart from the slight decrease in the accuracy rate when only one or two features are considered, the accuracy rate increases with the increase in the number of features. While the RPS performance always gets better with the increase in the number of features, that is, the prediction gets closer to the actual inspection.

After a drastic increase in the performance when three features are considered, both measurement metrics exhibit steady performance afterwards. Reason for this steady performance could be most of the assets may possess the same set of feature combinations. For example, with six features, there are 243 possible groups with different feature combinations, only 92 of them exists in the database. Among these 92 groups, 80% of them have less than two candidates. Therefore, it becomes unnecessary to consider too many features to form groups. Since considering three features already gives us a reasonably good performance, to avoid the increase in modelling and inference complexity, it is applied in the following experiments.

5.4.3 Multi-State Prediction Performance

This subsection investigates the performance of our approach compared to others in predicting asset deterioration with multiple states. The NBI deck information from year 1992 to 2010 in Wyoming are used to learn the deterioration distributions between different states. The most recent states of each deck are given by the inspection condition in year 2010. Inspected state at 2014 (accounts for two inspection-intervals in a regular regime) is used to validate the model performance. In total, there are 2249 testing data points in this case. Since there is no record of deck being in State 0 (S0) in the testing data set, only prediction performance of State 9 (S9) to State 1(S1) are considered here. The models are compared with several approaches:

- **HierBN**: the hierarchical BNs developed from Section 4.2 and 4.3.1 that learn between 27 groups separated by three selected features: ADT, age and maintenance, to give individualised predictions.
- **BN**: the BNs developed from Section 4.1.1 and 4.3.1 that learn the parameters using the overall population deterioration data to give aggregated predictions.
- **MCLR**: a Markov model, and its transition probability matrix is estimated using logistic regression. This model was developed by Chang [23] using the same NBI dataset to give aggregated predictions.
- **MCLR_G**: Developed by Chang [23] using the same NBI dataset, the bridges are separated into 20 groups, for each group, a Markov model, and its transition probability matrix is estimated using logistic regression, was used to give individualised predictions.
- **Mssurv**: a Markov model and its transition probability matrix is estimated using the Datta-Satten estimator. It allows modelling of censored data in a multi-state system developed by Ferguson et al. [47]. It is implemented here to give aggregated predictions.

To measure the accuracy rate, in each prediction, the state with the highest probability is considered as the predicted state. Therefore, we can compare the prediction against the actual observation. As a result, confusion matrixes for different approaches are generated.

Table 5.1 Confusion matrix: HierBN.

Observation	Prediction								
	S9	S8	S7	S6	S5	S4	S3	S2	S1
S9	**0**	0	0	0	0	0	0	0	0
S8	0	**28**	0	0	0	0	0	0	0
S7	0	12	**612**	2	0	0	0	0	0
S6	0	0	145	**658**	2	0	0	0	8
S5	0	0	21	131	**405**	0	0	0	0
S4	0	0	1	9	34	**104**	0	0	0
S3	0	0	0	2	3	7	**44**	0	0
S2	0	0	0	1	1	2	4	**11**	0
S1	0	0	0	0	0	0	0	0	**2**

Table 5.2 Confusion matrix: BN.

Observation	Prediction								
	S9	S8	S7	S6	S5	S4	S3	S2	S1
S9	**0**	0	0	0	0	0	0	0	0
S8	4	**8**	0	0	15	1	0	0	0
S7	1	5	**616**	0	5	1	0	0	0
S6	0	0	150	**660**	0	0	0	0	0
S5	0	0	25	130	**387**	0	0	0	17
S4	0	0	1	9	34	**48**	0	0	56
S3	0	0	0	2	3	6	**36**	0	9
S2	0	0	0	1	1	2	3	**8**	4
S1	0	0	0	0	0	0	0	0	**1**

Table 5.3 Confusion matrix: MCLR.

Observation	Prediction								
	S9	S8	S7	S6	S5	S4	S3	S2	S1
S9	**0**	0	0	0	0	0	0	0	0
S8	4	**8**	16	0	0	0	0	0	0
S7	1	5	**620**	0	0	0	0	0	0
S6	0	0	153	**660**	0	0	0	0	0
S5	0	0	23	130	**404**	0	0	0	0
S4	0	0	1	9	34	**101**	3	0	0
S3	0	0	0	2	3	7	**44**	0	0
S2	0	0	0	1	1	2	4	**1**	10
S1	0	0	0	0	0	0	0	0	**2**

Table 5.4 Confusion matrix: MCLR_G.

Observation	Prediction								
	S9	S8	S7	S6	S5	S4	S3	S2	S1
S9	**0**	0	0	0	0	0	0	0	0
S8	3	**13**	12	0	0	0	0	0	0
S7	0	7	**605**	14	0	0	0	0	0
S6	0	0	153	**660**	0	0	0	0	0
S5	0	0	22	131	**404**	0	0	0	0
S4	0	0	1	9	34	**102**	2	0	0
S3	0	0	0	2	3	7	**44**	0	0
S2	0	0	0	1	1	2	3	**12**	0
S1	0	0	0	0	0	0	0	0	**2**

Table 5.5 Confusion matrix: Mssurv.

Observation	Prediction								
	S9	S8	S7	S6	S5	S4	S3	S2	S1
S9	**0**	0	0	0	0	0	0	0	0
S8	2	**26**	0	0	0	0	0	0	0
S7	0	12	**404**	210	0	0	0	0	0
S6	0	0	134	**679**	0	0	0	0	0
S5	0	0	21	132	**404**	0	0	0	0
S4	0	0	1	9	34	**104**	0	0	0
S3	0	0	0	2	3	7	**44**	0	0
S2	0	0	0	1	1	2	4	**1**	10
S1	0	0	0	0	0	0	0	0	**1**

These matrixes give an overview of the performance of different approaches. The number marked in bold tells how many predictions match the observations for each state. By evaluating the performance purely from the quantity of correct prediction, the proposed approach HierBN gives the best prediction with 1864 correct predictions, closely followed by MCLR_G with 1842 and MCLR with 1840.

However, using a confusion matrix for a multiple states classification problem may create confusion in deciding the accuracy rate for each transition compares to a confusion matrix for a binary classification problem. For example, in the testing dataset, there are five decks start with S9, four of them transited to S8, and one transited to S7 in the second inspection. But we cannot evaluate the accuracy rate for Transition 9 (T9, the transition from S9 to other states) from the above confusion matrixes since records that start with S8 and S7 are overlapped. Hence, in addition to the confusion matrixes, an accuracy rate for each transition is also provided. For example, using HierBN, we predict all five decks started with S9 deteriorated into S8 in the second inspection, which gives us an 80% accuracy rate for T9 in HierBN. To provide a more in-depth evaluation for each approach, we also provide the overall average RPS as suggested in Section 5.4.1. The results are summarised in Table 5.6.

Table 5.6 Multi-state prediction performance comparison.

Methods	Accuracy rate								RPS
	T9	T8	T7	T6	T5	T4	T3	T2	
HierBN	**0.80**	**0.69**	**0.78**	0.81	**0.91**	**0.92**	0.92	**1.00**	**0.047**
BN	0.00	0.23	0.77	**0.82**	0.88	0.42	0.75	0.73	0.062
MCLR	0.00	0.40	**0.78**	**0.82**	**0.91**	0.89	0.92	0.09	0.053
MCLR_G	0.20	0.49	0.76	**0.82**	**0.91**	0.90	**0.94**	**1.00**	0.049
Mssurv	0.40	**0.69**	0.53	**0.82**	**0.91**	**0.92**	0.92	0.09	0.071

When predicting deck deterioration from states between S7 to S3 (i.e. T7 to T3), almost all methods have similar performance with relatively high accuracy rates. This is because the data amount within these states is considerably rich (over 300 data in each state). Hence, with enough data, these approaches perform similarly. Within these states, our approach guarantees a satisfying and steadily good performance compares to others. This is contributed by having successfully identified the small subgroups in our method and leveraged their deterioration learning specifically based on their features. Though the accuracy rates are not significantly higher than others, we suspect this is due to the population of the subgroup itself is small. Hence, its impact on both measurements is consequently low.

However, for data-poor states like State 9, 8 and 2, our model gave a promising performance compared to others. Though in these states, MCLR_G also gave decent accuracy rates comparing to MCLR, we believe this is benefited from its policy to group similar structures for estimating the transition probability matrix individually. But the reason that our model outperforms MCLR_G and others in most cases is that we not only consider separating the population into related groups but also learn from each subgroup.

Figure 5.13 presents a comparison of the distributions learned from BN and HierBN to the real observation. These decks belong to a group that their features ADT, age and maintenance are all rated as low-levels. The distributions showed in the figure are the pdf of Transition 8 (T8). Without grouping and learning, the learned distribution using a regular BN based on limited available data is showed with the purple bins, while the learned distribution of a HierBN is shown in yellow bins. The regular BN gives a relatively uninformative distribution with a wide range, but HierBN successfully predicts with a focus on early deterioration time that matches the pdf of the real observations (showed in red dotted line).

The last column in Table 5.6 gives the average RPS of the prediction of all the states. The scores are not drastically different since the majority of the structure population are between State 7 and 3 where almost all methods have good performance. The base of the other states' data is small; hence, the average performance of HierBN in RPS only excels slightly. Notes

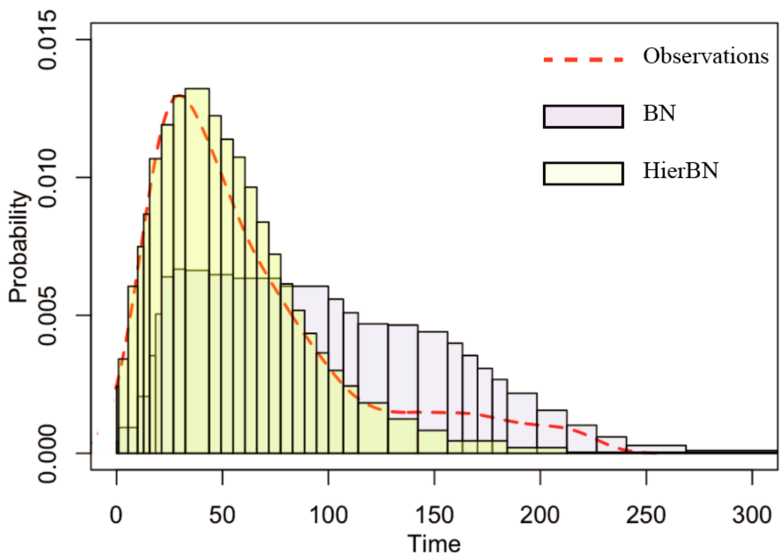


Figure 5.13 Probability density functions of Transition 8 (T8) of group rated as low-levels in all features.

that Mssurv has a decent performance in accuracy rates but poor in RPS, the reason we suspect is its predictions are usually too extreme and did not take too much uncertainty into consideration, which leads to a penalty in the RPS when the prediction is wrongly classified.

5.4.4 Future Prediction

This subsection investigates the prediction performance for future inspection. The training data for this subsection is same as the previous subsection, but we use data from year 2010 to 2017 to validate the performance of different approaches in predicting deck conditions over different future time length: from one year (year 2011) to seven years (year 2017).

The results are shown in Figure 5.14 measured by the average RPS of all states (the accuracy rates between methods are all quite close but with the same trending as RPS, hence not displayed). All methods have an increasing RPS over time; that is, the performance gets worse with the increase in prediction time. Our approach HierBN prevail other methods by having the lowest scores across these seven years. By comparing to the regular BN, we can also see the drastic improvement bring by separating the population into groups and learning between them. MCLR_G has a very close performance with the proposed approach. However, compare to MCLR, the benefit of grouping in MCLR_G is not significant in short future prediction but slowly increase over the years.

These experiments show we can provide a reasonably good prediction for a 2-year inspection (2012) with an RPS of 0.021 and an average accuracy rate of 87.6%, and a 4-year

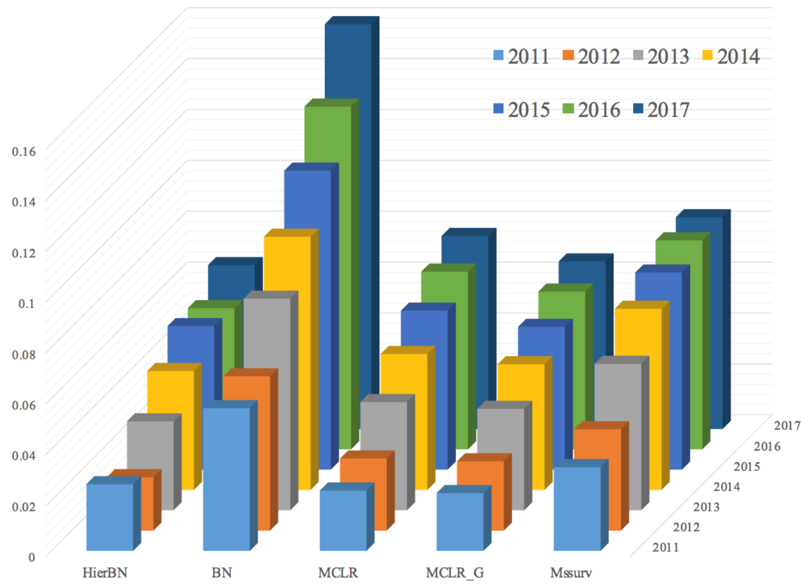


Figure 5.14 RPS (lower better) of different approaches with the increase in prediction time.

inspection (2014) with an RPS of 0.047 and an average accuracy rate of 77.5% in HierBN. This counts for one and two biennially inspections in a regular inspection scheme respectively, which is valuable information for future inspection planning and resources scheduling.

5.5 Summary

This chapter validates the deterioration prediction models developed in Chapter 4 using synthetic data and real-world data to fulfil Objective IV.

We first produce a synthetic dataset to show that, with enough data, it is possible to learn the parameters of a deterioration distribution. However, when there is less data relevant knowledge about the parameters in the form of Bayesian priors helps to improve the performance. We also simulate data from a mix of deterioration distributions and show that we can provide a better prediction if we can separate assets into groups, where each group follows one of the distribution. But the challenge is to know how to classify assets into groups in the first place.

We tackle this challenge using the features associated with each asset and explain the processes using the NBI database case study. First, we introduce the background of the NBI database and its challenge of having limited deterioration data in some cases. To separate assets into groups, the features that have the most impact on the deterioration time should be identified. We perform dimensionality reduction to identify the key features for predicting asset deterioration using a modified random forest. The values of features are used to separate assets into groups. We use a hierarchical BN developed in Chapter 4 to learn both the hyperparameters of the overall population and the local parameters within each group. The model can also learn the weights of each feature for influencing the deterioration rate. The transitions distributions learned in this way are further assembled to provide individual multi-state prediction given the state of an asset observed in the latest inspection and inspection time.

Lastly, we measure the performance of the developed deterioration prediction models using the NBI data. We first evaluate how many features to consider in the models. The example shows that the performance improves at a rate up to three features; after this, more features can only improve the performance slightly. We also compare the performance of our models with other available approaches for correctly predicting the condition of a multi-state system. Our results show that our proposed models excel at most predictions, especially for cases where there is little data. We also show that as the prediction time increases, the accuracy of the prediction drops. The proposed models can provide reasonably good accuracy over 1 or 2- biennially inspections in a regular inspection scheme, which could be useful for inspection planning.

Chapter 6

Inspection and Maintenance Decisions Support

The challenges in making inspection and maintenance decisions are to inspect a suitable asset and perform an appropriate repair action at a suitable time. Understanding asset states from its deterioration and the effectiveness of repair actions are the foundation to recommend these decisions. A range of techniques was proposed to support these decisions, and some of them have been implemented in industry. But as discussed in Section 2.3.1, classical deterioration models cannot properly handle uncertainty and situation with little deterioration data. This problem is common for critical infrastructure like bridges, where deterioration data are often uncertain, and data are rare (Section 2.2.1). Another challenge is from the complexity of the modelling with various assumptions (Section 2.3.2 and 2.3.3) and further, to reason inspection and maintenance decisions (Section 2.4) from the models, where classical approaches often lead to an unmanageable model size that becomes difficult to perform analyses.

Validated in Chapter 5, models build with a Bayesian framework offer the flexibility in handling uncertainty and can learn deterioration from data and expert knowledge even in the case with small data amount. Pointed out in Section 4.3, we can also use the models to represent various system configurations and extend them for complex and large-scale problems (later showed in Chapter 7). This study does not have access to conduct expert knowledge elicitation. Instead, this chapter shows how these models can be built using example knowledge from published documentation to support a variety of decisions, from inspection decision, repair decision, to maintenance planning using real-world case studies (Objective V).

The remainder of this chapter is as follows: Section 6.1 presents the use of deterioration models in evaluating asset conditions. It is illustrated by a GB bridge example and a US bridge example, where assets are assembled by multiple components with different system configurations. The US example is continuously studied in the following sections: for inspection decisions in Section 6.2, repair decisions in Section 6.3 and for strategic maintenance planning in Section 6.4. Section 6.5 concludes this chapter.

6.1 Condition Prediction and Structural Evaluation

With the deterioration prediction models from Chapter 4, we can predict the condition of components or assets, and their performance is validated in Chapter 5. This section introduces two applications of them, where the predicted conditions of two assets are determined by the condition of their components but with different system configurations.

6.1.1 Bridge Condition Prediction in Great Britain

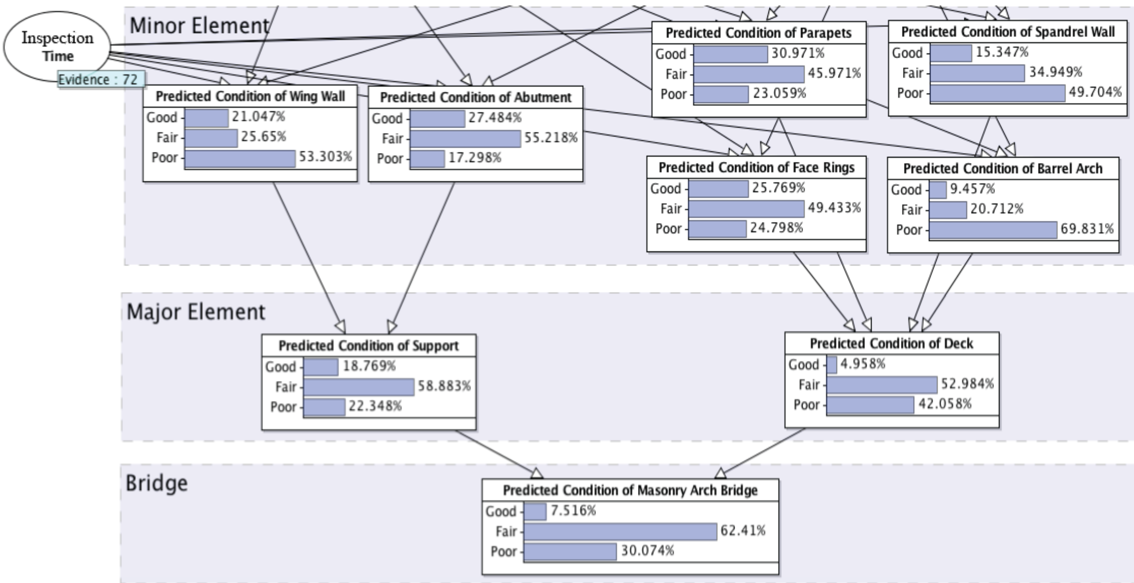
In Great Britain, the condition of each bridge component (element) can be modelled as a three-state variable, namely, poor, fair and good (see Section 2.1.2). Two failure transitions, from good to fair and fair to poor, are modelled for each component with an assumption that the time to failure of each transition follows a Weibull distribution. The parameters of the distribution are learnt from data and expert knowledge, using the models described in Chapter 4.

Bridges can be decomposed into major elements, such as a deck, a superstructure and a substructure, and each of them can be further subdivided into minor elements, such as an abutment and wing walls (see Section 2.1.1). Different types of bridges have different major and minor elements, and even when two bridges are of the same type, the number of elements may vary. Table 6.1 shows an example of elements of a masonry arch bridge presented in Rafiq et al. [144] (also see Figure 2.2).

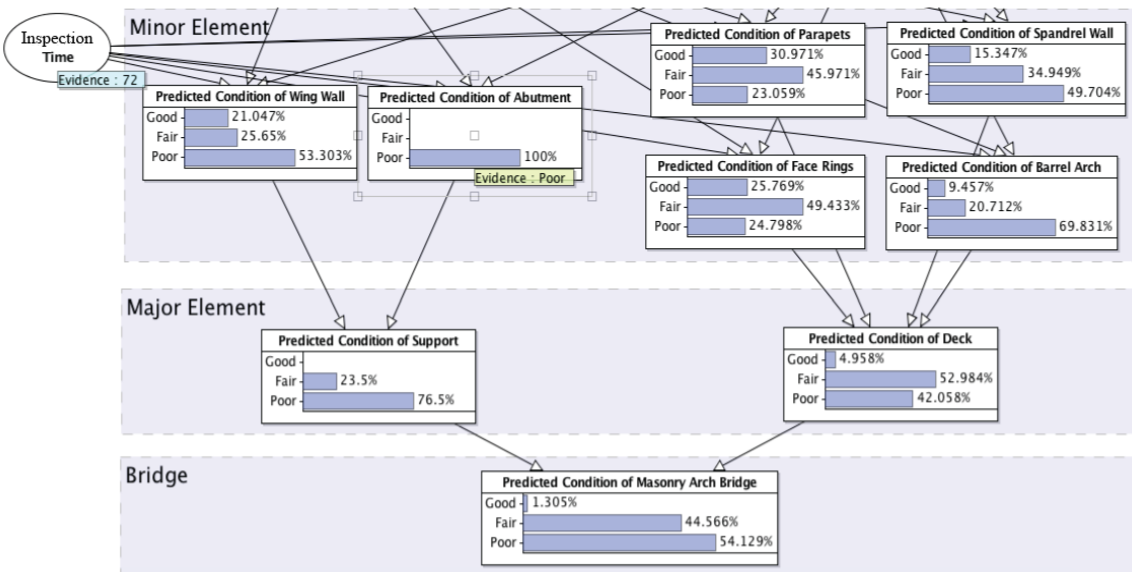
Table 6.1 Elements in a typical masonry arch bridge from Rafiq et al. [144].

Bridge	**Major Element**	**Minor Element**	**Weight**
Masonry arch bridge	Support	Wing wall	5
		Abutment	10
	Deck	Parapets	2.5
		Spandrel wall	3.5
		Face rings	3.5
		Barrel arch	10

This example is a typical bridge structure system configuration (see Section 2.3.3), where each element is assigned with a weight representing its effect on bridge condition given by



(a)



(b)

Figure 6.1 (a)Condition prediction of a typical masonry arch bridge six years later. (b)Condition prediction of a typical masonry arch bridge six years later given abutment is in Poor condition.

experts. To use these weights in our models, we need to calculate the relative weight of each element. For example, one of the major elements is the support, with minor elements wing walls and abutment, their weights are 5 and 10 respectively, that is, we have the relative weights of 0.33 and 0.67 when we consider the support as a subsystem. From Equation 2.15, the condition of the major element (e.g. support) is modelled by a TNormal distribution in a ranked node with a mean using a weighted mean function of these relative weights and condition of components, and variance about the confidence levels of experts when assigning these weights. The bridge condition is then further aggregated follow the same principle with a weighted combination of its major elements' conditions.

For this example, a complete BN is assembled from six minor elements, learnt from two groups of assets; the resulting model has 126 variables in total , of which, 79 of them are observations (see Figure 6.1, the deterioration learning models are omitted for display reasons). The inference of such model took approximately less than 2 minutes using AgenaRisk's API. To build such a model in general, we first need to determine the elements of the bridge (e.g. information from Table 6.1) and model each as described in Section 4.1.1. Then, if the target group lacks sufficient deterioration data, we can look for similar groups and pool the deterioration data using the model of Section 4.2. Finally, the model from Section 4.3 is applied to assemble the element condition models to assess the condition of the target bridge.

Assume all elements are currently in perfect condition, with the predicted condition of each minor element in Figure 6.1 (a), we can estimate the condition of this bridge: it has a 62.41% in Fair condition after six years (72 months showed in the figure). Therefore, we do not have a strong urgency for detail examination. Nevertheless, it would be a different story if we have evidence of some element's condition. For example, given the observation that the abutment is in Poor condition from the latest visual inspection, if no repair is performed on the abutment, in the next six years, the bridge would become vulnerable with a 54% in Poor condition (see Figure 6.1 (b)). This result suggests us to intervene in the abutment to mitigate this potential risk.

6.1.2 NBI Structural Deficiency Evaluation in the US

According to the Federal Highway Administration Bridge Preservation Guide, one of the common failure modes for bridge deficiency is structural deficient [44]. As of 2017, 8.87% bridges (54,560 out of total 615,002 bridges) in the NBI database are classified as SD. A deficient bridge has a higher possibility than a normal bridge in leading to a major structural event, such as bridge collapsion. This suggests the importance of imperative interventions.

In this subsection, we would like to use the predicted condition of each structure to estimate the probability of the bridge deteriorates into SD in a future time.

A highway bridge is defined as SD either (the relevant items are encoded in Weseman [182]):

- A condition rating of 4 or less for any of its structures, including deck (Item 58), superstructure (Item 59), substructure (Item 60) and culvert (Item 62), or
- An appraisal rating of 2 or less for structural evaluation (Item 67), or
- An appraisal rating of 2 or less for waterway adequacy (Item 71).

One of the criteria in resulting SD is any of its structures are considered as in Poor condition. We can, therefore, consider this bridge as a system with a series of subsystems arranged in series. Accordingly, we can model it using the OR gate from Section 4.3, where the system fails when any of the subsystems fail.

The structural evaluation is used to evaluate if the bridge's loading carrying capacity is significantly below its design standards. For structures other than culverts, the rating of structural evaluation is specified by the lowest rating between superstructure, substructure and a comparing rating by comparing ADT (Item 29) and inventory rating (Item 66). ADT and inventory rating are both fixed variables, where ADT encodes the average daily traffic volume of the bridge, for example, volume 3222; inventory rating represents a load level which can allow a structure to operate for an indefinite period of time, for example, MS 12.5 (MS is a measure for loading capacity). By using the specification table provided in Weseman [182], we can gather the comparing rating, for example, a rating is coded as 6 given 3222 ADT and MS 12.5 inventory rating. Since the determination of the structural evaluation rating is the lowest score among those ratings, we can also model these criteria using an OR gate.

A highway bridge is also SD when its waterway frequently overtops the bridge. The inspector often evaluates it from the on-site inspection. Similar to structures like a deck, we can learn its patterns from the inspection data and predicted the condition distribution of its adequacy in a future time.

Together, we can use a logical expression to combine the information by a Boolean node. It is true when one of the structures are rated 4 or less, or structural evaluation is rated 2 or less, or waterway adequacy is rated 2 or less. The example bridge in Figure 6.2 is a non-culvert bridge that has an average daily traffic volume of 3222, and a permitted loading capacity of MS 12.5. Its current conditions of deck, superstructure, substructure and waterway are rated as 7, 6, 7 and 5 respectively. The complete model of this example contains

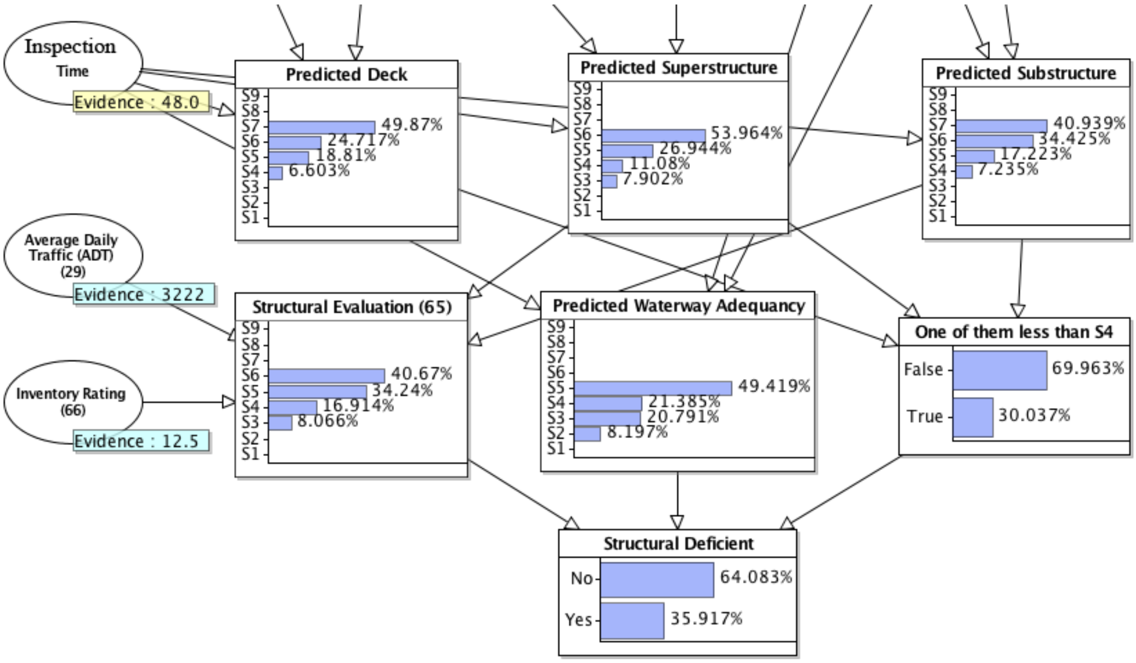


Figure 6.2 Structural deficiency evaluation.

more than 1000 variables where most them are observed. The calculation of this model was done using separate submodel for each transition, which varies in size and computation time. For example, it took less than 1 minute to compute the transition of deck condition from S7 to S6, while it took almost 1 hour for the transition of deck condition from S5 to S4 as it requires to learn from other groups.

This model predicts the possibility of this bridge being structural deficient in the next 48 months. The result shows that the probability is 35.92%. By setting a limit, for example, we need to intervene if the percentage of being a structural deficient is higher than 10%, we can, therefore, evaluate if this is a bridge with a low-risk level. Further, by reasoning with this model, we can recommend inspection decisions, including the time to inspect, and the priority (among other bridges) for inspection, which will be discussed in the following section.

6.2 Inspection Decisions Reasoning from Condition Prediction

This section focuses on the use of condition prediction to support inspection decisions. With the deterioration prediction models, we can predict the condition distributions of a structure over a finite time. This structure could be a component, or an asset assembled by multiple

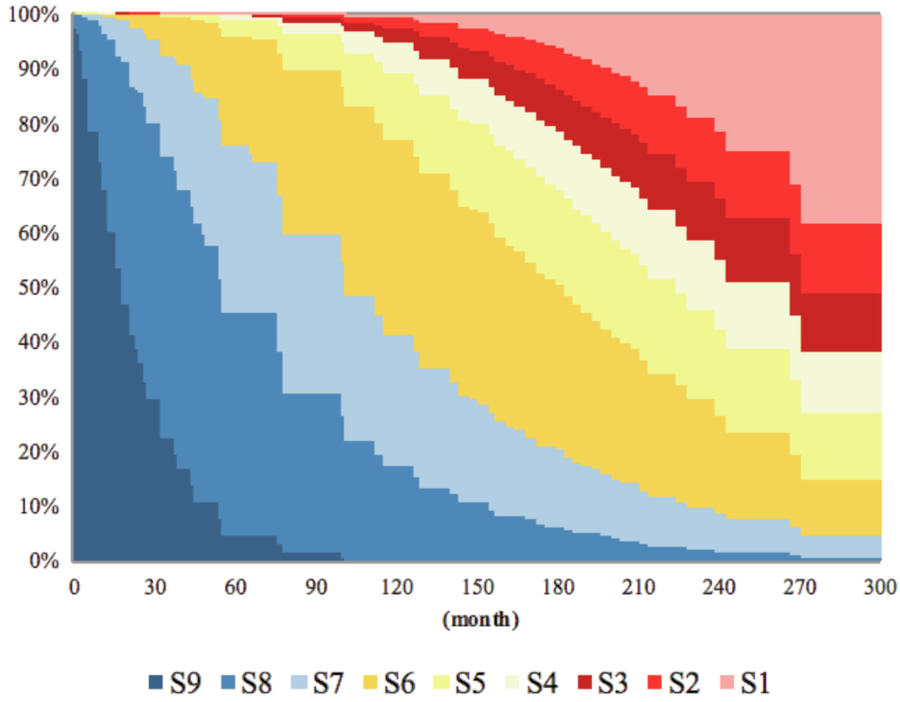


Figure 6.3 Deck condition distribution over time.

components as we showed in the last section. Figure 6.3 shows a prediction example of a new deck's condition distribution over the next 300 months without interventions. The deterioration is learned from the NBI dataset. With the increase in time, it is natural to see the trend of deterioration: the probability of S9 (perfect state) gradually decreases with the increase of S8, and S8 decreases with the increase of S7, and so on. For example, in around 90 months, the possibility of this deck still in S9 is nearly 0%. However, sometimes, decision-makers are more interested in knowing how reliable this structure is over some time. This information can be used to identify assets with higher risk and allow the decision maker to plan accordant action to mitigate the risk.

Failure rate function $f(t)$ is used to measure how fast a structure will deteriorate. By taking its integral, we have its cumulative distribution function $F(t)$ represents the failure probability that the random variable t takes on a value less than or equal to t. Since we are more interested in how reliable the structure is, as in Equation 2.9, we can take the complementary cumulative distribution function to form the reliability function $R(t)$. The reliability function measures the probability of an asset did not fail before moment t. For failure distributions like Weibull distribution, they are asymmetric. Therefore, reliability measures such as MTTF or MTBF (see Section 2.4.1), are therefore less interesting in describing its true behaviours. Instead, we are more interested in the overall reliability.

For a multiple state system (for example, bridges in GB and US are both rated with multiple states), to evaluate the reliability, first, we need to define a level of acceptable conditions. For example, defined in Federal Highway Administration [44], in the NBI dataset, structure states with a rating of 7, 8, or 9 are classified as Good condition, with a rating of 5 or 6 are classified as Fair condition, and the rest are classified as Poor condition (SD). This type of information can be used as an acceptable level for assets rated by multiple states: to measure the reliability of staying in these conditions without failing to other levels.

With this information, we can build a Boolean node (a constraint) as a child node of the predicted condition. The logical expression of the Boolean node is that this node is observed as true when the predicted condition is better than the acceptable level, such as predicted condition $\geq S7$ for structure staying in Good condition, or predicted condition $\geq S5$ for the structure staying in Good or Fair condition. The predicted condition could be a condition of a component, or a condition of an asset that is determined by its components. By entering an observation on this constraint as True, and remove the observation on the node Inspection Time, with the Bayesian inference, we can get the reliability $R(t)$ of this structure from the posterior distribution of node Inspection Time.

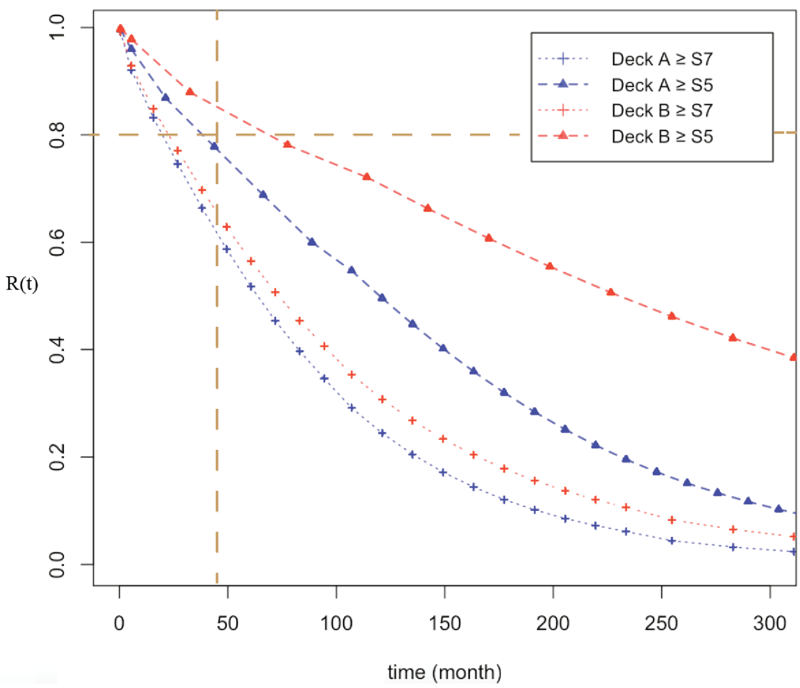


Figure 6.4 Reliability of decks.

Figure 6.4 shows four examples of two decks with two acceptable levels. One level has a higher standard that requires the structure in Good condition ($\geq S7$), and the other one accepts the structure in or above Fair condition ($\geq S5$) with lessening constraint. Naturally, within the same decks, the reliability of a deck in or above Fair condition is always higher than the one in Good condition. For example, in 48 months, Deck A condition $\geq S5$ has a reliability 0.78, condition $\geq S7$ has a reliability of 0.61; Deck B condition $\geq S5$ has a reliability 0.85, condition $\geq S7$ has a reliability of 0.63. Suggested in Section 2.4.1, with the reliability, we can make inspection decisions including to prioritise assets for inspection and to determine a suitable time for inspection.

To prioritise assets for inspection based on their risks in deteriorating into an unacceptable level, we need to set a failure limit first. An example of a failure limit is set at 0.8 in Figure 6.4, where we want to ensure the reliability of the decks is above 0.8. To ensure deck condition $\geq S7$, we can see both Deck A and B require intervention around 25 months. However, for condition $\geq S5$, we can see an apparent distinction between Deck A and B: Deck A requires intervention at around 40 months while B at around 70 months. This distinction reveals that Deck A has a higher risk in failing to Poor condition than Deck B. We can, therefore, to prioritise the inspection for Deck A over Deck B.

Another decision is to evaluate the suitability of inspection time. Most bridges in the US are inspected every 24 months, while in GB, bridges are visually inspected every 12 months and detailed inspected every 72 months. Most inspections are performed at fixed inspection time intervals in current practice. We can improve this situation by tailoring inspection time for each structure using the reliability function inferred from the developed models. Using the same example in Figure 6.4, here assumes a failure limit of reliability is set as 0.8 from experienced engineers. To prevent deck condition from failing to Poor condition, Deck A urges for an inspection to see if intervention is needed at around 40 months, while Deck B is at around 70 months.

6.3 Repair Recommendation and Evaluation

Given an asset's condition, either from inspection or prediction, we may want to recommend repair action to satisfy a range of objectives. It is a challenge since a maintenance action may only be suitable for a specific condition of a particular asset. Different repair actions may also come with different restoring capability and cost. With the observational model from Section 4.4, by extending it with a set of constraint nodes, we can reason the frequency of each repair decision made in history. The decision maker can refer to it as what action is usually taken historically. Another decision is to evaluate the effectiveness of maintenance, which is enabled by the intervention model from Section 4.4. Decision maker can use this evaluation to understand the potential repaired condition and plan ahead about what resources are needed.

These repair decisions are introduced aligned with the background of bridge deck maintenance in the US. Table 6.2 from Michigan Department of Transportation [116] is a typical guideline for maintenance of concrete bridge deck that has epoxy coated rebar. This table is built upon expert knowledge about deck deterioration from experienced engineers. Together with the deck condition, this table can provide reasonable guidance for repair action selection.

Table 6.2 Bridge Deck Preservation Matrix – Decks with epoxy coated rebar [116].

Deck Condition		Repair Options	Potential Result		Expected Life (years)
Top	Bottom		Top	Bottom	
≥ 5	N/A	Hold or Seal Cracks	No Change	No Change	1 to 4
	> 5	Epoxy Overlay	8, 9	No Change	10 to 15
	≥ 4	Deck Patch	Up by 1	No Change	3 to 10
4 or 5	4	Shallow Concrete Overlay	8, 9	No Change	20 to 25
		HMA Overlay with Waterproofing Membrane	8, 9	No Change	8 to 10
	2 or 3	HMA Cap[a]	8, 9	No Change	2 to 4
≤ 3	4 or 5	Shallow Concrete Overlay	8, 9	No Change	10
		HMA Overlay with Waterproofing Membrane	8, 9	No Change	5 to 7
	2 or 3	HMA Cap[a]	8, 9	No Change	1 to 3
		Replacement with Epoxy Coated Rebar Deck	9	9	60+

[a] Hot Mix Asphalt (HMA) Cap for deck improvement. After HMA Cap, deck replacement should be planned in the next five years.

Table 6.3 Repair cost for bridge deck with epoxy coated rebar [184].

Repair Options	Unit Price	Notes
Patching	$\$32/ft^2$	Includes hand chipping
Epoxy Overlay	$\$3.80/ft^2$	
HMA Overlay/Cap	$\$1.25/ft^2$	An extra $5/ft^2 if the repair includes waterproofing membrane
Shallow Concrete Overlay	$\$25/ft^2$	Includes joint replacement and hydro
Deck Replacement	$\$70/ft^2$	Includes removal of the old deck and new railings

The bridge deck condition is evaluated by the conditions of its top surface and its bottom surface. Available maintenance actions vary from its current condition, and for example, deck patching is only applicable when its top surface is rated not less than State 5 and bottom surface not less than State 4. The repairing capacity also varies, for example, deck patching can improve the condition of a deck top surface by 1 point with an expected service life of 3 to 10 years. While for shallow concrete overlay, it can improve top surface to State 9 or 8 with an expected service life of 20 to 25 years when it was rated at a State 4 or 5; 10 years when it was rated at a state less than 3. However, most actions cannot repair the deck bottom surface - it can only be replaced to improve its condition. Additionally, Table 6.3 shows the related repaired costs listed in Winn and Burgueño [184], which were collected from the authors' personal communication with the engineers. Information about the area of an individual deck can be retrieved from the NBI database: structure length is recorded in Item 49 and deck width is in Item 52 (see Weseman [182] for details).

6.3.1 Observational Model: Historical Frequency of Repair Action

The CPTs in the observational model from Section 4.4 can be collected from Table 6.2. The repair action is determined by the condition of the deck top surface and the bottom surface simultaneously, where each repair action has an equal probability if there are multiple applicable options under the same required conditions. For example, if the condition of deck top surface is rated 5 and the bottom surface is rated 4 in the most recent inspection, the applicable actions include deck patching, shallow concrete overlay, and HMA overlay with a waterproofing membrane. These three actions each has a 1/3 probability, and other actions have 0 in the CPT. The CPT for the repaired condition is generated by columns under potential result in the table, with a uniform distribution if there is more than one possible result. For example, if an epoxy overlay is performed, the top surface will restore to condition 9 with a 0.5 probability and condition 8 with a 0.5 probability. However, the bottom surface will stay as the same condition from the last inspection since this repair action does not affect it. Similarly, the expected service life is modelled using a partitioned expression from the repair actions and top and bottom deck surface conditions. For example, shallow concrete overlay action on deck top surface with a condition 4 and bottom surface with a condition 4, it has an anticipated service life that expressed by a uniform distribution with a lower bound of 20 and upper bound of 25. While for a deck with a condition 3 on its top surface, its anticipated life is only 10 years.

To reason repair action that is usually taken in history, three types of constraints are modelled, showed in Figure 6.5. These constraints are all modelled using Boolean nodes with logical expressions embedded. The structural deficiency is true when either the repaired deck

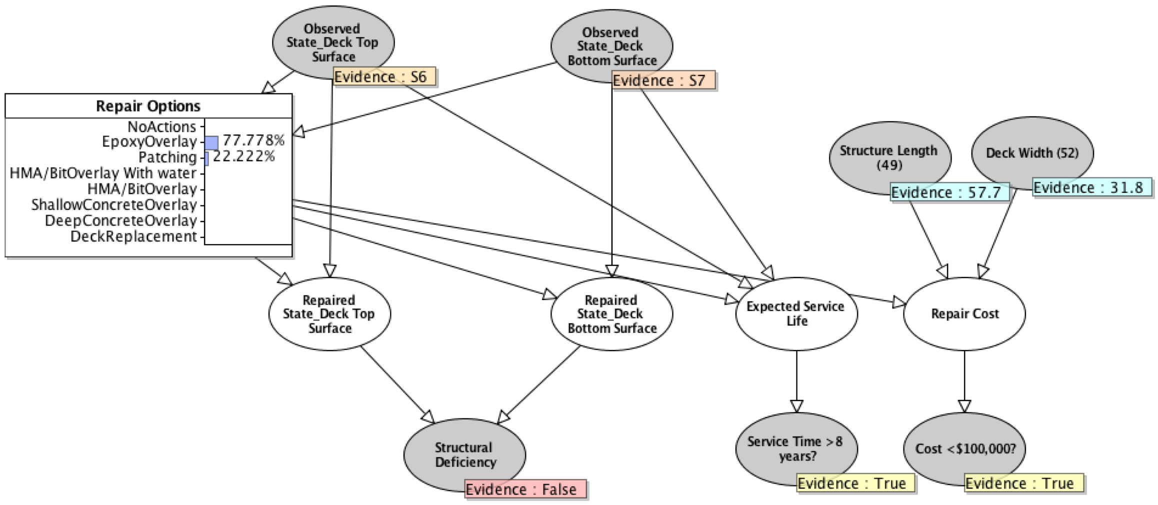


Figure 6.5 Maintenance action recommendation.

top or bottom surface has a rating less or equal to 4. The decision makers set the expected service life about how many years they anticipate the deck will survive. The cost of repair action can be modelled using a partitioned expression, where each repair option is modelled by the compound of cost per unit (see Table 6.3), and its area. Decision makers suggest the cost constraint about how much budget is allowed.

The example in Figure 6.5 is based on a deck with a top surface that is inspected with a rating of 6 and a bottom surface with a rating of 7. The length of this deck is $57.7ft$ and width is $31.8ft$. The decision maker does not want the repaired deck to become structurally deficient and wishes the service time to be greater than eight years with a cost less than $100,000. The model contains only 12 variables with 7 observations, and its inference took seconds. After the inference of this model, in history, under the same constraints, 77.78% of maintainers repaired with epoxy overlay and 22.22% repaired with deck patching. This model rules out repair options that weren't taken by other maintainers in the past automatically, which makes the decision-making process for evaluating repair action easier. But if we want to estimate the effectiveness of different maintenance action directly, an intervention model is more suited.

6.3.2 Intervention Model: Effectiveness of Maintenance

Decision maker can prioritise repair actions based on the result of the intervention analysis. In Table 6.2, interventions for deck condition are either imperfect maintenance or perfect maintenance. For example, HMA Cap action can imperfectly restore deck top surface to State 8 with a 0.5 probability and State 9 with a 0.5 probability while the replacement is a perfect maintenance action that can restore both deck top and bottom surface to as good as new condition. To reason the effectiveness of repair actions directly, in the intervention model, we want to manipulate intervention independently from its causes.

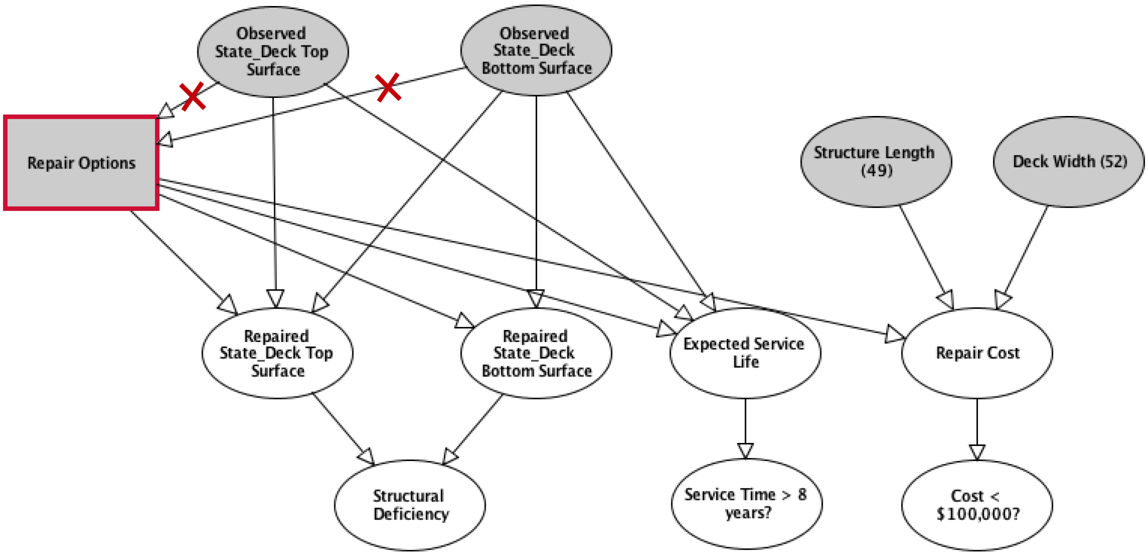


Figure 6.6 Intervention model for maintenance effectiveness.

To convert an observational model into an intervention model, we need to remove the edges from causes that enter the intervention, as demonstrated in Figure 6.6. The reason is, in the observational model, the use of repair action explains the historical frequency of actions taken under the observed conditions [28]. The repair options become a set of maintenance suggestions that are distributed probabilistically inferred from the posterior probabilities. In the intervention phase, the repair action is usually observed (hard evidence): only one action can be performed in each intervention. Only by removing the links from the parents we can remove the implication from past decisions, and directly show the effectiveness of repair action on manipulating structure repaired condition. At the same time, the constraints from the observational model all become unobservable in the intervention model to show the associated result of repair actions.

6.4 Maintenance Planning and Its Evaluation

Planning maintenance over a finite time horizon is valuable in the maintenance strategy design phase, and this plan can be used to allocate resources, make investment decisions and identify most cost-effectively maintenance choices. This section aims to utilise the developed models and extend them to evaluate the effectiveness of different maintenance plans. We present a simplified maintenance plan and show how to use it to analyse the life-cycle cost of the maintenance as an example.

Since inspection and maintenance activities for bridges are performed periodically with time intervals as discussed in Section 2.2.1, to perform life cycle cost analysis over a finite time horizon on bridges, we can discretise the time horizon into several cycles, where each cycle models an intervention process. For simplicity, this section illustrates this idea using a fixed interval of every two years over a hundred-year lifetime. However, notes that it is possible to implement with a dynamic intervention schedule by integrating with an optimisation technique.

In each cycle, we model the repair decision-making process together with its deterioration prediction up till to its next intervention cycle. Intervention model from Section 6.3.2 is employed to infer the effectiveness of maintenance. The choice of repair option is a decision node that relies on the anticipated conditions post-repair. Since we cannot directly observe the future condition of a structure as we showed in Figure 6.6, instead, we predict its future condition by learning their deterioration. By repeatedly linking the posterior of each cycle's model as the prior of the next cycle's model using the model from Section 4.5, the maintenance whole life cycle is modelled.

We continue using the deck repair example from in the section. In general, a deck bottom surface deteriorates slower than its top surface. Among the available repair actions, deck replacement is the only action in Table 6.2 that can improve the condition of the deck bottom surface, but it is also the most expensive action according to Table 6.3. Hence, in the example maintenance plan, deck replacement is assumed to be the last resort of maintenance. Instructed by Michigan Department of Transportation [116], its trigger is when both deck top and bottom surface are deficient with a rating of less or equal to 3 and a rating of 2 or 3, respectively. Similarly, the use of other repair actions requires both top and bottom surface are within the range of deck condition listed in Table 6.2. Notes that when deck top surface is rated less or equal to 3, HMA Cap repair action shares the same trigger conditions. As instructed from Michigan Department of Transportation [116], since HMA Cap is a temporary repair with low expected service life, a replacement is often scheduled in the next five years after an HMA Cap is performed.

Within an asset's whole life cycle, a structure may be maintained by multiple actions. A maintenance strategy is used to decide what action to take at what time. Here we use a simple strategy as an example to show how we can evaluate the cost of different maintenance actions over its lifetime. In this maintenance strategy, if the deck conditions fall into the trigger condition of the repair options, a repair action from Table 6.2 will be performed; otherwise,

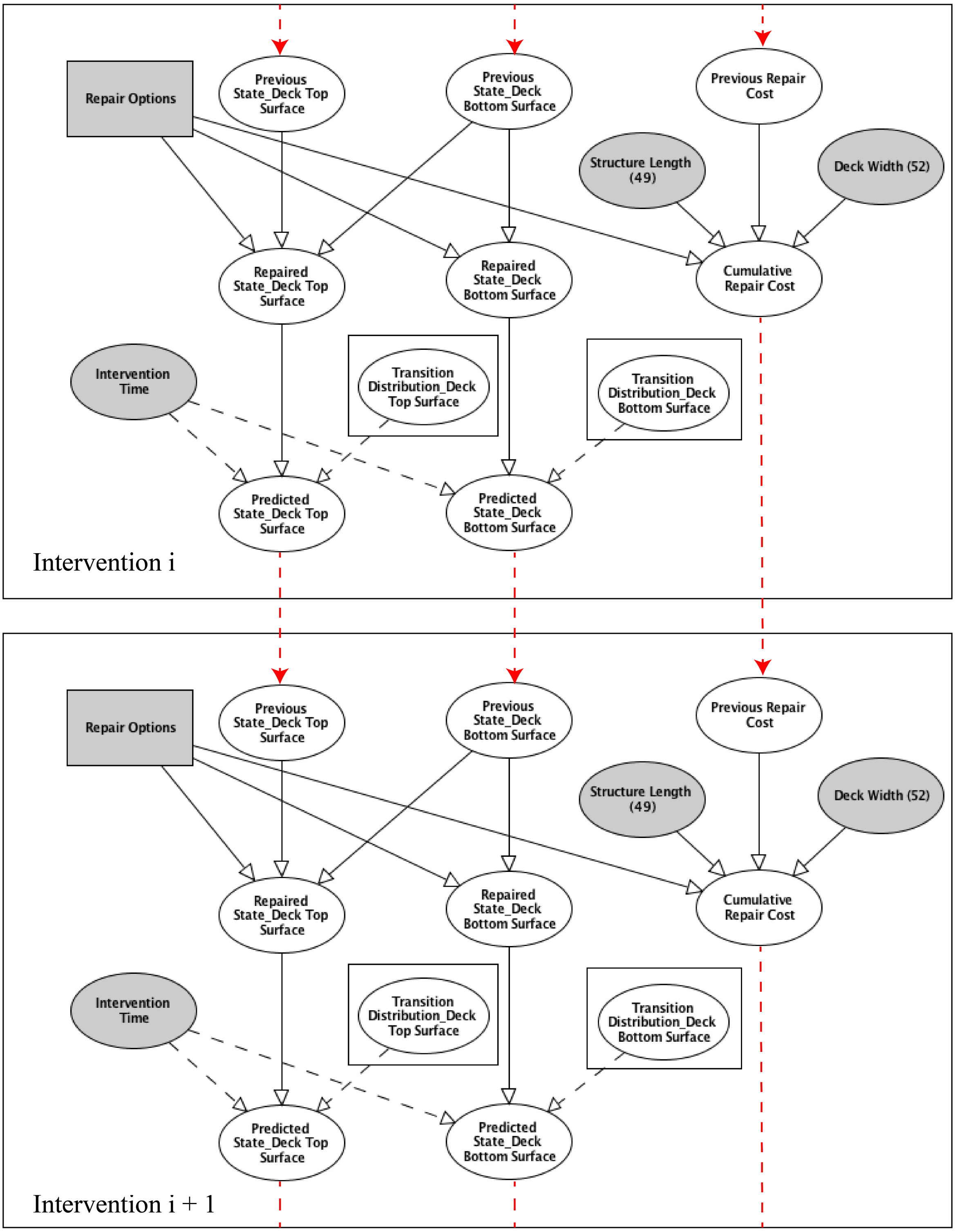


Figure 6.7 Life-cycle cost analysis from multiple sequentially organised BN objects.

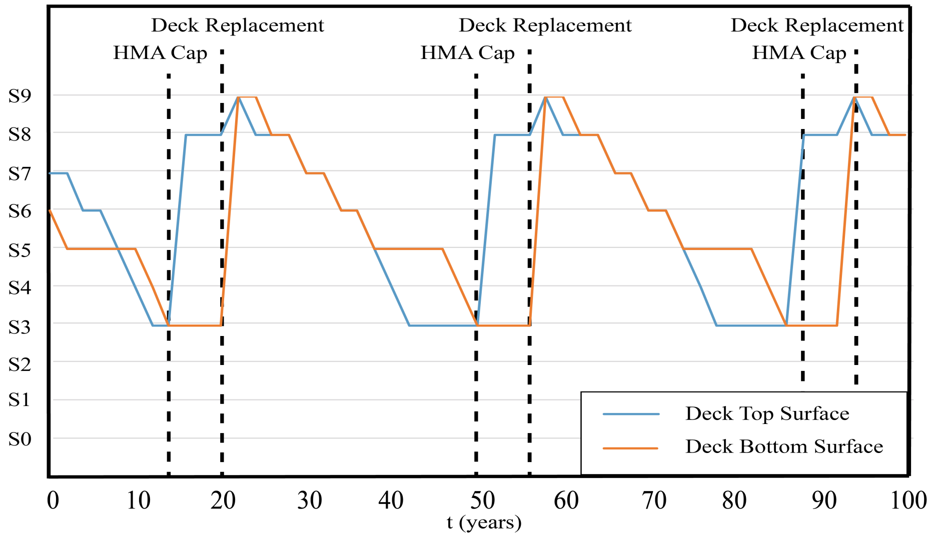


Figure 6.8 Deck top and bottom surface conditions with repairs in a 100 years horizon.

if the deck conditions satisfy the trigger condition of replacement, deck replacement will be performed; otherwise, no action will be taken.

In each cycle, depending on the number of required transitions, the model contains observations ranging from 100 to more than 1000 with a computational time from seconds to overnight. After the inference on the repaired condition, we can further predict its condition up to its next intervention. To model multiple intervention cycles sequentially, we can link the predicted condition in this intervention cycle as the input of the following intervention cycle. This can be modelled by multiple sequentially arranged BN objects from Section 4.5, where the predicted state at intervention i becomes the prior of previous state variable at intervention $i+1$, and used to reason the repaired condition as well as to predict its further deterioration as shown in Figure 6.7. Unlike the model in Figure 6.6, where the previous structure state is observed as hard evidence (e.g. 100% at S6). In each object, the previous state is a future event that cannot be observed directly but only predicted probabilistically as soft evidence (e.g. 80% at S6 and 20% at S5). This problem is resolved by the employment of multiple binary factorisations discussed in Section 4.5, where each state's further deterioration is modelled as a Markov model. The cost of all actions is recorded using a cumulative cost function and sequentially becomes the prior of the previous repair cost node for the next intervention.

Figure 6.8 shows an example of condition changes of the deck top and the bottom surface (with an initial state of S7 and S6 respectively) in the next 100-year horizon. The maintenance strategy comprised of maintenance action HMA Cap and deck replacement. If the state of deck top surface deteriorates to a state of 4 or 5 and the bottom surface is in a state of 2 or

3, or the deck top surface deteriorates to a state less or equal to 3, and the bottom surface is in a state of 2 or 3, HMA Cap is triggered. Since HMA Cap is a temporal repair action with a short expected service life of 1 to 3 years, as the footnote in Table 6.2 suggested, the deck is scheduled for replacement in the next five years. For example, in 14 years, the top surface is predicted with a state of 3 the same as the bottom surface. Therefore, an HMA Cap is performed that restores the condition of the top surface to 8 while the bottom surface remains the same. After five years, the deck is replaced with both top and bottom surface back to state 9. In the next 100 years, implementing this maintenance strategy requires three HMA Cap actions and three replacements. Together with the cost presented in Table 6.3, in this example, for a 57.7ft length and 31.8ft width deck, we estimated it has an overall repair cost of \$ 392,201.33.

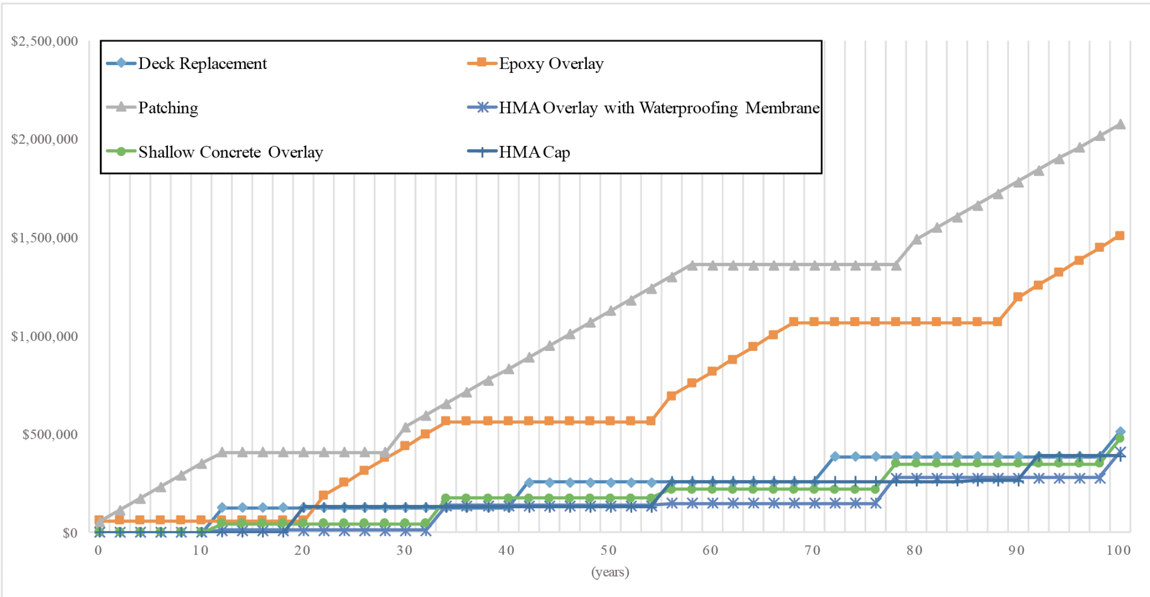


Figure 6.9 Repair cost with different maintenance strategy in the next 100 years.

Different maintenance action-based strategies are evaluated, and their cumulative costs are presented in Figure 6.9. Epoxy Overlay and patching are the two most costly strategies since they prefer to repair the deck top surface while they are still in fair conditions. Other strategies share similar costs; among them, HMA Overlay with waterproofing membrane strategy has the lowest cost. Though deck replacement strategy (only repair with replacement) has a lower cost than some strategies, the deck stays in poor condition most of the time. In some cases, keeping structures staying in better conditions throughout its lifetime, even in the cost of a higher expenditure, is still desired due to safety and reliability reasons.

The repair decisions in this example are simplified, only a limited type of actions are considered in each strategy, and the repairs are triggered by the asset conditions only. However, usually, there are more constraints in deciding repair decisions, such as repair action availability, cost and reliability - this form a multi-criteria decision problem. Our model gives a framework to integrate deterioration prediction with repair decisions. Together with an optimisation technique, by providing a set of constraints, we can make optimal decisions for maintenance. For example, along with the model shown in Figure 6.7, we can use a Genetic Algorithm (GA) to generate a maintenance strategy that takes both asset deterioration and maintenance effectiveness into consideration. We can therefore optimally select different repair actions to restore an asset condition to a suitable condition at an appropriate time with an objective of minimum cost and high level of reliability. The strategy can be implemented in the decision node Repair Options in Figure 6.7, and further used to estimate its life cycle cost for maintenance planning.

6.5 Conclusion

This chapter uses the BNs proposed in Chapter 4 for making inspection and repair decisions as well as maintenance plans. Objectives V is fulfilled by the case studies of rail bridge maintenance in GB and NBI bridge assets in the US. These models support decisions in several cases including:

1. Multi-state condition prediction and structural evaluation for assets with components that contribute in different ways to the overall state.
2. Prioritising the inspection of an asset to inspect by performing a reliability analysis from the predicted condition and suggesting inspection times given a reliability threshold.
3. Choosing a repair given an inspection result, either based the repair action usually performed for an asset in a similar state, or by evaluating the effectiveness of different repair options.
4. Evaluating the effectiveness of different maintenance strategy over a time horizon by combining models of repair and deterioration over several maintenance cycles.

This chapter also points out the potential to combine our models with an optimisation technique for planning maintenance. Multi-criteria and multi-objective decision making have been intensively studied in maintenance problems with satisfying performance (see De Almeida et al. [32] for a review). As a future step, with constraints like cost, asset state and reliability, and objectives like a certain level of reliability or minimum cost, we would like to find the optimal selection of repair over some time.

Chapter 7

Managing Large and Complex Maintenance Modelling

Chapter 4 has shown a set of generic model options available for maintenance modelling:

- Learning deterioration with a statistical distribution from uncertain data and elicited knowledge (Section 4.1).
- Learning deterioration from other assets (Section 4.2).
- Condition prediction of an asset with multiple states (Section 4.3.1).
- Condition prediction of an asset assembled by multiple components (Section 4.3.2).
- Repair decisions reasoning (Section 4.4) and maintenance planning (Section 4.5).

We can combine these models in various ways for different problems as shown in Chapter 6 using real-world cases. However, the details of these models often vary when different maintenance problems have different assumptions or specifications. Also, it becomes more complicated when assets are rated by multiple states and assembled by multiple components, the scope and size of the resulting model increase exponentially. To tailor a maintenance model for a specific problem, especially when the required model is large-scaled and complicated, it not only demands the users to understand the complex relationships but also needs the users to have substantial knowledge about the underlying models and about how to assemble them. These requirements limit the applicability of these models, especially for the actual users - the maintenance engineers. Therefore, this chapter aims to tackle this challenge (Objective VI): to show how the modelling choices could be effectively managed for maintenance modelling with various specifications.

Applicability to Other Infrastructure Assets

Our model of bridge maintenance includes many characteristics that also apply to the modelling of the maintenance of other critical infrastructure assets. In particular: i) the asset condition is rated by a set of ordinal states based on the levels of safety and reliability, and the asset's deterioration can be described by the transition from one state to another, and ii) the asset can be considered as a system assembled from multiple components with various system configurations. These characteristics are common in static infrastructure asset, for example,

- **Pavement**: i) Pavement Condition Index (PCI) is an example measurement that is used to evaluate the health of pavement. It ranged from 0 to 100, where 0 represents the worst condition and 100 represents the perfect state. In practice, these states are often categorised into a smaller number depending on the repair requirement, for example, in Cheng and Remenyte-Prescott [25], PCI is divided into 5 categories. In this example, the deterioration of pavement is also modelled in the form of Markov process to represent the transition between these states, and each transition is assumed to follow a Weibull distribution or an exponential distribution as in Lethanh and Adey [97]; ii) An example pavement section was presented in Dilip et al. [35] with three components including asphalt layer, granular layer and subgrade. The system failure of this pavement section was modelled with a series configuration of these components.

- **Railway track**: i) Track condition can be measured by its vertical alignment to the rails. In Audley [9], track quality was classified from very good to poor based on the standard deviation of the track alignment, and each transition between two states follows a Weibull distribution; ii) Together with components such as points operating equipment, lineside cabinet, ballast and sleepers, railway tracks can form a track switches system. Bemment et al. [14] studied the reliability of track switches system based on various arrangements of components with different system configurations.

The differences between bridge maintenance modelling and other infrastructure assets' modelling mainly lie in some of the assumptions and requirements. One example is given by the assumed transition distribution. Some assume this follows a Weibull distribution with two parameters while others use an exponential distribution with one parameter. However, we have not further explored the application of our framework to other classes of assets in this thesis.

Outline

Section 7.1 introduces the background of the model-based approach, which is later used to resolve this problem. Section 7.2 develops a model-based maintenance framework that extends the generic models from Chapter 4 with model assumptions and a relational database. Section 7.3 presents a prototype to show how we can use this framework: we manage to build a variety of model variants using a number of relational tables. Section 7.4 gives a summary of this chapter.

7.1 Introduction to Model-Based Approach

The emerging field of Model-Based System Engineering (MBSE) gives a framework to define models and systematically organise them. It proposes to develop a range of generic models to define and design a system under development with an agile modelling formalism to meet different modelling requirements without changing the entire model [143]. These domain models are used as the primary way to explore and exchange information between engineers before system implementation, whilst reducing the dependence on traditional document-based communication. By doing so, MBSE facilitates the feedback on problem specification and modelling decisions from engineers and gives an efficient way to guide system development [42].

The model-based formalism is implemented using a unified language (e.g. SysML language) to provide a platform for integrating different modelling requirements to perform various system analysis. This formalism has been extended to the safety and reliability domain, so-called Model-Based Safety Assessment (MBSA) (see Lisagor et al. [98]). MBSA aims to unify classical safety and reliability modelling (e.g. fault tree and Petri net as discussed in Section 2.4.4) and to generate an integrated structure for a range of safety and reliability analysis (e.g. fault tree analysis and system diagnosis). This concept has been applied in maintenance assessment in recent years. For example, the AltaRica modelling language [7], separates system specification from maintenance-related analysis with a range of reusable modelling techniques. In project AltaRica 3.0 [143], the implementation of the stochastic process, Markov chain and Petri net enable this language to model and compile maintenance analysis for a complex system.

However, to our knowledge, the current practice of MBSA does not yet encompass learning deterioration rates from inspection data and knowledge, or learning from related assets, and lack evidential reasoning. Of which, the BNs for maintenance modelling proposed in the previous chapters are capable of serving these purposes. They not only provide us with the possibility to model different aspects of maintenance problems but also enable us to include alternative data and unused expert knowledge. However, these models cannot yet be presented to a maintenance expert easily since adapting the underlying generic models to a particular problem context requires an engineer to comprehend the implementation and inference of BNs. Fortunately, advances in model-based machine learning languages provide us with a promising perspective to tackle this problem.

In model-based machine learning (see Bishop [17] and Ghahramani [55]), models and problem specifications are defined in a compact language as the primary mean of communication between users, while inference or machine learning algorithm codes are generated and run automatically. Among them, probabilistic programming languages such as Figaro [139] targets at modelling and inferencing problems probabilistically is suited in implementing BNs. Once the generic BN models are defined by the modellers using a probabilistic programming language, engineers can manage the use of models without a deep understanding of BNs and their inference.

To help users manage the models, model-based approaches often use the object-oriented paradigm to provide a library of generalised models for reuse. This is not provided by traditional BNs where they have a fixed set of variables and relationships. This issue has been widely researched for BNs, for example, the development of OOBN models as discussed in Section 3.5. The use of OO in BNs not only can benefit the model representation, but also the inference of them. However, when structuring OOBN models, the relations between objects are described deterministically only. This limits the use of them, especially when the modelling problem has complex relationships between objects.

Probabilistic relational models, developed by Koller [81] combines a relational structure with OO designed probabilistic graphical models (e.g. OOBN models). A PRM combines probabilistic dependencies with a relational schema that describes the entities in the problem domain. This representation provides a separation between the model library and structure relationships. This representation further eases the pain for engineers when organising the models for large and complex maintenance problems.

Therefore, in order to assist models building for large and complex maintenance problems, inspired by MBSA and the development of model-based machine learning languages, this chapter develops a model-based framework for asset maintenance assessment. The framework separates reusable low-level models from modelling choices and asset descriptions. The framework is encoded with a PRM representation. The problem specification of the target domain is represented by a relational database instantiated by its relational schema mapped with the model assumptions. The ground model for a specific maintenance problem is therefore instantiated based on the query on the database. The database recalls a range of generic BNs (each represented by its probabilistic dependencies) for asset maintenance modelling from different perspectives.

7.2 Model-Based Asset Maintenance Framework

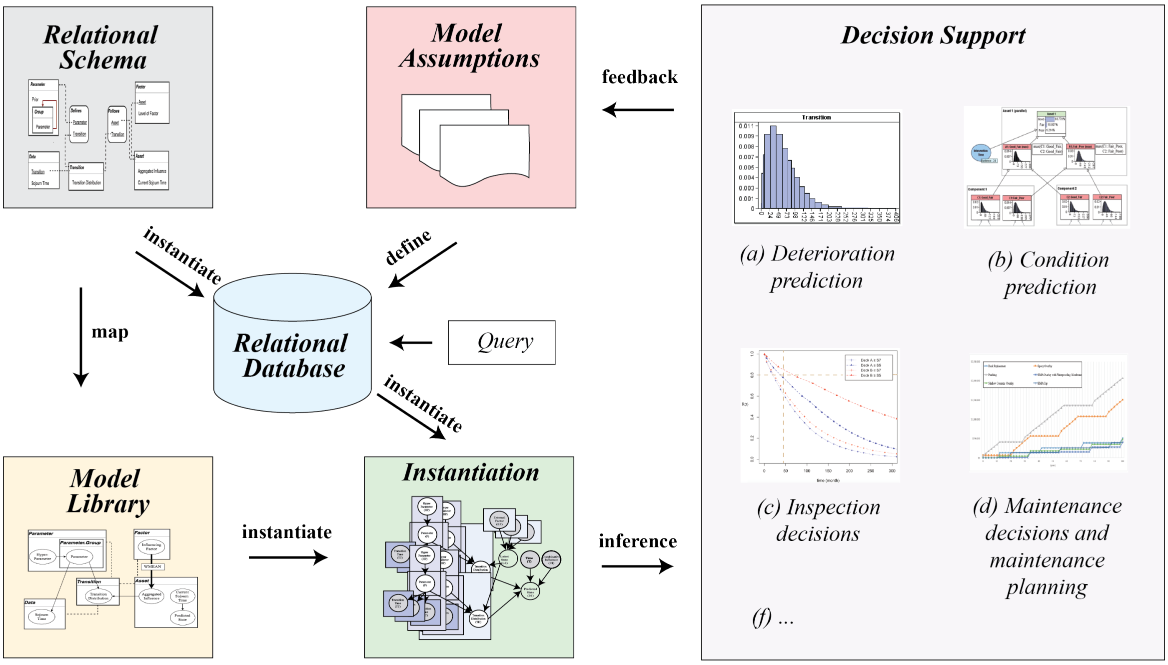


Figure 7.1 Stages of model-based asset maintenance framework.

To address the problem of many variants of maintenance-related models, we develop a framework to help domain experts express problem specification and related data and knowledge. The stages of model-based asset maintenance framework can be illustrated in Figure 7.1:

- **Relational schema** (Section 7.2.1) is used to organise the asset information and its relationship with the model assumptions. It is a blueprint of how the relational database is built.
- **Model assumptions** (Section 7.2.2) are used to define the specifications of the problem domain.
- **Relational database** (Section 7.2.3) is the instantiation of the relational schema. It includes the configuration between classes mapped from model assumptions and input from historical data and expert knowledge.
- **Model library** (Section 7.2.4) encodes the possible dependencies of probabilistic models in the problem domain.
- **Instantiation** (Section 7.2.5) is performed after a query is entered, its inference is performed by the engines supported by the model-based language.

- **Decision support** (Section 7.2.6) are made based on the queries. They can also be evaluated to provide feedback about the selected models and knowledge in order to refine the selected models.

The following subsections describe each aspect of the framework.

7.2.1 Relational Schema

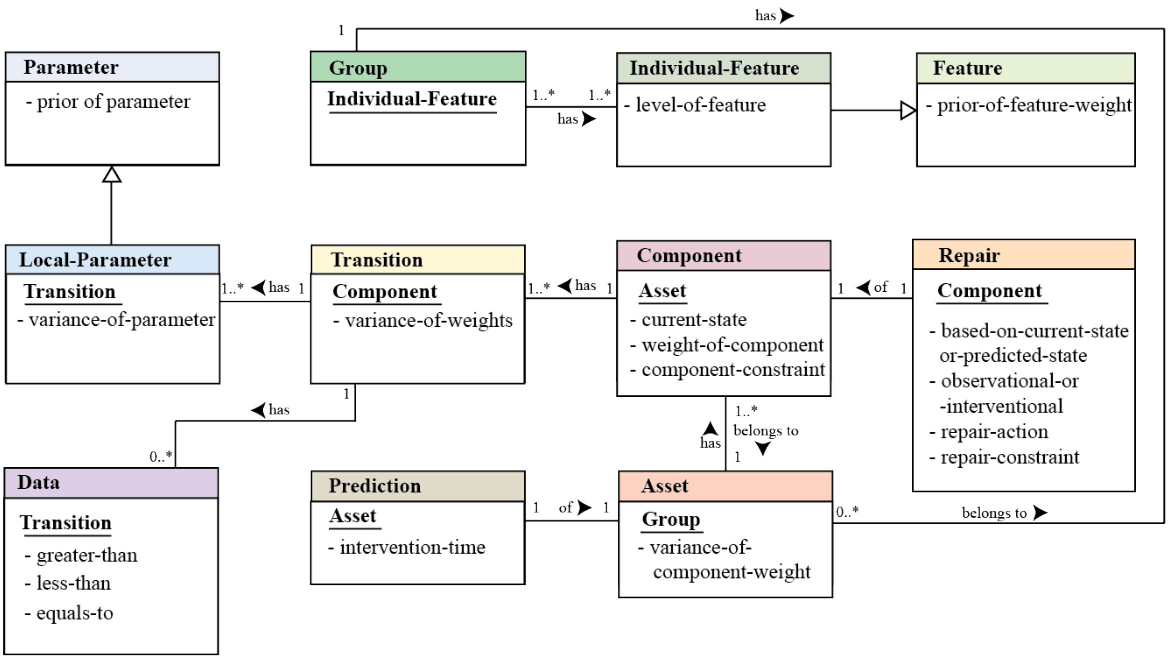


Figure 7.2 Class diagram of the PRM's relational schema.

Inspired by the generic models proposed in Chapter 4 about maintenance problems from different perspectives, a relational schema is derived and represented in the form of a class diagram shown in Figure 7.2. As one of the components of a PRM, a relational schema allows us to exhibit domain relationships between objects. A relational model consists of classes $X = \{X_1, X_2, ..., X_n\}$, where each class has a number of attributes $A(X)$ and reference slots $R(X)$ to refer an object to another object. $X.A$ is used to denote the attribute A of class X and $X.R$ is used to denote the reference slot R of class X. In this figure, the reference slots are bolded and underlined. For example, class Transition has an attribute variance-of-weights: **Transition.variance-of-weights** and a reference slot Component: **Transition.Component**.

In Figure 7.2, between classes, there are notations representing the information flow and types of relations between them:

- **One-to-one relation (1 : 1)**: for example, class Prediction to class Asset has a one-to-one relation. It represents one prediction object corresponds to one asset object only. Its inverse relation remains the same as one-to-one relation. But in this example, there is only one information flow: the prediction of an asset. This information flow is used to navigate the instantiation of an asset when a query about a specific prediction is made.

- **One-to-many relation (1 : 1..*)**: for example, class Asset to class Component has a one-to-many relation. It represents there are multiple component objects ($n \geq 1$) for one asset object. Its information flow can be used in cases like a prediction of an asset's condition that is evaluated by its components. Its inverse relation is a one-to-one relation meaning each component object belongs to one asset object only. Its information flow can be used to determine which asset the component belongs to, and further to indirectly associate which group the component belongs to.

- **One-to-zero-or-more relation (1 : 0..*)**: for example, class Asset to class Group, it represents there are zero or more asset objects for one group object. When there is no asset for that group, class Asset will not be instantiated; this can be further used to model the existence uncertainty when learning the model structure (this is out of the scope of this thesis but can refer to Koller [81] for further study). Its information flow tells which group the asset belongs to when learning from other groups. Its inverse relation is also a one-to-one relation meaning one asset object belongs to one group object only. Its information flow is a critical path in deciding what assets are used to learn deterioration.

- **Many-to-many relation (1..* : 1..*)**: for example, class Group to class Individual-Feature, it represents for each group object, it has multiple feature objects where each feature object has an individual feature level object. Its inverse relation for class Individual-Feature to class Group is also a many-to-many relation. It represents that for each individual feature level object, it may belong to multiple group objects.

- **Inheritance relation**: for example, class Local-Parameter is a subclass of class Parameter. The subclass inherits the attributes from the superclass. Therefore, class Parameter preserves the attribute prior-of-parameter from its superclass. This allows us to specialise functions of the subclasses when modelling with hierarchy.

One of the practical uses of reference slots is to connect attributes from other classes that are indirectly related using a slot chain. A sequence of reference slots forms a slot chain, it allows us to define functions that compose of attributes from other objects even when they are indirectly connected. For example, to define the function of deterioration time in class Data, we can assess its parameter(s) by a slot chain: **Data.Transition.Local-Parameter** (see Figure 7.2). Since class Data to class Transition is an inverse relationship of one-to-zero-or-more relation, hence, it is a one-to-one relation. And class Transition to class Local-Parameter is a one-to-many relation. Therefore, the slot chain has a one-to-many relation. This enables us to model the function of deterioration time with many parameters ranging from 1 to n (e.g. a one-parameter exponential distribution or a two-parameter Weibull distribution).

The relational schema is a structure skeleton of the relational database. Before input the asset information, we need to establish the relationship between classes of a problem domain, where the stage of model assumptions plays this important role.

7.2.2 Model Assumptions

The model assumptions are mapped to the relational database to configure the relationships between classes and later used to define the expression of the variable in the model library. The model assumptions contribute to the modelling specifications, which are fixed for a specific problem but varies for different problems. They can be defined by industry standards, guidelines, the nature of the system itself, the practical experience made by domain experts, feedback from the previous models or aided by some machine learning techniques. This subsection describes the key aspects of the assumptions considered here, but detail functions can refer to Chapter 4:

- **The number of deterioration states**: Each asset is usually rated with a state representing its level of functionality. For example, a 3-point grading system shown in Section 6.1.1 for GB bridge and 10-point grading system shown in Section 6.1.2 for US bridges are used. Grading systems are normally adopted from industry standards and are often used to identify the associated risk of the asset. They vary from infrastructure type, countries, and sometimes, inspection agencies. An n states grading system results in $n-1$ transitions represented in the instantiation of class Transition. For inference benefits, they are aggregated using the binary factorisation when evaluating the state of a component as discussed in Section 4.3.1.

- **The choice of transition distribution**: Different distributions can be used to estimate transition times of deterioration, based on their goodness of fit. The goodness of fit of the distribution is usually done by visual observation and hypothesis tests [115]. A range of study has been developed to find the best fit distribution of asset state deterioration times. For example, the exponential distribution has been used for railway track [63, 67] and the Weibull distribution for bridges [3, 94] (see Section 2.3.1). The choice of distribution defines the function of the variable transition distribution and variable deterioration time. The number of parameters in the distribution's pdf fixes the number of instances of the Parameter class for each Transition (e.g. exponential distribution with one parameter and Weibull distribution with two parameters).

- **The classification of assets**: Under the assumption that similar assets may deteriorate similarly, we could use their features to quantify assets into different groups - where each group is assumed to have the same deterioration rates as shown in Chapter 5. By doing so, historical data can be pooled and used to learn assets deterioration parameters between groups.
  - Grouping by features: The levels of asset's features define the asset group. However, first, we need to consider how many features, what features to be used to decide the grouping and their weights. This information configures the structure of class Group and class Feature. It can be done by experts knowledge, where they believe there are dominant features that are significantly affecting the deterioration of assets. For example, masonry bridge is considered to decay significantly slower than other bridge types from personal communication with a bridge engineer. By having this information, we can consider grouping bridges by their materials. Experts can assign the weights of features, but if this information is unavailable, uninformative priors can be assigned. The grouping can be refined by feedback from the previous models.
  - Individual feature levels: We can use a scalar like low, medium and high influence on deterioration rate as shown in Section 5.3.2 to quantify the level of features. Either experts or clustering techniques can do the assignment of feature level. For example, for feature material, experts can assign bridges built with masonry has a low influence on accelerating its deterioration rate, while bridges built with metal has a high influence. Clustering technique can be used to divide assets with different feature values into different clusters using the training data. Each cluster can be modelled with a feature level. Multiple features, where each feature has its feature level and weight, are later aggregated using a weighted mean function with a variance about the confidence level about their weights.
  - Prior knowledge about deterioration characteristics: Once the groups are defined, we can ask experts to assign the priors for the group parameters (see Section 4.1.3 about how to elicit priors of parameters in a Weibull distribution). This information is mapped to class Parameter accompanied by variances in class Local

Parameter about the differences between group parameters and local parameters. In the case where experts choose not to group assets, the prior of the parameter becomes the prior of the asset's parameter itself.

- **The asset's components and their configurations**: Asset could be considered as a single component system or a multi-component system depending on its nature. These components can be assembled with various configurations such as in parallel, series or bridge structure as discussed in Section 2.3.3. For traditional parallel or series configuration, we can define the aggregation from class Component to class Asset as a maximum or minimum function as an FT gate. For bridge structure, we can define a weighted mean function where each component is given a weight by experts representing its importance in deciding the condition of the asset. Similarly, this can also be extended to describe a parallel or series system with weights using functions like weighted maximum and weighted minimum.

- **The available repair actions for components and their effectiveness**: Different components may have different repair actions, and different actions may have different effectiveness in restoring their states. Examples can be found in Section 6.3. The configuration of its CPT can be learned from historical records, such as the frequency of restoring the component to a particular state using a specific repair action or given by experts.

7.2.3 Relational Database

Koller [81] introduced a way to map the class diagram of the PRM and instantiated it as a relational database to store asset information. Each class corresponds to a table in the database. Among the table, each column corresponds to an attribute from the class diagram, and the reference slots become the foreign keys to link another table. An example of how the foreign keys are connected between classes is later given in Section 7.3.1. The model assumptions from the last subsection configure the number of relations between the classes. For example, for a specific transition T, a two-parameter Weibull distribution is chosen. Hence, in the relational table of class Local-Parameter, among the many rows, two of them (two local parameters) have the same foreign key Transition T. More examples are illustrated in Section 7.3 in order to represent different model variants.

7.2.4 Model Library

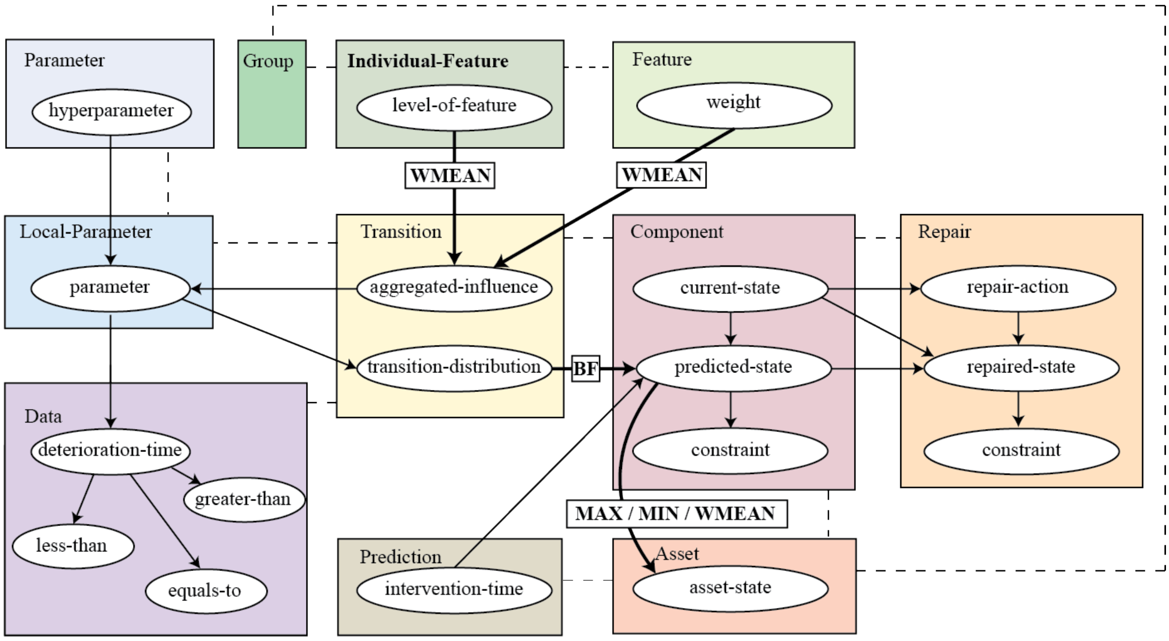


Figure 7.3 Probabilistic dependencies of the PRM.

The generic models from Chapter 4 are mapped to the relational schema, and stored in the model library represented as the probabilistic dependencies of a PRM showed in Figure 7.3. Notes that though class Group does not have any variable, it is still essential to build it as an individual class so that we can link related objects in the relational database. Also note that the model library is extendable, in the future we can include more generic models to satisfy more objectives and assumptions.

In this figure, a square (called a class) defines a group of identical objects that share the same set of variables or probabilistic models. An oval within a square defines a variable, and the directed edge defines the dependency of variables. Aggregation is defined by a bold arrow accompanied by a text about the type of aggregation. The dashed line represents the relational connections between classes, which is identical to the connections in the relational database.

To recompile the maintenance-related BNs discussed in the previous sections, dependency classes involved in the model library (some may require more classes depending on the requirements) can be summarised as the following table:

Type	Classes
Learning deterioration with a statistical distribution from uncertain data and elicited knowledge (Section 4.1)	Data, Local Parameter, Transition
Learning deterioration from other assets (Section 4.2)	Data, Parameter, Local Parameter, Transition, Individual Feature, Feature
Condition prediction of an asset with multiple states (Section 4.3.1)	Data, Local Parameter, Transition, Prediction, Component
Condition prediction of asset assembled by multiple components (Section 4.3.2)	Data, Local Parameter, Transition, Prediction, Component, Asset
Repair decisions reasoning (Section 4.4) and maintenance planning (Section 4.5)	Prediction, Component, Repair

The model probabilistic dependencies are abstract as it is the foundation (the variables and their dependencies involved in the BNs) to represent a number of different structures. However, the variable expressions and their observations are not defined, and class relationships are not configured. These issues are resolved when communicated together with a relational database to build a specific model based on the query.

7.2.5 Instantiation

Instantiation of a PRM allows us to build a specific ground BN from the probabilistic dependencies in the model library and the asset information in the relational database. The resulting ground BN is built on a set of variables $X.A$ (attribute A from class X), each variable depends probabilistically on its parents (defined by the probabilistic dependencies in Figure 7.3) - either directly using a reference slot or indirectly using a slot chain. Their functions or CPTs are defined from the model assumptions. Later Section 7.3 shows how the instantiation is done, it is illustrated by a number of BN examples with various common structures.

After the ground BN is built, many algorithms are available to provide inference based on the query when modelling using an MBML language like probabilistic programming language Figaro. A detail introduction about the available inference algorithms in Figaro can be found in Pfeffer [139].

7.2.6 Decision Support

A range of inspection and maintenance decisions support as shown in Figure 7.1 can be made using the model-based asset maintenance framework. The decisions are made based on the query embedded in the relational database. For example, when provided with a sequence of intervention time in table Prediction for assetID:X, we can reason the condition prediction of assetID:X or its components over time. When provided with a constraint about the anticipated component state, we can analyse its reliability to find suitable intervention time meeting criteria set by the decision makers. Alternatively, we can also set a constraint about the repaired state to suggest maintenance action. The variety of decisions we can make are the same as introduced in Chapter 6. However, note that we can make more decisions if more generic models are implemented in the model library.

The decisions made can be used to provide feedback for the model assumption and relational database in the early implementation stage. We can evaluate the model performance by comparing different variants, with metrics such as predictive accuracy or computational speed. Further refinement of data sources (e.g. from other source domain), expert knowledge (e.g. different experts or different types of expertise), and variations of assumptions (e.g. different groupings of assets or distributions) are possible. This process repeats until a level of acceptable performance is accepted, or it exhausts all the resource. As a future study, with the success of automatic inference software [17], the refinement process can be made automatically with a defined threshold in an MBML framework.

7.3 Prototype System: Asset Maintenance Model Variants

Customised instantiation of this framework for practical uses has been developed in Chapter 5 emphases on deterioration prediction and Chapter 6 emphases on decisions support. The focus of this section is to show how the model-based asset maintenance framework encodes different model assumptions in the relational database and how models are instantiated based on the query from decision makers that may happen in practice.

A series of model variants is presented, including deterioration prediction, condition predictions with various configuration, repair decisions and maintenance planning. We wish these variants are sufficient to convey the idea of providing a straightforward use of the prototype system for engineers: rather than letting engineers building BNs directly, instead, this allows them to manipulate the model assumptions and a relational database to build models of interests.

7.3.1 Variant 1: Deterioration Prediction

Before mapping the model assumptions and storing asset information to the relational database, we need to map the classes in a class diagram to a set of relational tables. Within each class, the class name becomes the primary key of the table used to identify the entity of the object. The attributes are mapped to a number of columns in the table, so as the foreign keys. An example is shown in Figure 7.4. For example, class Data is mapped to table Data, where DataID is the primary key when referring to a data object. It has a foreign key TransitionID indicating which transition the data object belongs to. Three attributes are mapped to three columns of the table respectively to model the type of each data.

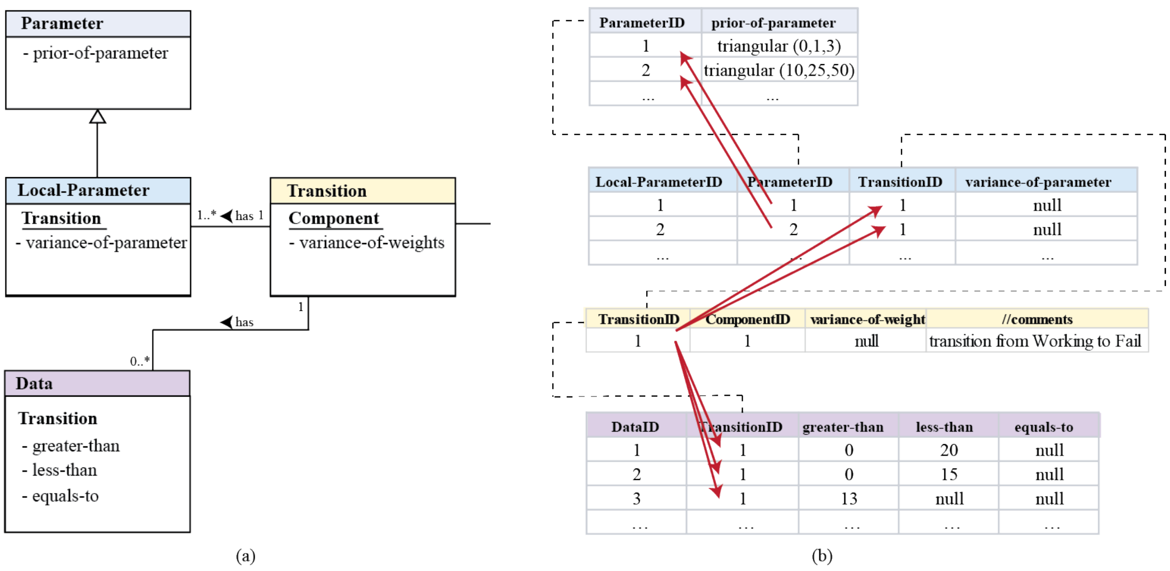


Figure 7.4 Mapping a class diagram to relational tables for variant 1.

After the skeleton of the tables is defined, data and knowledge can be entered before the query and instantiation. This example focuses on a simple parametric distribution learning for deterioration prediction, the model assumptions are:

- TransitionID:1 is a Weibull distributed transition.
- TransitionID:1 does not need to learn from other groups.

Figure 7.4 (b) presents the inputs. Assume the engineers want to query the transition distribution of TransitionID:1, its model is instantiated based on the relational database:

- **Table Transition**: Since we do not want to learn the distribution from others, the variance of feature weights is set as null in TransitionID:1 to indicate variable TransitionID:1.aggregated-influence is not implemented (showed as a dotted variable).

- **Table Local-Parameter**: The model assumes a Weibull distribution for TransitionID:1, therefore, there is a one-to-two relation between object TransitionID:1 and its corresponding local parameters: Local-ParameterID:1 and 2. They are both set as null in variance-of-parameter to indicate each parameter is directly impacted by its own (not shared with other groups) - hence, variable Local-ParameterID:1.parameter and Local-ParameterID:2.parameter are not implemented. However, at the same time, class Local-Parameter has an inheritance relation with class Parameter - each local parameter can share the attribute(s) from its superclass. An additional foreign key of the ParameterID is added to assess each object's superclass table.

- **Table Parameter**: Each ParameterID has a prior expression for its parameter, they are instantiated as the hyperparameter. Since variable parameter is not instantiated but variable hyperparameter is, it takes over the dependency between class Local-Parameter and other classes. For example, ParameterID:1.hyperparameter with a prior of triangular distribution (0,1,3) becomes one of the parents of TransitionID:1.transition-distribution, DataID:1.deterioration-time, DataID:2.deterioration-time and DataID:3.deterioration-time.

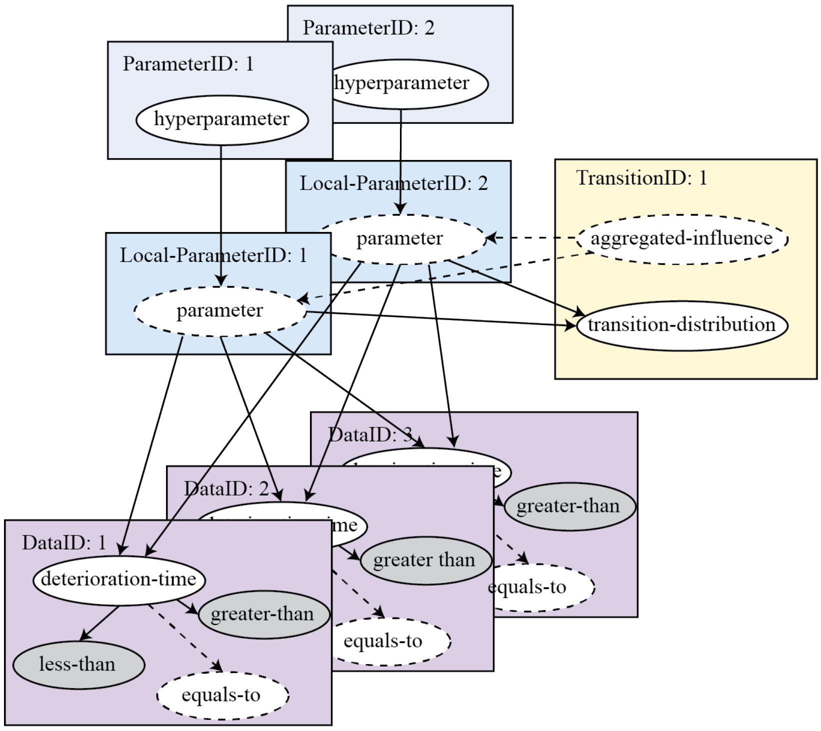


Figure 7.5 Instantiation of Variant 1.

- **Table Data**: Three data objects have the same foreign key TransitionID:1 in this table. For example, in object DataID:1, it has attributes DataID:1.greater-than with a value of 0 and DataID:1.less-than with a value of 20 and null for DataID:1.equals-to. In the instantiated model, only variable greater-than and variable less-than are instantiated, they are translated into Boolean expressions if (deterioration time $>$ 0, "True", "False") and if (deterioration time $<$ 20, "True", "False") and both are observed as "True". This represents an interval-censored data. Apart from this data, DataID:2 is an interval-censored data and DataID:3 is a right-censored data.

Figure 7.5 presents the instantiated BN. The posterior distribution in TransitionID:1.transition-distribution is used to answer the query from the engineers. This subsection shows how the probabilistic dependency of a PRM from Figure 7.3 is instantiated guided by its relational database instantiated from the class diagram in Figure 7.4 and the model assumptions.

7.3.2 Variant 2: Condition Prediction - Deterioration Rate Learned from Others

This model variant focuses on the condition prediction of an asset where its component deterioration distribution is learned from others. The model assumptions are:

- AssetID:1 has only one component ComponentID:1.
- ComponentID:1 is rated with a binary-state scale. Therefore, it has only one transition TransitionID:2 represents its transition from Working to Fail.
- TransitionID:2 and 12 follows a Weibull distribution respectively.
- FeatureID:1 and 2 are used to separate assets into groups, each feature is quantified into three levels representing their influence on deterioration rate from low to high.
- AssetID:1 belongs to GroupID:1 and AssetID:3 belongs to GroupID:2.
- Here assumes the scale parameter Local-ParameterID:4 in TransitionID:2 (from GroupID:1) can learn from the scale parameter Local-ParameterID:6 in TransitionID:12 (from GroupID:2).

Figure 7.6 presents the inputs of the related database. In this example, the engineers want to predict the condition of AssetID:1 after 24 months, the model instantiation follows the relational database:

- **Table Prediction**: To predict AssetID:1's condition after 24 months, variable PredictionID:1.intervention-time is instantiated with an observation of 24, it becomes one of the parents of AssetID:1's component(s).

- **Table Asset**: This table is an intermediate relational table to access table Group and table Component. It indicates that AssetID:1 belongs to GroupID:1, so its component

PredictionID	AssetID	intervention-time
1	1	24

AssetID	GroupID	variance-of-component-weight
1	1	0
…	…	…
3	2	0

GroupID	1st-Individual-FeatureID	2nd-Individual-FeatureID
1	1	4
2	2	4

Individual-FeatureID	FeatureID	leve-of-feature
1	1	low
2	1	medium
…	…	…
4	2	low

FeatureID	prior-of-feature-weight
1	uniform(0, 1)
2	uniform(0, 1)

ComponentID	AssetID	current-state	weight-of-component	component-constraint
1	1	Working	1	null
…	…	…	…	…
11	3	Working	1	null

TransitionID	ComponentID	variance-of-weight	//comments
…	…	…	
2	1	0.05	transition from Working to Fail
..	…	…	
12	11	0.1	transition from Working to Fail

ParameterID	prior-of-parameter
…	…
3	triangular (1,2,4)
4	normal (25, 10)
5	triangular (0,1,1.3)

DataID	TransitionID	greater-than	less-than	equals-to
…	…	…	…	…
5	2	12	25	null
6	12	32	48	null
7	12	13	null	null
8	12	null	null	23

Local-ParameterID	ParameterID	TransitionID	variance-of-parameter
…	…	…	…
3	3	2	null
4	4	2	15
5	5	12	null
6	4	12	5

Figure 7.6 Relational tables for Variant 2.

ComponentID:1 also belongs to GroupID:1; similarly, AssetID:3 belongs to GroupID:2, so its component ComponentID:11 also belongs to GroupID:2. Since both assets only have one component, the variance of component weight is both sets as null.

- **Table Group**: This table is used to encode the many-to-many relation to table Individual-Feature. From the model assumptions, two features are used to classify assets into groups, each of them is mapped into a column of this table. Each column becomes a foreign key linked to table Individual-Feature. A different combination of the individual feature levels becomes an identical group.

- **Table Individual-Feature**: Each feature is quantified into three levels. Additionally, each feature level inherits the prior for its corresponds feature's weight from its superclass Feature.

- **Table Feature**: Each FeatureID has a prior expression for its weight, they are instantiated as variable weight. In this example, the experts are not confident in assigning their weights. Therefore, uninformative priors uniform distributions with lower bounds of 0 and upper bounds of 1 are assigned for FeatureID:1 and 2. The weights are together marginalised as summed to 1 when inference is performed.

- **Table Component**: Since AssetID:1 only has one component. Hence there is only one row in this table with a foreign key of AssetID:1, and that component along has a weight of 1 to form this asset. The entry of current-state is 'Working', therefore, there is one transition for this component: from Working to Fail. Because the query of this example is to predict the condition rather than reliability analysis, the constraint is not instantiated, therefore, it is set as null.

- **Table Transition**: The variance in this table is used to represent the credibility of the features' weights. For example, TransitionID:2.aggregated-influence is instantiated, it is modelled as a Ranked node with a function of TNormal distribution $(\mu, \sigma^2, 0, 1)$, its mean μ is an aggregated function from its corresponding Individual-FeatureID:1 and 4 that are weighted by the weights of its features: FeatureID:1 and 2, and its σ^2 is the variance of the weights.

- **Table Local-Parameter**: Transitions in this example are both assumed to be Weibull distributed, therefore each identical TransitionID has two rows in this table. In the column of variance-of-parameter, Local-ParameterID:3 and 5 are both set as null to represent these two parameters are directly impacted by their own group - they inherit attributes from their own superclasses. However, Local-ParameterID:4 and 6 have the same ParameterID:4 with different variances - they inherit the same attribute from the same superclass ParameterID:4.

- **Table Parameter**: Variable ParameterID:3.hyperparameter and ParameterID:5.hyperparameter take over the corresponding dependencies of variables between class Local-ParameterID:3 and 5 and other classes. ParameterID:4.hyperparameter becomes the parent of Local-ParameterID:4.parameter and Local-ParameterID:6.parameter.

- **Table Data**: TransitionID:2 has 1 interval-censored data and TransitionID:12 has 1 interval-censored, 1 right-censored and 1 complete data.

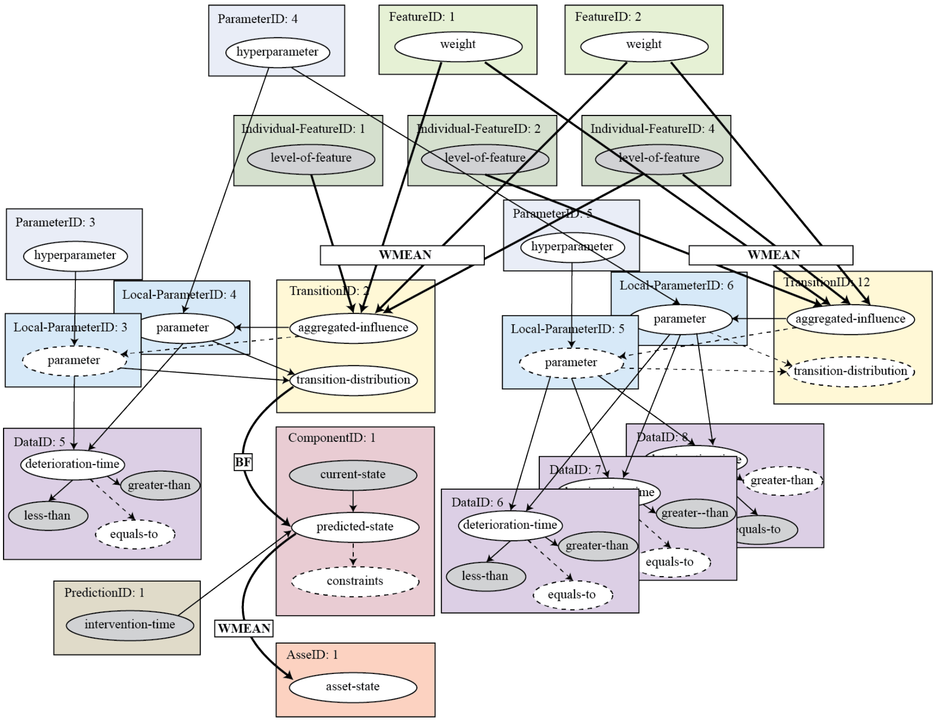


Figure 7.7 Instantiation of Variant 2.

The instantiated BN is shown in Figure 7.7. The posterior probabilities of variable AssetID:1.asset-state can be used to answer the query about its condition prediction after 24 months.

7.3.3 Variant 3: Condition Prediction - Multiple State System

This model variant focuses on the condition prediction of an asset where the condition of its component is rated by multiple states. The model assumptions are:

- AssetID:3 has only one component ComponentID:3.
- ComponentID:3 is rated with a 9-point scale. Therefore, it has 8 transitions: TransitionID:26 represents transition from S9 to S8, ..., TransitionID:31 represents transition from S4 to S3, TransitionID:32 represents transition from S3 to S2, and TransitionID:33 represents transition from S2 to S1.
- Each transition, from TransitionID:26 to 32, follows an exponential distribution respectively, and TransitionID:33 follows a Weibull distribution.
- Transitions of AssetID:3 do not need to learn it from other groups.

PredictionID	AssetID	intervention-time
…	…	…
3	3	36

AssetID	GroupID	variance-of-component-weight
…	…	…
3	null	0

ComponentID	AssetID	current-state	weight-of-component	component-constraint
…	…	…	…	…
3	3	S3	1	null

TransitionID	ComponentID	variance-of-weight	//comments
…	…	…	…
26	3	null	transition from S9 to S8
…	…	…	…
31	3	null	transition from S4 to S3
32	3	null	transition from S3 to S2
33	3	null	transition from S2 to S1

ParameterID	prior-of-parameter
…	…
22	triangular (0.01,0.03,0.08)
…	…
27	triangular (0.01,0.02,0.04)
28	normal (0.05, 0.03)
29	normal (0.5, 1)
30	normal (35, 25)

DataID	TransitionID	greater-than	less-than	equals-to
…	…	…	…	…
76	26	63	78	null
…	…	…	…	…
83	31	25	48	null
84	31	28	49	null
85	31	74	98	null
86	32	null	13	null
87	32	0	23	null
88	33	73	85	null
89	33	71	84	null

Local-ParameterID	ParameterID	TransitionID	variance-of-parameter
…	…	…	…
26	22	26	null
…	…	…	…
31	27	31	null
32	28	32	null
33	29	33	null
34	30	33	null

Figure 7.8 Relational tables for Variant 3.

Figure 7.8 presents the inputs of the related database. In this example, the engineers want to predict the condition of AssetID:3 after 36 months, the model instantiation follows the relational database:

- **Table Prediction**: To predict AssetID:3's condition after 36 months, variable PredictionID:3.intervention-time is instantiated with an observation of 36, it becomes one of the parents of AssetID:3's component(s).
- **Table Asset**: One of the model assumptions indicates transitions of AssetID:3 do not need to learn from other groups. Therefore, the GroupID is set as null to prevent further instantiation of classes related to table Group. Since this asset only has one component, the variance of component weight is set as null as well.
- **Table Component**: AssetID:3 only has one component ComponentID:3 with a weight of 1. The entry of current-state is 'S3', therefore, without repair, there are only 2 transitions available for this component: from S3 to S2 and from S2 to S1. The constraint is set as null due to the query of this example is to predict the condition only.
- **Table Transition**: ComponentID:3 is rated with a 9-point scale with 8 transitions, but since its current state is in 'S3', TransitionID:26 to 31 will not be instantiated. Also,

because we do not want to learn the distributions from others, the variance-of-weight of all transitions are set as null.

- **Table Local-Parameter**: The model assumes exponential distributions for TransitionID:26 to 32, and a Weibull distribution for TransitionID:33, therefore, in this table, there is one row dedicated for each transition from TransitionID:26 to 32 and two rows dedicated for TransitionID:33. Since TransitionID:26 to 31 are not instantiated, their corresponding Local-Parameter objects (ID from 26 to 31) are not instantiated as well. The variance-of-parameter of Local-ParameterID:32, 33, 34 are all set as null showed their variable parameters are not initiated. But at the same time, they can still inherit attributes from their superclass Parameter.

- **Table Parameter**: Variable ParameterID:28.hyperparameter becomes the parameter λ of an exponential distribution with a prior of a normal distribution (0.05, 0.03), it takes over the dependencies of variable Local-ParameterID:32.parameter. Similarly, variable ParameterID:29.hyperparameter and ParameterID:30.hyperparameter become the parameter β and η respectively of a Weibull distribution, and they take over the dependencies of variable Local-ParameterID:33.parameter and Local-ParameterID:34.parameter respectively.

- **Table Data**: TransitionID:32 has 1 left-censored data and 1 interval-censored data and TransitionID:33 has 2 interval-censored data.

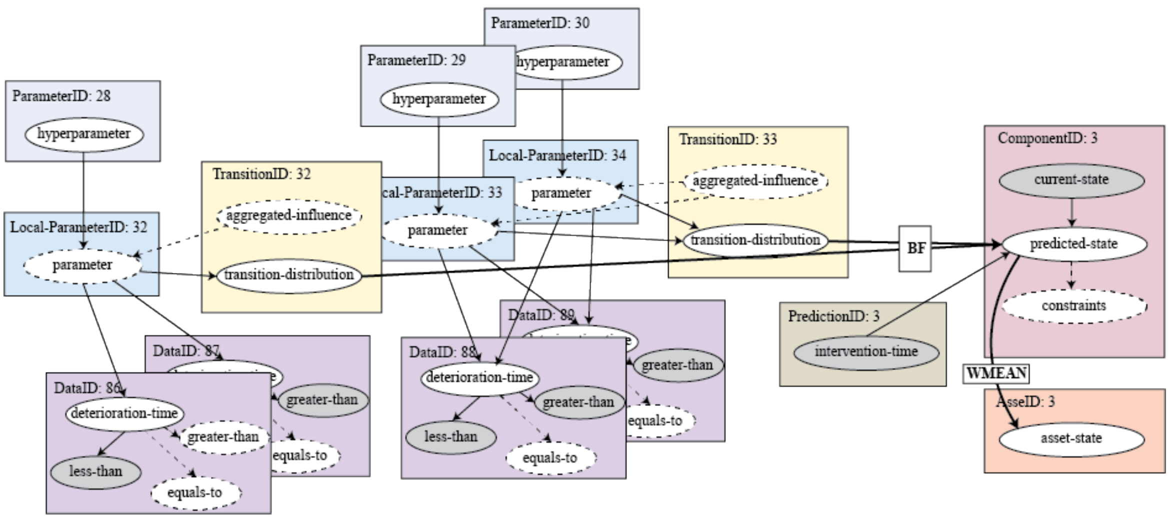


Figure 7.9 Instantiation of Variant 3.

The instantiated BN is shown in Figure 7.9. The posterior probabilities of variable AssetID:3.asset-state can be used to answer the query about its condition prediction after 36 months.

7.3.4 Variant 4: Condition Prediction - Multiple Components System

This model variant focuses on the condition prediction of an asset assembled by multiple components. The model assumptions are:

- AssetID:4 has 3 components ComponentID:15, 16, and 17.
- Each component is rated by a binary-state scale. Therefore, each has only one transition: the transition from Working to Fail.
- Each transition follows an exponential distribution and does not need to learn from other groups.

PredictionID	AssetID	intervention-time
…	…	…
4	4	36

AssetID	GroupID	variance-of-component-weight
…	…	…
4	null	0.05

ParameterID	prior-of-parameter
…	…
17	normal (0.1, 0.08)
18	normal (0.05, 0.01)
19	normal (0.08, 0.03)

ComponentID	AssetID	current-state	weight-of-component	component-constraint
…	…	…	…	…
15	4	Working	0.3	null
16	4	Working	0.5	null
17	4	Working	0.2	null

TransitionID	ComponentID	variance-of-weight	//comments
..	…	…	
51	15	null	transition from Working to Fail
52	16	null	transition from Working to Fail
53	17	null	transition from Working to Fail

DataID	TransitionID	greater-than	less-than	equals-to
…	…	…	…	…
29	51	48	74	null
30	51	24	48	null
31	52	145	168	null
32	53	121	145	null

Local-ParameterID	ParameterID	TransitionID	variance-of-parameter
…	…	…	…
21	17	51	null
22	18	52	null
23	19	53	null

Figure 7.10 Relational tables for Variant 4.

Figure 7.10 presents the inputs of the related database. In this example, the engineers want to predict the condition of AssetID:4 after 36 months, the model instantiation follows the relational database:

- **Table Prediction**: To predict AssetID:4's condition after 36 months, variable PredictionID:4.intervention-time is instantiated with an observation of 36, it becomes one of the parents of AssetID:4's component(s).
- **Table Asset**: One of the model assumptions indicates transitions of AssetID:4 do not need to learn from other groups, the GroupID is therefore set as null to prevent further instantiation of classes related to table Group. Since this asset has multiple components, the variance of component weight is set as 0.05 to encode the uncertainty about the weights later assigned in table Component.
- **Table Component**: AssetID:4 has three components ComponentID:15, 16, and 17, they are assigned with a weight of 0.3, 0.5 and 0.2 respectively about their contribution to the state of AssetID:4. These weights are used to aggregate their predicted-state variables to the asset-state variable with a variance of 0.05 given in table Asset. The entry of current-state are all set as 'Working', therefore, there is one transition for each component. The constraint is set as null due to the query of this example is to predict the condition only.
- **Table Transition**: Components in this asset are rated by a binary-state scale. Hence, each has one transition. Also, because we do not want to learn the distributions from others, the variance-of-weight of all transitions are set as null.
- **Table Local-Parameter**: The model assumes exponential distributions for all transitions in this example, hence, each transition has a row dedicated for its local parameter. The variance-of-parameter of Local-ParameterID:21, 22, 23 are set as null because we do not want to learn from other parameters. As a result, their variable parameters are not initiated. However, at the same time, they can still inherit attributes from their correspond superclass Parameter.
- **Table Parameter**: Variable hyperparameter in each ParameterID becomes the parameter λ of an exponential distribution with a prior distribution, it takes over the dependencies of variable parameter in its correspond Local-Parameter object.
- **Table Data**: TransitionID:51 has 2 interval-censored data, TransitionID:52 has 1 interval-censored data and TransitionID:53 has 1 interval-censored data.

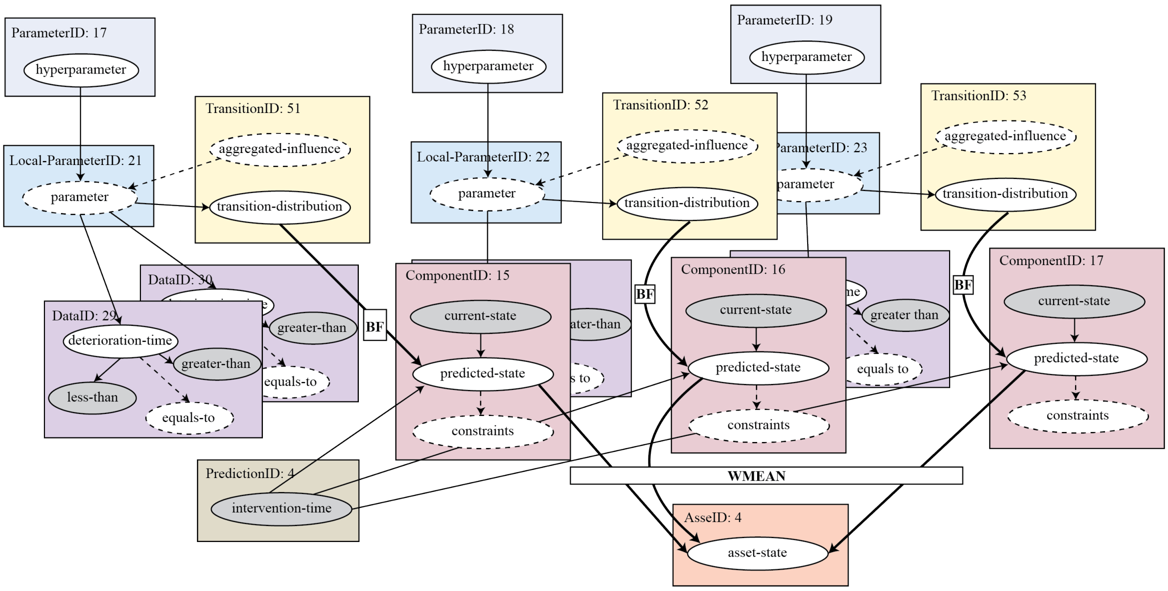


Figure 7.11 Instantiation of Variant 4.

The instantiated BN is shown in Figure 7.9. The posterior probabilities of variable AssetID:4.asset-state can be used to answer the query about its condition prediction after 36 months.

7.3.5 Variant 5: Inspection Decisions

This model variant focuses on making inspection decisions. The model assumptions are the same as in Variant 3 in Section 7.3.3.

Figure 7.12 presents the inputs of the related database. In this example, the engineers define a component is reliable by having its state greater or equal to S2. They want to find out the reliability of ComponentID:3 over time in order to decide when to inspect it. The model instantiation follows the same relational database as Variant 3 except table Prediction and table Component:

- **Table Prediction**: Since the query is to find out the reliability of ComponentID:3 over time, variable PredictionID:5.intervention-time is instantiated but not observed. This variable becomes one of the parents of AssetID:3's component(s).
- **Table Component**: AssetID:3 only has one component ComponentID:3 with a weight of 1. The entry of current-state is 'S3', therefore, without repair, there are only two transitions available for this component: from S3 to S2 and from S2 to S1. The constraint is set as $\geq S2$, it can be translated into a CPT where the constraint is 'True'

PredictionID	AssetID	intervention-time
…	…	…
5	3	NA

AssetID	GroupID	variance-of-component-weight
…	…	…
3	null	0

ParameterID	prior-of-parameter
…	…
22	triangular (0.01,0.03,0.08)
…	…
27	triangular (0.01,0.02,0.04)
28	normal (0.05, 0.03)
29	normal (0.5, 1)
30	normal (35, 25)

Local-ParameterID	ParameterID	TransitionID	variance-of-parameter
…	…	…	…
26	22	26	null
…	…	…	…
31	27	31	null
32	28	32	null
33	29	33	null
34	30	33	null

ComponentID	AssetID	current-state	weight-of-component	component-constraint
…	…	…	…	…
3	3	S3	1	≥ S2

TransitionID	ComponentID	variance-of-weight	//comments
…	…	…	…
26	3	null	transition from S9 to S8
…	…	…	…
31	3	null	transition from S4 to S3
32	3	null	transition from S3 to S2
33	3	null	transition from S2 to S1

DataID	TransitionID	greater-than	less-than	equals-to
…	…	…	…	…
76	26	63	78	null
…	…	…	…	…
83	31	25	48	null
84	31	28	49	null
85	31	74	98	null
86	32	null	13	null
87	32	0	23	null
88	33	73	85	null
89	33	71	84	null

Figure 7.12 Relational tables for Variant 5.

when the predicted-state is in S9, …, S3 or S2, otherwise, it is set as 'False'. The constraint variable is observed as 'True' in order to reason the variable of intervention-time.

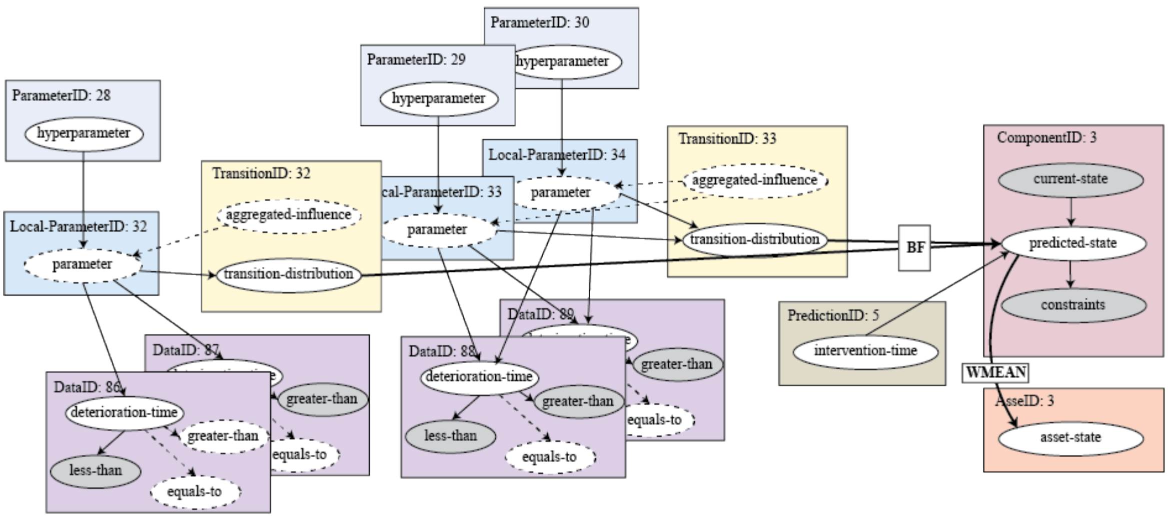


Figure 7.13 Instantiation of Variant 5.

The instantiated BN is shown in Figure 7.13. The posterior pdf or cdf of variable PredictionID:5.intervention-time is used to answer the query about the reliability of ComponentID:3 over time, by setting a threshold about the reliability, engineers can make decisions about when to inspect this asset.

7.3.6 Variant 6 and 7: Repair Decisions

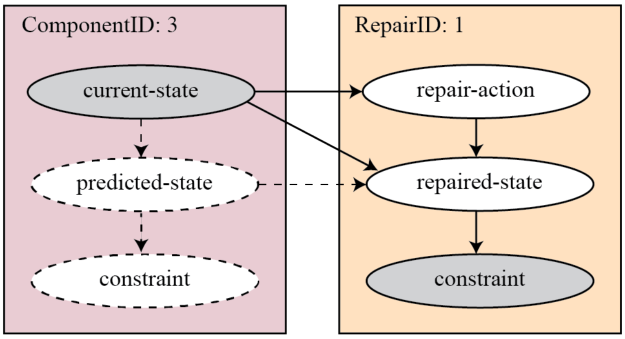
This subsection presents two model variants: Variant 6 is an observational model, and Variant 7 is an intervention model, both developed from Section 4.4. They both assume having knowledge of the available repair actions and their effectiveness on restoring the state of components.

Figure 7.14 presents the inputs of the related database. For both variants, engineers define a constraint about the repaired state of ComponentID:3 should be better or equal to S6. For Variant 6, the engineers want to know given the current state of ComponentID:3, in history, what repair actions are more likely to be taken in order to satisfy the constraint. For Variant 7, the engineers want to know if a major repair is performed, what is the condition distribution of the repaired state given its current state. The model instantiation for Variant 6 follows the relational database in Figure 7.14 (a), and Variant 7 follows the relational database in Figure 7.14 (b):

- **Table Repair**: For both variants, the repaired states are based on the current state. Hence, the edges from predicted-state to repair-state are not implemented. The constraint is set as the repaired state is $\geq S6$, it can be translated into a CPT where the constraint is 'True' when repaired state is in S9, ..., S7 or S6, otherwise, it is set as 'False'.
  - For Variant 6, the query from engineers refers to the use of an observational model. Hence, the edge from ComponentID:3.current-state to RepairID:1.repair-action is instantiated, RepairID:1.repair-action is not observed, and RepairID:1.constraint is observed as 'True'.
  - For Variant 7, the query from engineers refers to the use of an intervention model. Hence, the edge from ComponentID:3.current-state to RepairID:1.repair-action is not implemented, RepairID:2.repair-action is observed as 'major repair', and RepairID:2.constraint is not observed.
- **Table Component**: The entries of ComponentID:3's current-state for both variants are both 'S3', therefore, ComponentID:3.current-state are both observed as 'S3'. The constraint is set as null, hence, not instantiated.

RepairID	ComponentID	based-on-current-state-or-predicted-state	observational-or-interventional	repair-action	repair-constraint
1	3	current-state	observational	null	≥ S6
...	...	...	...	...	...

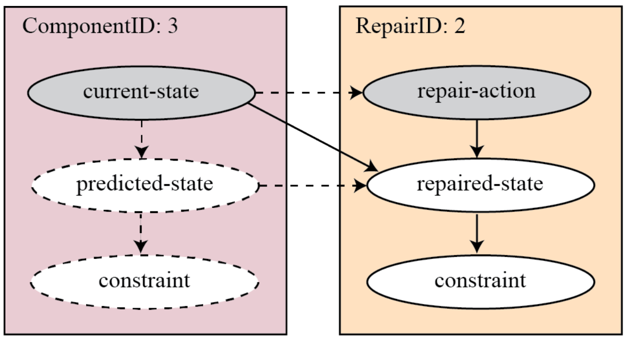
ComponentID	AssetID	current-state	weight-of-component	component-constraint
...	...	...	...	...
3	3	S3	1	null



(a)

RepairID	ComponentID	based-on-current-state-or-predicted-state	observational-or-interventional	repair-action	repair-constraint
2	3	current-state	interventional	major-repair	≥ S6
...	...	...	...	...	...

ComponentID	AssetID	current-state	weight-of-component	component-constraint
...	...	...	...	...
3	3	S3	1	null



(b)

Figure 7.14 (a) Relational tables and its instantiation for Variant 6; (b) Relational tables and its instantiation for Variant 7.

The instantiated observational BN for Variant 6 is shown in Figure 7.14 (a), the posterior probability distribution of RepairID:1.repair-action is used to suggest repair actions given the current state and constraints. The instantiated intervention BN for Variant 7 is shown in Figure 7.14 (b), the posterior probability distribution of RepairID:2.repaired-state is used to answer what is the probability distribution of the component's state if the given repair action is performed.

7.3.7 Variant 8: Maintenance Planning

This model variant focuses on maintenance planning. The model assumptions are the same as in Variant 3 in Section 7.3.3. Additionally, we assume having knowledge of the available repair actions and their effectiveness on restoring the state of components.

PredictionID	AssetID	intervention-time
…	…	…
3	3	36

RepairID	ComponentID	based-on-current-state-or-predicted-state	observational-or-interventional?	repair-action	repair-constraint
...	...	...	...	...	…
3	3	predicted-state	interventional	no-action /minor-repair	≥ S6

AssetID	GroupID	variance-of-component-weight
…	…	…
3	null	0

ComponentID	AssetID	current-state	weight-of-component	component-constraint
…	…	…	…	…
3	3	S3	1	null

ParameterID	prior-of-parameter
…	…
22	triangular (0.01,0.03,0.08)
…	…
27	triangular (0.01,0.02,0.04)
28	normal (0.05, 0.03)
29	normal (0.5, 1)
30	normal (35, 25)

TransitionID	ComponentID	variance-of-weight	//comments
…	…	…	…
26	3	null	transition from S9 to S8
…	…	…	…
31	3	null	transition from S4 to S3
32	3	null	transition from S3 to S2
33	3	null	transition from S2 to S1

Local-ParameterID	ParameterID	TransitionID	variance-of-parameter
…	…	…	…
26	22	26	null
…	…	…	…
31	27	31	null
32	28	32	null
33	29	33	null
34	30	33	null

DataID	TransitionID	greater-than	less-than	equals-to
…	…	…	…	…
76	26	63	78	null
…	…	…	…	…
83	31	25	48	null
84	31	28	49	null
85	31	74	98	null
86	32	null	13	null
87	32	0	23	null
88	33	73	85	null
89	33	71	84	null

Figure 7.15 Relational tables for Variant 8.

Figure 7.15 presents the inputs of the related database. In this example, the engineers define a component is reliable when its state is greater or equal to S6. An intervention is performed every 36 months, if the probability of componentID:3's reliability is greater than

90%, no action will be taken, otherwise, minor repair is performed. The decision makers want to know how many repair actions are taken in the next 120 years, which can be used to evaluate the life cycle cost for its maintenance planning. The model instantiation for each intervention follows the same relational database as Variant 3 excepts table Repair:

- **Table Repair**: In this example, we want to evaluate the repaired state based on the predicted state given an intervention time, hence, the edge from ComponentID:3.predicted-state to RepairID:3.repaired-state is instantiated instead of from ComponentID:3.current-state. To estimate the effectiveness of repair action, intervention model is applied - the edge from ComponentID:3.current-state to RepairID:3.repair-action is not instantiated, instead, RepairID:3.repair-action is observed. The observation of repair action can choose between no action and minor repair in this example. It is guided by the posterior of RepairID:3.constraints from the previous intervention: when its probability of being 'True' is greater than 90%, no action is observed; otherwise minor repair is observed.

The instantiated model for each intervention is shown in Figure 7.16. Multiple interventions (40 interventions for 120 years) will be performed, where the posterior of variable RepairID:3.repaired-state becomes the observation of ComponentID:3.current-state in the next intervention, and RepairID:3.constraint is used to decide the observation of RepairID:3.repair-action for the next intervention. The frequency of using minor repairs are counted to answer the query from the decision makers.

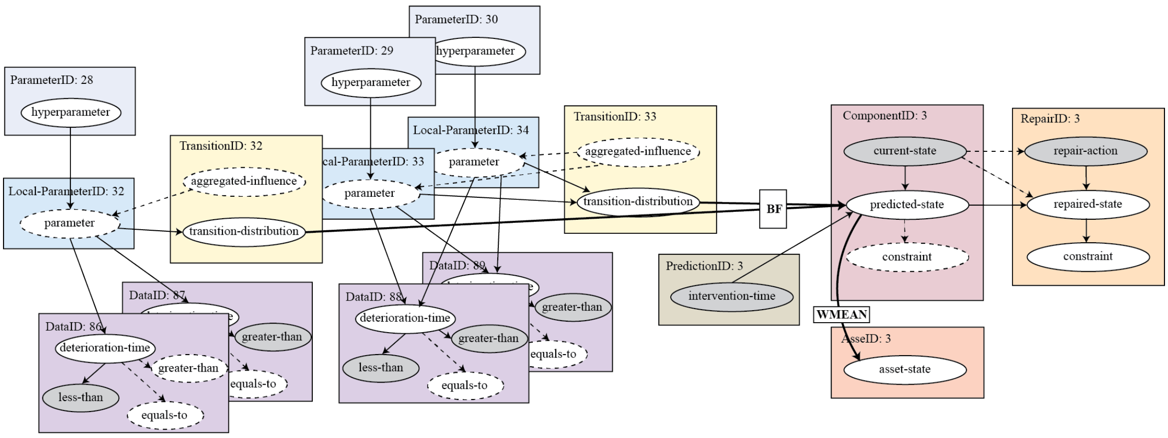


Figure 7.16 Instantiation of Variant 8.

7.4 Summary

Maintenance engineers might find it challenging to use the models from Chapter 4 without substantial knowledge of Bayesian modelling techniques. The models in Chapter 4 are generic and need, as we have shown in Chapters 5 and 6, to be adapted to the specific configuration of asset types and components. This chapter tackles this situation by providing an effective way for engineers to manage and build specific models from the generic components of Chapter 4.

We adopt concept from model-based approaches to use generic functional models as the primary mean of communication with engineers facing a maintenance problem. The generic models are represented in the form of PRM in a model-based asset maintenance framework. In the PRM, the generic models encode the probabilistic dependencies which are then assembled into a final BN using data (in a relational database) that describes the assumptions of the specific maintenance problem.

Encoding the maintenance models in a model-based framework using a PRM provides a clear separation of the stages of designing and applying the generic models for a specific maintenance problem. The role of the maintenance decision maker, who builds the model by manipulating the relational database without a detailed understanding of Bayesian modelling techniques, is separated from the role of the modeller, a specialist in Bayesian modelling techniques who encodes the probabilistic dependencies. To build a ground BN, the maintenance decision maker has only to manipulate a relational database, or even a spreadsheet with multiple tabs, containing the assumptions made about the problem and the asset information they have.

In summary, this chapter addresses Objective VI, which is to show how to manage maintenance modelling effectively. We show how this is done with a set of examples of the configurations that may be seen in practice. We finally note that within this framework, the model library is extendable, so we could include further generic models to describe a wider variety of model types and decisions if necessary.

Chapter 8

Conclusions and Further Work

In this thesis, we address some of the challenges in modelling bridge asset deterioration and supporting related decisions using BNs. Chapter 2 and 3 introduce the problem background and methodology respectively. These introductions lead to the developed models in Chapter 4, we have models that can learn deterioration from data and knowledge and models that can support the decisions making. The performance of the deterioration prediction models are validated in Chapter 5 and decision models are applied in Chapter 6 using real bridge case studies. In Chapter 7 we show how we can manage these models effectively to build complex maintenance models. Though the proposed models in this thesis are illustrated with examples of bridge assets, the models are generic and applicable to other similar maintenance problems such as railways and roads.

The following section summaries the contributions of this thesis, and it is followed by a discussion of further work.

8.1 Contributions

This section discusses the contributions of this thesis. It is organised according to the research objectives defined in Section 1.2. Among the objectives, I to IV are about deterioration prediction, V focuses on the decision support applications and VI addresses the management of complex and large-scale modelling. We show how each objective has been fulfilled:

Objective I: ***Develop a model that can learn asset deterioration behaviours from data and knowledge. Show how to encode data with uncertainty inferred from inspection history and how to derive engineering knowledge from experts that can be used in the model.***

This objective is achieved by the models in Chapter 4. Specifically:

- The foundation is given in Section 4.1.1, where we show that the parameters of a Weibull distribution can be learnt in a hierarchical BN.
- Section 4.1.2. shows how censored data – inevitably resulting from periodic inspection – are included in this model using a Boolean node with a logical expression.
- Section 4.1.3 presents a set of explanations for the characteristics of the parameters of the Weibull distribution, showing how they correspond to a maintainer's knowledge, such as trends (e.g. whether the rate of deterioration is getting faster or slower) and percentages (e.g. the time for about two thirds of the assets to deteriorate to the next state). Though this study has not worked with engineers to investigate the feasibility of eliciting this knowledge, we believe we have proposed a straightforward way to make this possible.
- Section 5.2.1 applies these models to give a practical example of how to derive deterioration data from inspection records.

Objective II: ***Make use of the feature information to study the relationship between different asset groups. Use this to leverage the learning of asset deterioration between groups, even when there is little historical inspection record for some groups.***

This objective is achieved in two stages. The first, based on the review of individual deterioration prediction in Section 2.2.2, is to cluster similar assets into groups and provide deterioration prediction on each group rather than the overall population. Since the evidence shows that deterioration rate can be affected by asset's features (e.g. a bridge with heavier loading may deteriorate faster than one with less loading), we use these features to separate assets into groups. Specifically:

- Section 4.2 extends the Bayesian parameter learning models with a further hierarchical level including an additional level of parameters - hyperparameters. This gives us a framework to include multiple groups of assets within the same model. Each group has its own local parameters, and all these local parameters from different groups are supervised by the global parameters (hyperparameters). Instead of eliciting the priors for the local parameters directly, we first elicit the priors for the hyperparameters that account for the general characteristics of all assets. Then we estimate the local parameters about their deviations from the global settings by experts. With the help of hyperparameters, we can link different groups of assets. This linkage gives an opportunity to learn between groups, where some groups may have more data that can strengthen the learning of the hyperparameters by their local parameters, which consequently, leverage the learning of local parameters of other groups with little data.

- Since experts may find it difficult to estimate the differences between groups via their local parameters directly, we use the asset group's feature values to quantify the differences in deterioration. Each feature is quantified into three levels, from low, medium to high, where a low degree represents a longer deterioration time. The variables for each feature and combined in a linear model using AgenaRisk's ranked nodes.

- Section 5.3.1 and 5.3.2 present practical examples of how to select important features and assign their levels. The feature variables of each group are aggregated using a weighted mean function. Experts can assign the weight to each feature, or if there are enough groups, we can learn the weight as another hyperparameter. The aggregated influence is applied to the local parameters, reducing or increasing the rate of deterioration.

Objective III: ***Apply the deterioration model to predict the state of a given asset over time. Extend the deterioration prediction model to support more practical modelling assumptions, such as an extended description of asset states and the interaction between multiple components.***

This objective is achieved by extending the models developed in Section 4.1 and 4.2. Specifically:

- Section 4.3 shows how to use a Boolean node to evaluate whether the given prediction time is likely to higher than a time sampled from the transition distribution, for an asset with only two condition states.

- Section 4.3.1 shows how to extend this for multiple condition states. To reason about transitions between multiple states, several transition time distributions are modelled, and the Boolean node becomes a labelled node, with one state of each asset condition state. A logical function can express the probability of each state at a given time, but because the transition distributions are continuous, this simple mechanism results in a large CPT that becomes computational. The binary factorisation is used to resolve this problem.

- Section 4.3.2 shows how to model the three basic but common system configurations using BNs. For assets with components interact in parallel, fault tree construct AND gate is used as the function to link components that maximise their transition distributions or states. Similarly, for interaction in a series configuration, an OR gate is used to minimise the transition distributions or states. While for asset assembled in a bridge structure, a weighted function is used to connect components. It also discusses how to include experts' certainty level about assigning the weights of the components. The weight represents the contribution of the state of each component to the state of the asset as a system. The presented system configurations give the basic elements that can be combined or extended to model more complex configurations.

Objective IV: ***Use the developed deterioration prediction model in a real-world context, and compare its performance with other existing methods.***

Chapter 5 presents the validation of the deterioration prediction models. Specifically:

- It first validates the proposed methods using synthesis data. This validation supports the proposal of using Bayesian parameter learning technique to learn deterioration distributions and the idea of individualising assets into groups when they follow different distributions.

- A case study of deterioration of bridge decks from the US State of Wyoming using the data from the National Bridge Inventory is then used to build the proposed models. It shows how to use a modified random forest for feature selection and how to apply the selected features when building the hierarchical BN models. The learned transition distributions are assembled to provide multi-state predictions. Three approaches are used to examine this model against the data in the NBI. First, we compare the performance as the number of features considered in the model increases. We show that, in this case, the models reach a reasonable performance with three features, and adding more can only improve the performance slightly. Secondly, we compare our model with other available approaches in predicting the state of decks and shows that the developed model excels at most predictions, especially for cases where there is little inspection data available. Lastly, we show that as the prediction time increases, the performance of models drops. However, the developed model is capable of providing short term condition prediction that can count for a 1 or 2-biennial inspections in a regular inspection scheme, which is useful information for future inspection planning.

Objective V: ***Develop models that can reason and support a variety of inspection and maintenance-related decisions, and demonstrate the uses of them using real-world cases.***

This objective is achieved through further elaborations of the models, demonstrated using case studies. Specifically:

- Section 6.1 presents case studies of structural evaluation in railway bridge in Great Britain and NBI assets in the US. In both cases, we show how the models are applied given the number of condition states and the inspection data available in practice. Bridges are decomposed into components, and the interaction of the components is modelled appropriately for the two scenarios.

- Section 6.2 shows how to perform reliability analysis to support inspection decisions. This analysis can help prioritise asset to inspect, or help suggests inspection time given a reliability threshold.

- Section 4.4 introduces two modelling approaches to support decisions about repairs. The observational model suggests repair actions based on historical repair records and given constraints. The intervention model predicts what would happen if a specific type of repair action is taken. These models are based on the causal structures introduced in Section 3.2.3. These two models are applied in Section 6.3 for deck repairs in the US.

- The intervention model can be combined with the deterioration prediction model to model the deterioration of an asset after repair. Section 4.5 shows multiple instances of this model can be arranged sequentially to provide a framework to monitor the behaviour of an asset throughout its life cycle. Section 6.4 presents an example application of this idea, based on the earlier model of deck repair. It shows how the condition of a deck changes over a time period when both repairs and further deterioration are considered. A simple maintenance strategy - selecting no action, a specific repair, or deck replacement - is also illustrated. It outputs the life cycle maintenance cost over the next 100 years given different repair actions. This method gives a framework to plan maintenance at a strategical level.

Objective VI: ***Show how the modelling choices could be effectively managed for maintenance modelling with various specifications.***

This objective is achieved in Chapter 7. Specifically:

- We develop a model-based asset maintenance framework that enables engineers to manage the models without an in-depth understanding of BNs, even for problems with complex specifications. The models are encoded in the form of probabilistic dependencies within a PRM. When building a model for a specific maintenance problem, they are assembled according to a relational database that mapped from the model assumptions. Asset information is, therefore, organised in a relational database, or even a spreadsheet with multiple tables, based on the model assumptions and structure of the PRM.

- Inference is supported by the model-based language automatically, and suggestions of decision can be generated based on the query. This separation allows us to distinguish the role of users - who build the models by manipulating the relational database, from the modellers - who encodes the generic models in the early stage.

- Section 7.3 later presents a range of ground BNs generated using a prototype implementation built on these ideas. It shows how engineers can manage the modelling for different asset configurations by simply manipulating the data in a spreadsheet.

8.2 Further Work

This thesis has achieved its objective of providing a framework that can support inspection and maintenance decision making from available data and expert knowledge. To achieve the potential benefits for this framework, some further validation of the knowledge elicitation process and model performance is needed:

- Though this thesis has presented an approach to elicit parameter priors of a Weibull distribution from the knowledge of maintenance engineers who are not Bayesian specialists, it has not been confirmed whether this type of elicitation is deliverable in practice. In addition, as summarised in Section 4.6.2, apart from knowledge expressed as the priors of the parameters, there are many types of knowledge can be included in our models. So far, the applications presented in this thesis are mostly inspired by the literature. It would be helpful to interview maintenance decision makers to ensure that practicality of these applications. Additionally, we need further work to investigate the process of knowledge elicitation when there are multiple experts, for example, by hosting a series of training workshops [170, 27].

- This thesis only compares the performance of the deterioration prediction with a limited number of approaches as the complexity of the deterioration data prevents many prediction techniques being used: in most cases, we only have a small amount of data, and these data are often uncertain (censored). Recent advances in transfer learning (see Pan et al. [135]) could have the potential to tackle the problem of the limited data available for some asset groups. Censored data has been intensively studied in the area of survival analysis (see Klein and Moeschberger [78]). Borrowing ideas from this area could help understand how to better encode and reason uncertainty from censored data.

In addition to the improvements addressed above, we also identify some extensions that could expand the benefits of our framework:

- The effectiveness of different repair actions given asset's current state depends on a CPT that needs to be elicited from experts. In Section 6.3, we resolved this by basing the CPT on a table published in a maintenance guideline. Unfortunately, elicitation purely from engineers may be difficult in most cases, and the results maybe bias or inaccurate. An alternative direction would be to develop models, similar to the models used for learning deterioration proposed in this thesis, to learn repair effectiveness from both experts and data. The main reason this is not performed in this thesis is that there is no publicly available data. Negotiation to acquire maintenance data from organisations such as the US Department of Transportation and National Rail in the UK is needed.

- Further work could be done to integrate our framework with an optimisation algorithm. We could use it to plan maintenance activity at a strategical level: for example, to select optimal repair actions or to perform repairs at the most cost-effective times. There is a wide range of techniques to choose from: for example, heuristics technique such as genetic algorithm has been employed in many studies to tackle similar maintenance problems (see examples in Audley [9], Seif and Rabbani [150], Le [94] and Yang et al. [186]).

- The PRM extension of Bayesian networks has been used to advantage in this thesis to organise complex models. PRMs could have more uses. They can be extended to reason about 'link uncertainty', where we are uncertain which two objects are related (see Getoor [54]). This might be useful if part of the information is missing or uncertain or, for example, it is uncertain to which group a newly built bridge should belong. Similarly, PRM class hierarchies might be used to determine the grouping of assets by evaluating the performance of different hierarchical structures.

References

[1] Abramowitz, M. and Stegun, I. A. (1964). *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*. Dover Publications, ISBN: 0486612724.

[2] Agena Ltd (2019). AgenaRisk: Bayesian Network Software for Risk Analysis and Decision Making. https://www.agenarisk.com. Last accessed 10 January 2019.

[3] Agrawal, A. K., Kawaguchi, A., and Chen, Z. (2009). Bridge Element Deterioration Rates. Research Report to New York State Department of Transportation, Report No. C-01-51.

[4] Andrade, A. R. and Teixeira, P. F. (2015). Statistical Modelling of Railway Track Geometry Degradation Using Hierarchical Bayesian Models. *Reliability Engineering & System Safety*, 142:169–183, DOI: 10.1016/j.ress.2015.05.009.

[5] Andrews, J. (2013). A Modelling Approach to Railway Track Asset Management. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 227(1):56–73, DOI: 10.1177/0954409712452235.

[6] Andrews, J. and Fecarotti, C. (2017). System Design and Maintenance Modelling for Safety in Extended Life Operation. *Reliability Engineering & System Safety*, 163:95–108, DOI: 10.1016/j.ress.2017.01.024.

[7] Arnold, A., Point, G., Griffault, A., and Rauzy, A. (1999). The AltaRica Formalism for Describing Concurrent Systems. *Fundamenta Informaticae*, 40(2, 3):109–124, DOI: 10.3233/FI-1999-402302.

[8] Arunraj, N. S. and Maiti, J. (2007). Risk-Based Maintenance - Techniques and Applications. *Journal of Hazardous Materials*, 142(3):653–661, DOI: 10.1016/j.jhazmat.2006.06.069.

[9] Audley, M. (2014). *Rail Track Geometry Degradation and Maintenance Decision Making*. PhD thesis, University of Nottingham.

[10] Ayyub, B. M. and McCuen, R. H. (2011). *Probability, Statistics, and Reliability for Engineers and Scientists*. CRC Press, ISBN: 978-1-4398-9533-7.

[11] Barlow, R. and Hunter, L. (1960). Optimum Preventive Maintenance Policies. *Operations Research*, 8(1):90–100, DOI: 10.1287/opre.8.1.90.

[12] Barlow, R. E. and Proschan, F. (1996). *Mathematical Theory of Reliability*. Siam, ISBN: 978-1-61197-119-4.

[13] BBC News (2018). Italy Bridge Collapse: What We Know So Far. www.bbc.co.uk/news/world-europe-45193452. Published on 19 August 2018.

[14] Bemment, S. D., Goodall, R. M., Dixon, R., and Ward, C. P. (2018). Improving the Reliability and Availability of Railway Track Switching by Analysing Historical Failure Data and Introducing Functionally Redundant Subsystems. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of rail and rapid transit*, 232(5):1407–1424, DOI: 10.1177/0954409717727879.

[15] Berg, M. and Epstein, B. (1976). A Modified Block Replacement Policy. *Naval Research Logistics Quarterly*, 23(1):15–24, DOI: 10.1002/nav.3800230103.

[16] Bergman, B. (2006). Optimal Replacement Under a General Failure Model. *Advances in Applied Probability*, 10(02):431–451, DOI: 10.2307/1426944.

[17] Bishop, C. M. (2013). Model-Based Machine Learning. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984), DOI: 10.1098/rsta.2012.0222.

[18] Bobbio, A., Portinale, L., Minichino, M., and Ciancamerla, E. (2001). Improving the Analysis of Dependable Systems by Mapping Fault Trees into Bayesian Networks. *Reliability Engineering & System Safety*, 71(3):249–260, DOI: 10.1016/S0951-8320(00)00077-6.

[19] Bowles, J. B. (2002). Commentary - Caution: Constant Failure-Rate Models May Be Hazardous to Your Design. *IEEE Transactions on Reliability*, 51(3):375–377, DOI: 10.1109/TR.2002.801850.

[20] Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32, DOI: 10.1023/A:1010933404324.

[21] Castanier, B., Bérenguer, C., and Grall, A. (2003). A Sequential Condition-Based Repair/Replacement Policy with Non-Periodic Inspections for a System Subject to Continuous Wear. *Applied Stochastic Models in Business and Industry*, 19(4):327–347, DOI: 10.1002/asmb.493.

[22] Cesare, M. A., Santamarina, C., Turkstra, C., and Vanmarcke, E. H. (1992). Modeling Bridge Deterioration with Markov Chains. *Journal of Transportation Engineering*, 118(6):820–833, DOI: 10.1061/(ASCE)0733-947X(1992)118:6(820).

[23] Chang, M. (2016). *Investigating and Improving Bridge Management System Methodologies Under Uncertainty*. PhD thesis, Utah State University.

[24] Chang, M., Maguire, M., and Sun, Y. (2017). Framework for Mitigating Human Bias in Selection of Explanatory Variables for Bridge Deterioration Modeling. *Journal of Infrastructure Systems*, 23(3):04017002, DOI: 10.1061/(asce)is.1943-555x.0000352.

[25] Cheng, Z. and Remenyte-Prescott, R. (2018). Two Probabilistic Life-Cycle Maintenance Models for the Deteriorating Pavement. *Eksploatacja i Niezawodnosc – Maintenance and Reliability*, 20(3):394–404, DOI: 10.17531/ein.2018.3.7.

[26] Constantinou, A. C. (2013). *Bayesian Networks for Prediction, Risk Assessment and Decision Making in an Inefficient Association Football Gambling Market*. PhD thesis, Queen Mary University of London.

[27] Constantinou, A. C., Fenton, N., and Neil, M. (2016). Integrating Expert Knowledge with Data in Bayesian Networks: Preserving Data-Driven Expectations When the Expert Variables Remain Unobserved. *Expert Systems with Applications*, 56:197–208, DOI: 10.1016/j.eswa.2016.02.050.

[28] Constantinou, A. C., Yet, B., Fenton, N., Neil, M., and Marsh, W. (2015). Value of Information Analysis for Interventional and Counterfactual Bayesian Networks in Forensic Medical Sciences. *Artificial Intelligence in Medicine*, 66:41–52, DOI: 10.1016/j.artmed.2015.09.002.

[29] Coolen, F. (1996). On Bayesian Reliability Analysis with Informative Priors and Censoring. *Reliability Engineering & System Safety*, 53(1):91–98, DOI: 10.1016/0951-8320(96)00037-3.

[30] Cooper, G. F. (1990). The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks. *Artificial Intelligence*, 42(2-3):393–405, DOI: 10.1016/0004-3702(90)90060-D.

[31] D'Amico, G., Morabito, A., D'Amico, M., Pasta, L., Malizia, G., Rebora, P., and Valsecchi, M. G. (2018). Clinical States of Cirrhosis and Competing Risks. *Journal of Hepatology*, 68(3):563–576, DOI: 10.1016/j.jhep.2017.10.020.

[32] De Almeida, A. T., Ferreira, R. J. P., and Cavalcante, C. A. V. (2015). A Review of the Use of Multicriteria and Multi-Objective Models in Maintenance and Reliability. *IMA Journal of Management Mathematics*, 26(3):249–271, DOI: 10.1093/imaman/dpv010.

[33] de Melo, A. C. and Sanchez, A. J. (2008). Software Maintenance Project Delays Prediction Using Bayesian Networks. *Expert Systems with Applications*, 34(2):908–919, DOI: 10.1016/j.eswa.2006.10.040.

[34] Díez, F. (1993). Parameter Adjustment in Bayes Networks. The Generalized Noisy OR–Gate. In *Uncertainty in Artificial Intelligence*, pages 99–105. Elsevier, DOI: 10.1016/b978-1-4832-1451-1.50016-0.

[35] Dilip, D. M., Ravi, P., and Babu, G. S. (2013). System Reliability Analysis of Flexible Pavements. *Journal of Transportation Engineering*, 139(10):1001–1009, DOI: 10.1061/(ASCE)TE.1943-5436.0000578.

[36] Do, P., Voisin, A., Levrat, E., and Iung, B. (2015). A Proactive Condition-Based Maintenance Strategy with Both Perfect and Imperfect Maintenance Actions. *Reliability Engineering & System Safety*, 133:22–32, DOI: 10.1016/j.ress.2014.08.011.

[37] Dohi, T., Kaio, N., and Osaki, S. (2000). A Graphical Method to Repair-Cost Limit Replacement Policies with Imperfect Repair. *Mathematical and Computer Modelling*, 31(10-12):99–106, DOI: 10.1016/S0895-7177(00)00076-5.

[38] Dohi, T., Kaio, N., and Osaki, S. (2003). A New Graphical Method to Estimate the Optimal Repair-Time Limit with Incomplete Repair and Discounting. *Computers and Mathematics with Applications*, 46(7):999–1007, DOI: 10.1016/S0898-1221(03)90114-3.

[39] Droguett, E. L., das Chagas Moura, M., Jacinto, C. M., and Silva Jr, M. F. (2008). A semi-markov model with bayesian belief network based human error probability for availability assessment of downhole optical monitoring system. *Simulation Modelling Practice and Theory*, 16(10):1713–1727, DOI: 10.1016/j.simpat.2008.08.011.

[40] Duan, R.-x. and Zhou, H.-l. (2012). A New Fault Diagnosis Method Based on Fault Tree and Bayesian Networks. *Energy Procedia*, 17:1376–1382, DOI: 10.1016/j.egypro.2012.02.255.

[41] Enright, M. P. and Frangopol, D. M. (1999). Condition Prediction of Deteriorating Concrete Bridges. *Journal of Structural Engineering*, 125(10):1118–1125, DOI: 10.1061/(ASCE)0733-9445(1999)125:10(1118).

[42] Estefan, J. A. (2007). Survey of Model-Based Systems Engineering (MBSE) Methodologies. *Incose MBSE Focus Group*, 25(8).

[43] Estes, A. C. and Frangopol, D. M. (2001). Minimum Expected Cost-Oriented Optimal Maintenance Planning for Deteriorating Structures: Application to Concrete Bridge Decks. *Reliability Engineering & System Safety*, 73(3):281–291, DOI: 10.1016/S0951-8320(01)00044-8.

[44] Federal Highway Administration (2018). Bridge Guide Preservation: Maintaining a Resilient Infrastructure to Preserve Mobility. Report to U.S. Department of Transportation: Federal Highway Administration.

[45] Fenton, N. and Neil, M. (2018). *Risk Assessment and Decision Analysis with Bayesian Networks*. CRC Press, ISBN: 978-1439809105.

[46] Fenton, N. E., Neil, M., and Caballero, J. G. (2007). Using Ranked Nodes to Model Qualitative Judgments in Bayesian Networks. *IEEE Transactions on Knowledge and Data Engineering*, 19(10):1420–1432, DOI: 10.1109/TKDE.2007.1073.

[47] Ferguson, N., Brock, G., and Datta, S. (2015). msSurv : An R Package for Nonparametric Estimation of Multistate Models . *Journal of Statistical Software*, 50(14):1–24, DOI: 10.18637/jss.v050.i14.

[48] Fink, D. (1997). A Compendium of Conjugate Priors. Research Report to Montana State University.

[49] Fink, O., Zio, E., and Weidmann, U. (2014). Predicting Component Reliability and Level of Degradation with Complex-Valued Neural Networks. *Reliability Engineering & System Safety*, 121:198–206, DOI: 10.1016/j.ress.2013.08.004.

[50] Frangopol, D. M., Kallen, M.-J., and van Noortwijk, J. M. (2004). Probabilistic Models for Life-Cycle Performance of Deteriorating Structures: Review and Future Directions. *Progress in Structural Engineering and Materials*, 6(4):197–212, DOI: 10.1002/pse.180.

[51] Frangopol, D. M., Lin, K.-Y., and Estes, A. C. (1997). Life-Cycle Cost Design of Deteriorating Structures. *Journal of Structural Engineering*, 123(10):1390–1401, DOI: 10.1061/(ASCE)0733-9445(1997)123:10(1390).

[52] Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. Chapman and Hall/CRC, ISBN: 978-1439840955.

[53] Getoor, L. and Taskar, B. (2007). *Introduction to Statistical Relational Learning*. MIT Press, ISBN: 978-0262072885.

[54] Getoor, L. C. (2002). *Learning Statistical Models from Relational Data*. PhD thesis, Stanford University.

[55] Ghahramani, Z. (2015). Probabilistic Machine Learning and Artificial Intelligence. *Nature*, 521(7553):452–459, DOI: 10.1038/nature14541.

[56] Gigerenzer, G. and Hoffrage, U. (1995). How to Improve Bayesian Reasoning Without Instruction: Frequency Formats. *Psychological Review*, 102(4):684–704, DOI: 10.1037/0033-295X.102.4.684.

[57] Goldberg, A. and Robson, D. (1983). *Smalltalk-80: The Language and Its Implementation*. Longman Higher Education, ISBN: 978-0201113716.

[58] Gorjian, N., Ma, L., Sun, Y., Yarlagadda, P., and Mittinty, M. (2011). A Review on Degradation Models in Reliability Analysis. In *Engineering Asset Lifecycle Management*, pages 369–384. Springer, DOI: 10.1007/978-0-85729-320-6_42.

[59] Greenland, S. (2005). Multiple-Bias Modelling for Analysis of Observational Data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 168(2):267–306, DOI: 10.1111/j.1467-985X.2004.00349.x.

[60] Griffith, W. S. (1980). Multistate Reliability Models. *Journal of Applied Probability*, 17(3):735–744, DOI: 10.2307/3212967.

[61] Grigoriev, A., Van De Klundert, J., and Spieksma, F. C. (2006). Modeling and Solving the Periodic Maintenance Problem. *European Journal of Operational Research*, 172(3):783–797, DOI: 10.1016/j.ejor.2004.11.013.

[62] Groeneveld, R. A. (1986). Skewness for the Weibull Family. *Statistica Neerlandica*, 40(3):135–140, DOI: 10.1111/j.1467-9574.1986.tb01509.x.

[63] Guler, H., Jovanovic, S., and Evren, G. (2011). Modelling Railway Track Geometry Deterioration. *Proceedings of the Institution of Civil Engineers-Transport*, 164(2):65–75, DOI: 10.1680/tran.2011.164.2.65.

[64] Hagmayer, Y., Sloman, S. A., Lagnado, D. A., and Waldmann, M. R. (2007). Causal Reasoning Through Intervention. In *Causal Learning: Psychology, Philosophy, and Computation*, pages 86–100. Oxford University Press, ISBN: 978-0195176803.

[65] Hajipour, Y. and Taghipour, S. (2016). Non-Periodic Inspection Optimization of Multi-Component and K-Out-Of-M Systems. *Reliability Engineering & System Safety*, 156:228–243, DOI: 10.1016/j.ress.2016.08.008.

[66] Han, D., Kaito, K., and Kobayashi, K. (2014). Application of Bayesian Estimation Method with Markov Hazard Model to Improve Deterioration Forecasts for Infrastructure Asset Management. *KSCE Journal of Civil Engineering*, 18(7):2107–2119, DOI: 10.1007/s12205-012-0070-6.

[67] He, Q., Li, H., Bhattacharjya, D., Parikh, D. P., and Hampapur, A. (2013). Railway Track Geometry Defect Modeling: Deterioration, Derailment Risk and Optimal Repair. In *Proceedings of the Transportation Research Board 92nd Annual Meeting*. Transportation Research Board.

[68] Heng, A., Zhang, S., Tan, A. C., and Mathew, J. (2009). Rotating Machinery Prognostics: State of the Art, Challenges and Opportunities. *Mechanical Systems and Signal Processing*, 23(3):724–739, DOI: 10.1016/j.ymssp.2008.06.009.

[69] Higgins, A. (1998). Scheduling of Railway Track Maintenance Activities and Crews. *Journal of the Operational Research Society*, 49(10):1026–1033, DOI: 10.1057/palgrave.jors.2600612.

[70] Hong, F. and Prozzi, J. A. (2006). Estimation of Pavement Performance Deterioration Using Bayesian Approach. *Journal of Infrastructure Systems*, 12(2):77–86, DOI: 10.1061/(ASCE)1076-0342(2006)12:2(77).

[71] Jang, H., Lee, S., and Kim, S. W. (2010). Bayesian Analysis for Zero-Inflated Regression Models with the Power Prior: Applications to Road Safety Countermeasures. *Accident Analysis & Prevention*, 42(2):540–547, DOI: 10.1016/j.aap.2009.08.022.

[72] Jensen, K. (2013). *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*. Springer Science & Business Media, ISBN: 978-3-662-03241-1.

[73] Jiang, R. and Murthy, D. (2011). A Study of Weibull Shape Parameter: Properties and Significance. *Reliability Engineering & System Safety*, 96(12):1619–1626, DOI: 10.1016/j.ress.2011.09.003.

[74] Jiang, Y., Saito, M., and Sinha, K. C. (1988). Bridge Performance Prediction Model Using the Markov Chain. *Transportation Research Record*, 1180:25–32.

[75] Kallen, M. and Van Noortwijk, J. (2006). Statistical Inference for Markov Deterioration Models of Bridge Conditions in the Netherlands. In *Proceedings of the Third International Conference on Bridge Maintenance, Safety and Management (IABMAS)*, pages 16–19. CRC Press.

[76] Kang, C. and Golay, M. (1999). A Bayesian Belief Network-Based Advisory System for Operational Availability Focused Diagnosis of Complex Nuclear Power Systems. *Expert Systems with Applications*, 17(1):21–32, DOI: 10.1016/S0957-4174(99)00018-4.

[77] Khakzad, N., Khan, F., and Amyotte, P. (2011). Safety Analysis in Process Facilities: Comparison of Fault Tree and Bayesian Network Approaches. *Reliability Engineering & System Safety*, 96(8):925–932, DOI: 10.1016/j.ress.2011.03.012.

[78] Klein, J. P. and Moeschberger, M. L. (2006). *Survival Analysis: Techniques for Censored and Truncated Data*. Springer Science & Business Media, ISBN: 978-0-387-21645-4.

[79] Kleiner, Y. (2001). Scheduling Inspection and Renewal of Large Infrastructure Assets. *Journal of Infrastructure Systems*, 7(4):136–143, DOI: 10.1061/(ASCE)1076-0342(2001)7:4(136).

[80] Kohavi, R. and John, G. H. (1997). Wrappers for Feature Subset Selection. *Artificial Intelligence*, 97(1-2):273–324, DOI: 10.1016/S0004-3702(97)00043-X.

[81] Koller, D. (1999). Probabilistic Relational Models. In *International Conference on Inductive Logic Programming*, pages 3–13. Springer.

[82] Koller, D. and Pfeffer, A. (1997). Object-Oriented Bayesian Networks. In *Proceedings of the Thirteenth conference on Uncertainty in Artificial Intelligence*, pages 302–313. Morgan Kaufmann Publishers Inc., ISBN: 1558604855.

[83] Kolowrocki, K. (2004). *Reliability of Large Systems*. Elsevier, ISBN: 978-0-08-044429-1.

[84] Kozlov, A. V. and Koller, D. (1997). Nonuniform Dynamic Discretization in Hybrid Networks. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 314–325. Morgan Kaufmann Publishers Inc.

[85] Kruschke, J. K. and Vanpaemel, W. (2015). Bayesian Estimation in Hierarchical Models. In *The Oxford Handbook of Computational and Mathematical Psychology*, pages 279–299. Oxford University Press.

[86] Kuipers, B., Moskowitz, A. J., and Kassirer, J. P. (1988). Critical Decisions Under Uncertainty: Representation and Structure. *Cognitive Science*, 12(2):177–210, DOI: 10.1207/s15516709cog1202_2.

[87] Kuo, W. and Prasad, V. R. (2000). An Annotated Overview of System-Reliability Optimization. *IEEE Transactions on Reliability*, 49(2):176–187, DOI: 10.1109/24.877336.

[88] Kursa, M. B. and Rudnicki, W. R. (2010). Feature Selection with the Boruta Package. *Journal of Statistical Software*, 36(11):1–13, DOI: 10.18637/jss.v036.i11.

[89] Langseth, H. and Portinale, L. (2007). Bayesian Networks in Reliability. *Reliability Engineering & System Safety*, 92(1):92–108, DOI: 10.1016/j.ress.2005.11.037.

[90] Laskey, K. B. and Mahoney, S. M. (1997). Network Fragments: Representing Knowledge for Constructing Probabilistic Models. In *Proceedings of the Thirteenth conference on Uncertainty in Artificial Intelligence*, pages 334–341. Morgan Kaufmann Publishers Inc., ISBN: 1558604855.

[91] Laskey, K. B. and Mahoney, S. M. (2000). Network Engineering for Agile Belief Network Models. *IEEE Transactions on Knowledge and Data Engineering*, 12(4):487–498, DOI: 10.1109/69.868902.

[92] Lauritzen, S. L. (1992). Propagation of Probabilities, Means, and Variances in Mixed Graphical Association Models. *Journal of the American Statistical Association*, 87(420):1098–1108, DOI: 10.1080/01621459.1992.10476265.

[93] Lauritzen, S. L. and Jensen, F. (2001). Stable Local Computation with Conditional Gaussian Distributions. *Statistics and Computing*, 11(2):191–203, DOI: 10.1023/A:1008935617754.

[94] Le, B. (2014). *Modelling Railway Bridge Asset Management*. PhD thesis, University of Nottingham.

[95] Le, B. and Andrews, J. (2016). Petri Net Modelling of Bridge Asset Management Using Maintenance-Related State Conditions. *Structure and Infrastructure Engineering*, 12(6):730–751, DOI: 10.1177/1748006X17701667.

[96] Lee, S., Kalos, N., et al. (2014). Non-Destructive Testing Methods in the Us for Bridge Inspection and Maintenance. *KSCE Journal of Civil Engineering*, 18(5):1322–1331, DOI: 10.1007/s12205-014-0633-9.

[97] Lethanh, N. and Adey, B. T. (2013). Use of Exponential Hidden Markov Models for Modelling Pavement Deterioration. *International Journal of Pavement Engineering*, 14(7):645–654, DOI: 10.1080/10298436.2012.715647.

[98] Lisagor, O., Kelly, T., and Niu, R. (2011). Model-Based Safety Assessment: Review of the Discipline and Its Challenges. In *2011 9th International Conference on Reliability, Maintainability and Safety (ICRMS)*, pages 625–632. IEEE, ISBN: 1612846661.

[99] Lisnianski, A., Frenkel, I., and Ding, Y. (2010). *Multi-State System Reliability Analysis and Optimization for Engineers and Industrial Managers*. Springer Science & Business Media, ISBN: 1849963207.

[100] Lisnianski, A., Frenkel, I., and Karagrigoriou, A. (2017). *Recent Advances in Multi-State Systems Reliability: Theory and Applications*. Springer, ISBN: 3319634232.

[101] Louit, D. M., Pascual, R., and Jardine, A. K. (2009). A Practical Procedure for the Selection of Time-To-Failure Models Based on the Assessment of Trends in Maintenance Data. *Reliability Engineering & System Safety*, 94(10):1618–1628, DOI: 10.1016/j.ress.2009.04.001.

[102] Lu, J.-C. (1992). Bayes Parameter Estimation for the Bivariate Weibull Model of Marshall-Olkin for Censored Data (Reliability Theory). *IEEE Transactions on Reliability*, 41(4):608–615, DOI: 10.1109/24.249597.

[103] Lunn, D., Jackson, C., Best, N., Spiegelhalter, D., and Thomas, A. (2012). *The BUGS Book: A Practical Introduction to Bayesian Analysis*. Chapman and Hall/CRC, ISBN: 9781584888499.

[104] Mahboob, Q. (2013). *A Bayesian Network Methodology for Railway Risk, Safety and Decision Support*. PhD thesis, Technische Universität Dresden.

[105] Mahoney, S. M. and Laskey, K. B. (1996). Network Engineering for Complex Belief Networks. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 389–396. Morgan Kaufmann Publishers Inc.

[106] Malik, M. A. K. (1979). Reliable Preventive Maintenance Scheduling. *AIIE Transactions*, 11(3):221–228, DOI: 10.1080/05695557908974463.

[107] Marquez, D., Neil, M., and Fenton, N. (2007). A New Bayesian Network Approach to Reliability Modelling. In *Mathematical Methods in Reliability (MMR07)*.

[108] Marquez, D., Neil, M., and Fenton, N. (2010). Improved Reliability Modeling Using Bayesian Networks and Dynamic Discretization. *Reliability Engineering & System Safety*, 95(4):412–425, DOI: 10.1016/j.ress.2009.11.012.

[109] Marsh, W., Nur, K., Yet, B., and Majumdar, A. (2016). Using Operational Data for Decision Making: A Feasibility Study in Rail Maintenance. *Safety and Reliability*, 36(1):35–47, DOI: 10.1080/09617353.2016.1148923.

[110] McCool, J. (2012). *Using the Weibull Distribution: Reliability, Modeling, and Inference*. John Wiley& Sons, ISBN: 1118217985.

[111] Meder, B., Gerstenberg, T., Hagmayer, Y., and Waldmann, M. R. (2010). Observing and Intervening: Rational and Heuristic Models of Causal Decision Making. *The Open Psychology Journal*, 3:119–135, DOI: 10.2174/1874350101003010119.

[112] Meder, B., Hagmayer, Y., and Waldmann, M. R. (2006). Understanding the Causal Logic of Confounds. In *Twenty-Eighth Annual Conference of the Cognitive Science Society*, pages 579–584.

[113] Medina-Oliva, G., Weber, P., and Iung, B. (2013). PRM-Based Patterns for Knowledge Formalisation of Industrial Systems to Support Maintenance Strategies Assessment. *Reliability Engineering & System Safety*, 116:38–56, DOI: 10.1016/j.ress.2013.02.026.

[114] Medina-Oliva, G., Weber, P., Levrat, E., and Iung, B. (2010). Use of Probabilistic Relational Model (PRM) for Dependability Analysis of Complex Systems. *IFAC Proceedings Volumes*, 43(8):501–506, DOI: 10.3182/20100712-3-FR-2020.00082.

[115] Mendenhall, W., Beaver, R. J., and Beaver, B. M. (2012). *Introduction to Probability and Statistics*. Cengage Learning, ISBN: 1133711677.

[116] Michigan Department of Transportation (2008). Bridge Deck Preservation Matrix. Report to Michigan Department of Transportation.

[117] Murata, T. (1989). Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580, DOI: 10.1109/5.24143.

[118] Murphy, K. P. (1998). Inference and Learning in Hybrid Bayesian Networks. Technical Report UCB/CSD-98-990, EECS Department, University of California, Berkeley.

[119] Murphy, K. P. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley.

[120] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press, ISBN: 978-0262018029.

[121] Nakagawa, T. and Kowada, M. (1983). Analysis of a System with Minimal Repair and Its Application to Replacement Policy. *European Journal of Operational Research*, 12(2):176–182, DOI: 10.1016/0377-2217(83)90221-7.

[122] Nasrollahi, M. and Washer, G. (2014). Estimating Inspection Intervals for Bridges Based on Statistical Analysis of National Bridge Inventory Data. *Journal of Bridge Engineering*, 20(9):04014104, DOI: 10.1061/(ASCE)BE.1943-5592.0000710.

[123] Neil, M., Chen, X., and Fenton, N. (2012). Optimizing the Calculation of Conditional Probability Tables in Hybrid Bayesian Networks Using Binary Factorization. *IEEE Transactions on Knowledge and Data Engineering*, 24(7):1306–1312.

[124] Neil, M., Fenton, N., and Nielson, L. (2000). Building Large-Scale Bayesian Networks. *The Knowledge Engineering Review*, 15(3):257–284, DOI: 10.1017/S0269888900003039.

[125] Neil, M., Tailor, M., and Marquez, D. (2007). Inference in Hybrid Bayesian Networks Using Dynamic Discretization. *Statistics and Computing*, 17(3):219–233, DOI: 10.1007/s11222-007-9018-y.

[126] Network Rail (2013). Management of Existing Bridges and Culverts. Report to Network Rail, Report No. NR/SP/CIV/0801.

[127] Nielsen, T. D. and Jensen, F. V. (2009). *Bayesian Networks and Decision Graphs*. Springer Science & Business Media, ISBN: 978-0-387-68282-2.

[128] Nivolianitou, Z., Leopoulos, V., and Konstantinidou, M. (2004). Comparison of Techniques for Accident Scenario Analysis in Hazardous Systems. *Journal of Loss Prevention in the Process Industries*, 17(6):467–475, DOI: 10.1016/j.jlp.2004.08.001.

[129] Noguchi, T., Fenton, N., and Neil, M. (2018). Addressing the Practical Limitations of Noisy-Or Using Conditional Inter-Causal Anti-Correlation with Ranked Nodes. *IEEE Transactions on Knowledge and Data Engineering*, DOI: 10.1109/TKDE.2018.2873314.

[130] Ohadi, A. and Micic, T. (2011). Stochastic Process Deterioration Modelling for Adaptive Inspections. In *Applications of Statistics and Probability in Civil Engineering-Proceedings of the 11th International Conference on Applications of Statistics and Probability in Civil Engineering*, pages 322–326. CRC Press.

[131] O'Hagan, A. (1998). Eliciting Expert Beliefs in Substantial Practical Applications. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(1):21–35, DOI: 10.1111/1467-9884.00114.

[132] O'Hagan, A., Buck, C. E., Daneshkhah, A., Eiser, J. R., Garthwaite, P. H., Jenkinson, D. J., Oakley, J. E., and Rakow, T. (2006). *Uncertain Judgements: Eliciting Experts' Probabilities*. John Wiley & Sons, ISBN: 9780470033319.

[133] Organisation for Economic Co-operation and Development (OECD) (2001). *Measuring Capital: OECD Manual, Annex 1 Glossary of Technical Terms Used in the Manual*. OECD Publishing.

[134] Organisation for Economic Co-operation and Development (OECD) (2006). *Infrastructure to 2030: Telecom, Land Transport, Water and Electricity*. OECD Publishing, ISBN: 9264023984.

[135] Pan, S. J., Yang, Q., et al. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, DOI: 10.1109/TKDE.2009.191.

[136] Pearl, J. (1986). Fusion, Propagation, and Structuring in Belief Networks. *Artificial Intelligence*, 29(3):241–288, DOI: 10.1016/0004-3702(86)90072-X.

[137] Pearl, J. (2009). *Causality: Models, Reasoning and Inference*. Cambridge University Press, ISBN: 978-0521773621.

[138] Percy, D. F., Kobbacy, K. A., and Fawzi, B. B. (1997). Setting Preventive Maintenance Schedules When Data Are Sparse. *International Journal of Production Economics*, 51(3):223–234, DOI: 10.1016/S0925-5273(97)00054-6.

[139] Pfeffer, A. (2009). Figaro: An Object-Oriented Probabilistic Programming Language. *Charles River Analytics Technical Report*, 137:96.

[140] Pfeffer, A., Koller, D., Milch, B., and Takusagawa, K. T. (1999). SPOOK: A System for Probabilistic Object-Oriented Knowledge Representation. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 541–550. Morgan Kaufmann Publishers Inc.

[141] Pham, H. and Wang, H. (1996). Imperfect Maintenance. *European Journal of Operational Research*, 94(3):425–438, DOI: 10.1016/S0377-2217(96)00099-9.

[142] Phares, B. M., Rolander, D. D., Graybeal, B. A., and Washer, G. A. (2001). Reliability of Visual Bridge Inspection. *Public Roads*, 64(5).

[143] Prosvirnova, T., Batteux, M., Brameret, P.-A., Cherfi, A., Friedlhuber, T., Roussel, J.-M., and Rauzy, A. (2013). The AltaRica 3.0 Project for Model-Based Safety Assessment. *IFAC Proceedings Volumes*, 46(22):127–132, DOI: 10.3182/20130904-3-UK-4041.00028.

[144] Rafiq, M. I., Chryssanthopoulos, M. K., and Sathananthan, S. (2015). Bridge Condition Modelling and Prediction Using Dynamic Bayesian Belief Networks. *Structure and Infrastructure Engineering*, 11(1):38–50, DOI: 10.1080/15732479.2013.879319.

[145] Rama, D. and Andrews, J. (2013). A System-Wide Modelling Approach to Railway Infrastructure Asset Management. In *Proceedings of the 20th Advances in Risk and Reliability Technology Symposium*, pages 7–22.

[146] Ryan, T. W., Hartle, R. A., Mann, J. E., and Danovich, L. J. (2006). Bridge Inspector's Reference Manual. Research Report to U.S Department of Transportation, Report No. FHWA NHI.

[147] Salmerón, A., Rumí, R., Langseth, H., Nielsen, T. D., and Madsen, A. L. (2018). A Review of Inference Algorithms for Hybrid Bayesian Networks. *Journal of Artificial Intelligence Research*, 62:799–828, DOI: 10.1613/jair.1.11228.

[148] Scholten, L., Scheidegger, A., Reichert, P., and Maurer, M. (2013). Combining Expert Knowledge and Local Data for Improved Service Life Modeling of Water Supply Networks. *Environmental Modelling & Software*, 42:1–16, DOI: 10.1016/j.envsoft.2012.11.013.

[149] Scutari, M. and Denis, J.-B. (2014). *Bayesian Networks: With Examples in R*. Chapman and Hall/CRC, ISBN: 978-1482225587.

[150] Seif, J. and Rabbani, M. (2014). Component Based Life Cycle Costing in Replacement Decisions. *Journal of Quality in Maintenance Engineering*, 20(4):436–452, DOI: 10.1108/JQME-08-2013-0053.

[151] Shachter, R. D. (1988). Probabilistic Inference and Influence Diagrams. *Operations Research*, 36(4):589–604, DOI: 10.1287/opre.36.4.589.

[152] Shafahi, Y. and Hakhamaneshi, R. (2009). Application of a Maintenance Management Model for Iranian Railways Based on the Markov Chain and Probabilistic Dynamic Programming. *International Journal of Science and Technology. Transaction A: Civil Engineering*, 16(1):87–97.

[153] Shafer, G. R. and Shenoy, P. P. (1990). Probability Propagation. *Annals of Mathematics and Artificial Intelligence*, 2(1-4):327–351, DOI: 10.1007/BF01531015.

[154] Shang, H. (2015). *Maintenance Modelling, Simulation and Performance Assessment for Railway Asset Management*. PhD thesis, University of Technology of Troyes.

[155] Silva, R. (2016). Observational-Interventional Priors for Dose-Response Learning. In *Advances in Neural Information Processing Systems*, pages 1561–1569.

[156] Siu, N. O. and Kelly, D. L. (1998). Bayesian Parameter Estimation in Probabilistic Risk Assessment. *Reliability Engineering & System Safety*, 62(1-2):89–116, DOI: 10.1016/S0951-8320(97)00159-2.

[157] Soleimanmeigouni, I., Ahmadi, A., and Kumar, U. (2018). Track Geometry Degradation and Maintenance Modelling: A Review. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 232(1):73–102, DOI: 10.1177/0954409716657849.

[158] Soliman, A. A., Ellah, A. A., and Sultan, K. (2006). Comparison of Estimates Using Record Statistics from Weibull Model: Bayesian and Non-Bayesian Approaches. *Computational Statistics & Data Analysis*, 51(3):2065–2077, DOI: 10.1016/j.csda.2005.12.020.

[159] Spiegelhalter, D., Thomas, A., Best, N., and Gilks, W. (1996). BUGS - Bayesian Inference Using Gibbs Sampling (Version ii). *MRC Biostatistics Unit, Institute of Public Health, Cambridge, UK*, pages 1–59.

[160] Spirtes, P., Glymour, C. N., Scheines, R., Heckerman, D., Meek, C., Cooper, G., and Richardson, T. (2000). *Causation, Prediction, and Search*. MIT Press, ISBN: 9780262194402.

[161] Stewart, M. G. (2001). Reliability-Based Assessment of Ageing Bridges Using Risk Ranking and Life Cycle Cost Decision Analyses. *Reliability Engineering & System Safety*, 74(3):263–273, DOI: 10.1016/S0951-8320(01)00079-5.

[162] Stewart, M. G. and Rosowsky, D. V. (1998). Structural Safety and Serviceability of Concrete Bridges Subject to Corrosion. *Journal of Infrastructure Systems*, 4(4):146–155, DOI: 10.1061/(ASCE)1076-0342(1998)4:4(146).

[163] Stoll, H. G. and Garver, L. J. (1989). *Least-Cost Electric Utility Planning*. Wiley New York, ISBN: 0471636142.

[164] Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., and Zeileis, A. (2008). Conditional Variable Importance for Random Forests. *BMC Bioinformatics*, 9(1):307, DOI: 10.1186/1471-2105-9-307.

[165] Strobl, C., Boulesteix, A.-L., Zeileis, A., and Hothorn, T. (2007). Bias in Random Forest Variable Importance Measures: Illustrations, Sources and a Solution. *BMC Bioinformatics*, 8(1):25, DOI: 10.1186/1471-2105-8-25.

[166] Tan, C. M. and Raghavan, N. (2008). A Framework to Practical Predictive Maintenance Modeling for Multi-State Systems. *Reliability Engineering & System Safety*, 93(8):1138–1150, DOI: 10.1016/j.ress.2007.09.003.

[167] Thompson, P. D., Small, E. P., Johnson, M., and Marshall, A. R. (1998). The Pontis Bridge Management System. *Structural engineering international*, 8(4):303–308, DOI: 10.2749/101686698780488758.

[168] Tillman, F. A., Hwang, C.-L., and Kuo, W. (1977). Optimization Techniques for System Reliability with Redundancy: A Review. *IEEE Transactions on Reliability*, 26(3):148–155, DOI: 10.1109/TR.1977.5220100.

[169] Tsuda, Y., Kaito, K., Aoki, K., and Kobayashi, K. (2006). Estimating Markovian Transition Probabilities for Bridge Deterioration Forecasting. *Structural Engineering/Earthquake Engineering*, 23(2):241–256, DOI: 10.2208/jsceseee.23.241s.

[170] Van Der Fels-Klerx, I. H., Goossens, L. H., Saatkamp, H. W., and Horst, S. H. (2002). Elicitation of Quantitative Data from a Heterogeneous Expert Panel: Formal Process and Application in Animal Health. *Risk Analysis*, 22(1):67–81, DOI: 10.1111/0272-4332.t01-1-00007.

[171] Van der Vaart, A. W. (2000). *Asymptotic Statistics*, volume 3. Cambridge university press, ISBN: 978-0521784504.

[172] Van Noortwijk, J. and Pandey, M. (2004). A Stochastic Deterioration Process for Time-Dependent Reliability Analysis. In *Proceedings of the Eleventh IFIP WG*, volume 7, pages 259–265. A.A. Balkema Publishers.

[173] Vermeer, T. E., Patton, T. K., and Styles, A. K. (2011). Reporting of General Infrastructure Assets Under Gasb Statement No. 34. *Accounting Horizons*, 25(2):381–407, DOI: doi/10.2308/acch-10029.

[174] Wang, H. (2002). A Survey of Maintenance Policies of Deteriorating Systems. *European Journal of Operational Research*, 139(3):469–489, DOI: 10.1016/S0377-2217(01)00197-7.

[175] Wang, H. and Pham, H. (1999). Some Maintenance Models and Availability With Imperfect Maintenance in Production Systems. *Annals of Operations Research*, 91:305–318, DOI: 10.1023/A:1018910109348.

[176] Wang, S. and Liu, M. (2014). Two-Stage Hybrid Flow Shop Scheduling with Preventive Maintenance Using Multi-Objective Tabu Search Method. *International Journal of Production Research*, 52(5):1495–1508, DOI: 10.1080/00207543.2013.847983.

[177] Washer, G., Nasrollahi, M., Applebury, C., Connor, R., Ciolko, A., Kogler, R., Fish, P., and Forsyth, D. (2014). *Proposed Guideline for Reliability-Based Bridge Inspection Practices*. Transportation Research Board, ISBN: 0309307910.

[178] Weber, P. and Jouffe, L. (2006). Complex System Reliability Modelling with Dynamic Object Oriented Bayesian Networks (DOOBN). *Reliability Engineering & System Safety*, 91(2):149–162, DOI: 10.1016/j.ress.2005.03.006.

[179] Weber, P., Medina-Oliva, G., Simon, C., and Iung, B. (2012). Overview on Bayesian Networks Applications for Dependability, Risk Analysis and Maintenance Areas. *Engineering Applications of Artificial Intelligence*, 25(4):671–682, DOI: 10.1016/j.engappai.2010.06.002.

[180] Weber, P., Suhner, M.-C., and Iung, B. (2001). System Approach-Based Bayesian Network to Aid Maintenance of Manufacturing Process. In *6th IFAC Symposium on Cost Oriented Automation, Low Cost Automation.* IFAC.

[181] Wei, P., Lu, Z., and Song, J. (2015). Variable Importance Analysis: A Comprehensive Review. *Reliability Engineering & System Safety*, 142:399–432, DOI: 10.1016/j.ress.2015.05.018.

[182] Weseman, W. (1995). Recording and Coding Guide for the Structure Inventory and Appraisal of the Nation's Bridges. Report to United States Department of Transportation, Federal Highway Administration, USA.

[183] Wilks, D. S. (2011). *Statistical Methods in the Atmospheric Sciences*. Academic Press, ISBN: 0123850223.

[184] Winn, E. K. and Burgueño, R. (2013). Development and Validation of Deterioration Models for Concrete Bridge Decks. Phase 1: Artificial Intelligence Models and Bridge Management System. Research Report to Michigan DOT, Report No. CEE-RR – 2013/01.

[185] Wu, S. and Zuo, M. J. (2010). Linear and Nonlinear Preventive Maintenance Models. *IEEE Transactions on Reliability*, 59(1):242–249, DOI: 10.1109/TR.2010.2041972.

[186] Yang, C., Remenyte-Prescott, R., and Andrews, J. (2015). Pavement Maintenance Scheduling Using Genetic Algorithms. *International Journal of Performability Engineering*, 11(2):135–152.

[187] Yianni, P. C. (2017). *A Modelling Approach to Railway Bridge Asset Management*. PhD thesis, University of Nottingham.

[188] Yianni, P. C., Neves, L. C., Rama, D., Andrews, J., and Dean, R. (2016). Incorporating Local Environmental Factors into Railway Bridge Asset Management. *Engineering Structures*, 128:362–373, DOI: 10.1016/j.engstruct.2016.09.038.

[189] Yianni, P. C., Rama, D., Neves, L. C., Andrews, J., and Castlo, D. (2017). A Petri-Net-Based Modelling Approach to Railway Bridge Asset Management. *Structure and Infrastructure Engineering*, 13(2):287–297, DOI: 10.1080/15732479.2016.1157826.

[190] Zhang, H. and Marsh, D. W. R. Learning from Uncertain Data, Knowledge and Similar Groups: Individualised Multi-State Deterioration Prediction for Infrastructure Asset. *In Submission*.

[191] Zhang, H. and Marsh, D. W. R. Managing Infrastructure Asset: Bayesian-Based Models for Inspection and Maintenance Decisions Reasoning and Planning. *In Submission*.

[192] Zhang, H. and Marsh, D. W. R. (2016). Bayesian Network Models for Making Maintenance Decisions from Data and Expert Judgment. In *European Safety and Reliability Conference 2016 (ESREL 2016)*, pages 1056–1063. CRC Press, DOI: 10.1201/9781315374987.

[193] Zhang, H. and Marsh, D. W. R. (2018a). Generic Bayesian Network Models for Making Maintenance Decisions from Available Data and Expert Knowledge. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 232(5):505–523, DOI: 10.1177/1748006X17742765.

[194] Zhang, H. and Marsh, D. W. R. (2018b). Towards A Model-Based Asset Deterioration Framework Represented by Probabilistic Relational Models. In *European Safety and Reliability Conference 2018 (ESREL 2018)*, pages 671–679. Taylor & Francis Group, DOI: 10.1201/9781351174664-83.

[195] Zhang, R. and Mahadevan, S. (2000). Model Uncertainty and Bayesian Updating in Reliability-Based Inspection. *Structural Safety*, 22(2):145–160, DOI: 10.1016/S0167-4730(00)00005-9.

[196] Zio, E. (2009). Reliability Engineering: Old Problems and New Challenges. *Reliability Engineering & System Safety*, 94(2):125–141, DOI: 10.1016/j.ress.2008.06.002.