# Transparency: From Tractability to Model Explanations

*Ioannis Papantonis*

Doctor of Philosophy

Artificial Intelligence Applications Institute

School of Informatics

The University of Edinburgh

2023

# Abstract

As artificial intelligence (AI) and machine learning (ML) models get increasingly incorporated into critical applications, ranging from medical diagnosis to loan approval, they show a tremendous potential to impact society in a beneficial way, however, this is predicated on establishing a *transparent* relationship between humans and automation. In particular, transparency requirements span across multiple dimensions, incorporating both technical and societal aspects, in order to promote the responsible use of AI/ML.

In this thesis we present contributions along both of these axes, starting with the technical side and *model transparency*, where we study ways to enhance *tractable probabilistic models* (TPMs) with properties that enable acquiring an in-depth understanding of their decision-making process. Following this, we expand the scope of our work, studying how providing explanations about a model's predictions influences the extent to which humans *understand* and *collaborate* with it, and finally we design an introductory course into the emerging field of explanations in AI to foster the *competent* use of the developed tools and methodologies.

In more detail, the complex design of TPMs makes it very challenging to extract information that conveys meaningful insights, despite the fact that they are closely related to Bayesian networks (BNs), which readily provide such information. This has led to TPMs being viewed as *black-boxes*, in the sense that their internal representations are elusive, in contrast to BNs. The first part of this thesis challenges this view, focusing on the question of whether it is feasible to extend certain transparent features of BNs to TPMs. We start with considering the problem of transforming TPMs into alternative graphical models in a way that makes their internal representations easy to inspect. Furthermore, we study the utility of existing algorithms in causal applications, where we identify some significant limitations. To remedy this situation, we propose a set of algorithms that result in transformations that accurately uncover the internal representations of TPMs.

Following this result, we look into the problem of incorporating probabilistic constraints into TPMs. Although it is well known that BNs satisfy this property, the complex structure of TPMs impedes applying the same arguments, thus advances on this problem have been very limited. Having said that, in this thesis we provide formal proofs that TPMs can be made to satisfy both probabilistic and causal constraints through parameter manipulation, showing that incorporating a constraint corresponds to solving a system of multilinear equations.

We conclude the technical contributions studying the problem of generating counterfactual instances for classifiers based on TPMs, motivated by the fact that BNs are the building blocks

of most standard approaches to perform this task. In this thesis we propose a novel algorithm that we prove is guaranteed to generate valid counterfactuals. The resulting algorithm takes advantage of the multilinear structure of TPMs, generalizing existing approaches, while also allowing for incorporating a priori constraints that should be respected by the final counterfactuals.

In the second part of this thesis we go beyond model transparency, looking into the role of explanations in achieving an effective collaboration between human users and AI. To study this we design a behavioural experiment where we show that explanations provide unique insights, which cannot be obtained by looking at more traditional uncertainty measures. The findings of this experiment provide evidence supporting the view that explanations and uncertainty estimates have complementary functions, advocating in favour of incorporating elements of both in order to promote a synergistic relationship between humans and AI.

Finally, building on our findings, in this thesis we design a course on explanations in AI, where we focus on both the technical details of state-of-the-art algorithms as well as the overarching goals, limitations, and methodological approaches in the field. This contribution aims at ensuring that users can make competent use of explanations, a need that has also been highlighted by recent large scale social initiatives. The resulting course was offered by the University of Edinburgh, at an MSc level, where student evaluations, as well as their performance, showcased the course's effectiveness in achieving its primary goals.

# Lay Summary

The adoption of automation systems has already demonstrated its potential to benefit society, reducing human labour and democratizing tools. However, it is quite common to have a well performing model, that is able to achieve a high accuracy, but is unable to provide any additional evidence that it bases its predictions on the appropriate criteria. This may pose serious challenges in high-stakes applications, as additional justification, other than accuracy, is needed in order to be confident that the model can be trusted as well as to guard against the adverse effects of potential model failures. For instance, when a model suggests a certain therapy for a patient, it is important to be able to provide some evidence that its decision-making process takes into account factors and interactions that are known to be important to human doctors, otherwise its questionable whether this prediction should be followed, due to the potentially life threatening associated risks.

These concerns have sparked an ongoing discussion on achieving a transparent relationship between models and human users. This includes making sure that models are able to explain their decision-making process, as well as that people are able to properly understand them. Unfortunately, most models cannot provide explanations due to their complex design, however some of them make assumptions that greatly facilitate extracting such information. Moreover, even when explanations are available, they are intended for a diverse audience, including data scientists, policy makers, doctors, and lay users which may have different backgrounds, so it is important to make sure that they satisfy the explanatory needs of all affected parties. Therefore, this thesis explores both technical and social aspects of transparency. We start with considering a recently introduced model class, where we show that it naturally supports multiple features that enable acquiring meaningful information out of it. Then we proceed to study the influence of various elements on the relationship between humans and AI, and finally we design a course to introduce people to model explanations.

# Acknowledgements

First of all, I would like to thank my supervisor, Vaishak Belle, for his support throughout my doctoral studies. Vaishak was has been nothing but considerate and willing to help with any challenge that came up during these years. Our collaboration has helped me develop both as a researcher and as a human being. Under his guidance I had plenty of room to grow, which allowed me to form my own ideas and be free to pursue any problem I found interesting. Moreover, each time I faced a personal struggle, he was more than supportive, always prioritizing my own well-being. I am deeply grateful for all of these, thank you for making this experience memorable.

I have also been fortunate enough to work with Peter Gostev, Matjaz Vidmar, and Kostas Kavoussanakis. Their advice has proved to be invaluable in helping me become effective in multidisciplinary teams, while they also showed me how academic results can be used in order to improve processes that affect the general public. Thank you for that, I will not forget it anytime soon.

I would also like to thank Andreas and Auste for all their hard work, which resulted in Chapter 8. Without their dedication and effort, this project would never come into existence. Moreover, I sincerely appreciate all the feedback that Subramanian Ramamoorthy and Christopher Lucas provided during my annual reviews, which helped me improve the quality of my work.

Of course, I cannot thank my family enough for supporting me during the entirety of my academic journey and for encouraging me to go on and pursue a PhD degree. If it was not for their assistance, my whole path in life would have been tremendously different.

Finally, I cannot put into words how in debt I am to my partner, Mara, for being in my life. She has been a constant source of inspiration, while her love has given me all the motivation I needed to overcome the times self-doubt tried to creep in. This thesis is a testament to how a loving environment can help us reach goals we would otherwise be unable to attain.

# Contents

# List of Figures

# List of Tables

# List of Algorithms

# Declaration

I declare that the thesis has been composed by myself and that the work has not be submitted for any other degree or professional qualification. I confirm that the work submitted is my own, except where jointly-authored work has been included. My contribution and those of the other authors to this work have been explicitly indicated below. I confirm that appropriate credit has been given within this thesis where reference has been made to the work of others.

The work presented in Chapter 8 has been the result of my collaboration with Andreas Bueff and Auste Simkute. I was responsible for writing the sections outlining the material of the course, generating the figures, and performing the quantitative analysis of our data. Andreas wrote most of the remaining sections, and developed the tutorials. Finally, Auste performed the qualitative part of the analysis.

*(Ioannis Papantonis)*

# Chapter 1

# Introduction

## 1.1 Preface

Artificial intelligence (AI) and especially machine learning (ML) has been increasingly incorporated into a wide range of critical applications, such as healthcare (Kononenko, 2001; Loftus et al., 2019), criminal justice (Chouldechova, 2017; Christin, 2017; Kleinberg et al., 2017), credit risk assessment (Chen et al., 2016b; Finlay, 2011), and loan approval (Wu et al., 2019). At the same time, automated systems have an effect on casual everyday decisions, by recommending news articles (Alvarado and Waern, 2018), movies (Bennett et al., 2007), and music (Mehrotra et al., 2018). At the core of this ML predominance lies the expectation that models can be more accurate than humans (Poursabzi-Sangdeh et al., 2021), something that has already been demonstrated in various tasks (Culverhouse et al., 2003; Goh et al., 2020; Hilder et al., 2009; Grove et al., 2000). Having said that, employing algorithms even as recommendation systems for cultural products (like movies) makes them part of the human culture, since they not only handle cultural products, but they also influence peoples' decisions and perceptions (Gillespie, 2016). This also means that they should not be viewed as mere tools (Bozdag, 2013), but rather as entities that hold their own values (Alvarado and Waern, 2018).

As such, it is paramount to make sure that their values align with those of humans, thus enabling ML's responsible integration to society (Russell et al., 2015; Gabriel, 2020; Christian, 2020). This need is further magnified by several recent instances of automated systems perpetuating undesired historical human biases, such as Amazon's recruitment algorithm exhibiting misogynistic behaviour (Meyer, 2018), or commercial tools utilized by the US criminal justice system being extremely biased against black defendants (Angwin et al., 2016; Dressel and Farid, 2018). Apart from that, ML failures can arise, for example, due to misuse, as in the case of an individual who spent an extra year in prison due to a typographical error in one of the inputs that was given to the ML system (Wexler, 2017). Of course, poor model design is another major source of catastrophic failures with far-reaching implications, such as putting people in danger due to inaccurate air quality assessment (McGough, 2018), or by

providing life-threatening cancer treatment recommendations (Strickland, 2019; Ross and Swetlitz, 2018). These and other similar pitfalls, along with the consequences and confusion that come with them (Galanos, 2019; Aleksander, 2017), has lead to some extreme arguments about ML potentially eroding the social fabric and even posing a threat to society's democratic foundation (Bozdag and Van Den Hoven, 2015).

In light of such concerns, there has been a surge of publications, increasing by about 100% every other year (Larsson et al., 2019), considering approaches that allow human users to get a more in-depth understanding of a model's decision making process. This encompasses many areas, including explainability in AI (XAI), fairness, and, responsibility (Arrieta et al., 2020; Mehrabi et al., 2021; Ehsan et al., 2022), which have now come to incorporate numerous perspectives from law regulation and philosophical ethics. When considering explainability, we can contrast approaches to extract *explanations in a post-hoc* manner to approaches that are *transparent by design*, such as when employing models that allow for readily inspecting their inner workings (Linardatos et al., 2020), like rule-based classifiers. These are seemingly complementary research directions, however both have the end-goal of conveying information regarding a model's decision-making process. Post-hoc methods come with the advantage of being applicable to virtually any model, albeit there is the potential risk of the generated explanations being inaccurate, for example due to some underlying approximations. On the other hand, transparent by design approaches provide insights that faithfully capture a model's reasoning, however this usually comes at the expense of sacrificing some of the accuracy.

Both of these directions can lead to significant advancements, and deciding which to use depends on the application at hand as well as the stakeholders that are going to interact with the resulting explanations. For example, while there is a growing body of academic work highlighting the benefits of employing transparent models in high-stakes applications (Rudin, 2019; Rudin et al., 2022; Afnan et al., 2021; Katsikopoulos et al., 2022; Quinn et al., 2022; von Eschenbach, 2021), there is also evidence that XAI generated explanations can facilitate the integration of AI in an array of critical domains, including finance (Bracke et al., 2019; Bussmann et al., 2021; Ariza-Garzón et al., 2020) and healthcare (Pawar et al., 2020; Antoniadi et al., 2021; Payrovnaziri et al., 2020). These results have motivated a series of works that empirically study the factors that influence the relationship between human users and automated systems (Lai et al., 2020; Lai and Tan, 2019; Poursabzi-Sangdeh et al., 2021; Chu et al., 2020; Carton et al., 2020).

This thesis is motivated by developments on both model transparency and XAI. The majority of the thesis (Chapters 3, 4, 5) is devoted on studying tractable probabilistic models (TPMs) and establishing properties that equip them with a series of transparent characteristics. TPMs provide a high degree of flexibility, allowing for incorporating context-specific information, as well as for efficiently representing distributions that are computationally intractable by more traditional graphical models (Darwiche, 2003). However, although TPMs are closely connected to Bayesian networks (BNs), which are considered to be one of the most transparent model classes (Arrieta et al., 2020), their complex inner representation makes it very challenging to

extract meaningful information out of them, as opposed to BNs, which can be easily inspected by human users. To address this issue, our research has focused on the question of whether it is feasible to extend certain transparent features of BNs to TPMs. In particular, we provide the following contributions:

- We provide novel transformation algorithms that can translate certain classes of TPMs into transparent graphical models, thus making their internal representations easy to inspect.
- We prove the theoretical feasibility of incorporating a priori probabilistic constraints into TPMs, similarly to Bayesian networks.
- We propose a way to generate counterfactual instances for classifiers based on TPMs, by taking into account their multilinear structure.

Following these results, the latter part of this thesis (Chapters 7, 8) looks into the broader landscape of transparency, which goes beyond obtaining algorithmic explanations and into identifying their role in driving the safe and responsible integration of AI into socially impacting applications (Principles, 2017). While the former part constitutes of technical contributions and is the primary focus of the thesis, this part considers social aspects of transparency, namely *user competence* and *human/AI collaboration*. This investigation was motivated by the understanding that in order to successfully integrate AI in social applications, it is necessary to study the effect of explanations on users' behaviour so we can develop informed practices that facilitate the formation of a high performance, transparent relationship between humans and AI (Ehsan et al., 2022). Following this, our research has focused on ways to ensure proper use and understanding of explanations, as well as studying their effect on the human/AI collaborative performance. In particular, we provide the following contributions:

- We design a behavioural experiment that explores how the interaction of human confidence, model confidence, and the degree of model assistance influence a user's behaviour. The final results suggest that explanations can provide unique insights that cannot be obtained be uncertainty estimates, as well as that the users' confidence in their capacity to perform a task significantly influences their reliance, understanding, and trust towards a model.
- We design a course introducing XAI and transparent by design approaches. Once both of them are introduced and contrasted with each other, the course focuses on recent developments in XAI, detailing its scope, goals, and limitations, while also introducing some state-of-the-art techniques to educate affected parties. Students' evaluations, as well as their performance showcased the course's effectiveness in familiarizing them with the field.

## 1.2 Transparency and TPMs

TPMs are a class of probabilistic models that extend traditional graphical models, such as BNs, in the sense that every directed/undirected graphical model can be represented as a TPM, while also potentially leading to exponential savings in both inference time and storing space (Darwiche, 2003). Although, this is a significant development that can be very useful in high-dimensional problems, it comes at the cost of TPMs losing all of the transparency-related properties that turned graphical models into an indispensable tool for scientific discovery during the last decades. For example, looking at BNs, all variable relationships are represented on a directed (usually acyclic) graph (Darwiche, 2009). This has the very appealing advantage of clearly expressing dependencies in the data, by only drawing arrows between variables. Furthermore, once the BN is specified, graphical tests can accurately recover all conditional independencies, without the need to perform any algebraic manipulations (Geiger et al., 1990). Due to these properties BNs are considered one of the most transparent model classes, since their internal representations can be easily inspected, by construction. It is this strength that has turned BNs into the backbone of causal inference, too; causal relationships are represented through a BN, while graphical criteria identify which causal effects can be estimated using observational data (Pearl, 2009b). Naturally, BNs have found numerous applications in integral applications (Kalet et al., 2015; Castelletti and Soncini-Sessa, 2007; Shenton et al., 2014; Uusitalo, 2007; Stewart-Koster et al., 2010; Friis-Hansen, 2000).

In addition to the above, BNs allow for incorporating various forms of a priori constraints, like temporal (Dechter et al., 1991) and probabilistic (Darwiche, 2009) ones, which can be encoded during model design, by directly adjusting the topology of the directed graph. The combination of all these properties imparts a high degree of flexibility, and offers a powerful alternative to black-box models, especially when considering high-stakes applications (Rudin, 2019; Rudin et al., 2022).

However, a downside is that inference in BNs is intractable, in the sense that computing marginal probabilities is NP-hard (Cooper, 1990). On top of that, specialized routines are required to perform the inferential step. This is the main motivation behind the recent emergence of TPMs, as an alternative approach that generalizes traditional BNs. TPMs directly encode the joint distribution of a set of variables, in a way that allows for a simple mechanism for performing inference, while potentially leading to reductions in both space and time complexity. Consequently, they have gathered significant attention in many applications (Wang and Wang, 2016; Rathke et al., 2017; Amer and Todorovic, 2015; Stelzner et al., 2019; Zheng and Pronobis, 2019).

Having said that, TPMs do not allow for gaining the same kind of in-depth insights about their internal representations, such as inspecting the dependencies between the various variables. Furthermore, constraint incorporation in TPMs has been very challenging, and apart from some simple cases, it is still unclear whether they have the capacity to perform this task. These

challenges have effectively turned TPMs into black-box models, despite them being motivated by and closely related to BNs. Therefore, our research addresses this issue by further exploring the link between TPMs and transparent graphical models, with the goal of moving TPMs out of the black-box territory. More specifically, the research questions we pursue are:

- **Is it possible to transform TPMs into non-trivial BNs?** It is well known that every BN can be transformed into an equivalent TPM (Darwiche, 2003). However, existing approaches for achieving the inverse transformation result in uninformative BNs (Zhao et al., 2015; Peharz et al., 2016; Butz et al., 2020). This is problematic since although a TPM might contain, for example, causal information, it is not possible to recover it, so it is essentially treated as a black-box model, but this is only due to the limitations of existing transformations. In Chapter 3, we formally prove this limitation, while we also examine sufficient conditions that can result in exact inverse transformations without any loss of information. Furthermore, we consider a certain class of TPMs and introduce a novel connection with chain graphs (Buntine, 1995). This results in a directed graph over an augmented set of variables which precisely captures the internal representation of models in the class, while also allowing for expanding their semantics to incorporate notions of causal inference.

- **Is it possible to incorporate probabilistic constraints to TPMs?** Probabilistic constraints into BNs can be easily integrated by manipulating their topology. On top of that, graphical criteria can be used to recover all conditional independencies that are implied by a certain BN (Geiger et al., 1990). Contrary to that, the problem of extending the same operations to TPMs has received very little attention, owing to their complex structure that inhibits a clear understanding of the relationships between variables. In Chapter 4, we consider the problem of incorporating probabilistic constraints into TPMs, looking into both conditional and unconditional independence constraints, as well as a certain kind of causal ones. In pursuing this question, we were able to provide formal proofs that constraint incorporation to TPMs corresponds to satisfying a system of multilinear equations, thus showing that it is indeed possible to enforce such constraints through parameter adjustments.

- **Is it possible to generate counterfactual instances for TPMs?** Counterfactual reasoning, i.e. the ability to imagine how something would have been, given what actually happened, has been a topic studied within philosophy, psychology and AI (Roese, 1997; Menzies and Beebee, 2001; Wachter et al., 2018). The ability to infer counterfactual probabilities by explicitly stating the causal dynamics that govern a set of variables and appropriately adjusting an underlying BN, has been a major breakthrough in AI, in particular (Pearl, 2009b). A related problem that has gained a lot of popularity during the last years is whether given a classification model and a datapoint it is possible to generate a new (hypothetical) datapoint that is as close as possible to the original one, but is classified differently by the model (Russell, 2019; Mothilal et al., 2020; Sokol and

Flach, 2019). The close relationship between BNs and TPMs, inspired us to explore if it is possible generate counterfactuals using TPMs, by taking advantage of their specific topology. The outcome of this endeavour is presented in Chapter 5, where we introduce a novel algorithm that is guaranteed to generate valid counterfactuals. Furthermore, the developed framework allows for constraining the final value of any set of variables, while it is applicable to the whole class of multilinear models, which includes rule-based classifiers and random forests. Finally, depending on the underlying model it is possible to generate infinite sets of counterfactuals, contrary to the majority of existing algorithms.

## 1.3 Transparency and Explanations

Following the main technical contributions, the final chapters focus on contextualizing XAI and model transparency in the broader endeavour of achieving the responsible integration of AI in social applications. Both of the aforementioned approaches aim at conveying information about a model's decision-making process to human users (either by designing models in certain ways or by developing post-hoc tools), so a natural next step would be to engage with parties that either use or are affected by AI in order to ensure that explanations are properly used, as well as to study their effect on the formation of the human-AI relationship. In fact, there seems to be a strong link between these two desiderata, supported by evidence suggesting that users' understanding and competence have a great influence on their willingness to employ an automated system (Balfe et al., 2018; Sheridan and Telerobotics, 1992; Merritt and Ilgen, 2008), meaning that both of these factors may have direct implications on the adoption of AI in practical applications (Linegang et al., 2006; Wright et al., 2019).

There is already a considerable body of work that explores the effectiveness of explanations on improving human accuracy (Mahbooba et al., 2021; Guo, 2020; Gunning et al., 2019; Gunning and Aha, 2019). This is a fairly reasonable approach, built on the premise that by explaining the reasons behind a model's prediction, human users can judge if the reasoning is sound, allowing them to reach an informed decision regarding whether to follow the corresponding prediction. Despite that, findings about the utility of explanations in making a model's internal reasoning clear to users are still ambiguous (Lai et al., 2020; Lai and Tan, 2019; Poursabzi-Sangdeh et al., 2021; Chu et al., 2020; Carton et al., 2020). This has raised concerns about the way humans perceive model explanations overall, calling for additional surveys to shed some light on this topic (Doshi-Velez and Kim, 2017; Vaughan and Wallach, 2020). Moreover, there is also alarming evidence suggesting that practitioners utilize XAI techniques in a wrong way (Kaur et al., 2020), where the misuse may arise both due to an incomplete technical understanding of XAI, as well as due to misunderstandings regarding XAI's intended use.

In addition to the above, users' own self-confidence in their ability to perform a task, is another factor that has an influence both on the human-AI relationship (Zhang et al., 2020; Bansal

et al., 2021b), and on the way users interact with explanations (Suresh et al., 2021; Liao and Varshney, 2021). Despite that, little is known about the specific role of self-confidence in the context of decision-making tasks, so it is still unclear how it might affect key metrics, such as accuracy, reliance, understanding, and trust towards a model. Motivated by the aforementioned challenges, our research provides for an attempt towards promoting the proper use of XAI, while also studying the effect of explanations on the relationship between human users and AI. More specifically, we target research questions related to:

- **Human-AI collaboration.** One of the ultimate goals of XAI is to facilitate building an effective relationship between users and AI, however, there is still no general consensus about the specifics of its influence. The majority of existing studies explore the capacity of explanations to improve the combined human-model predictive accuracy, however, there is evidence suggesting that uncertainty estimates might be more effective in inducing this effect (Poursabzi-Sangdeh et al., 2021; Chu et al., 2020; Carton et al., 2020). Having said that, this approach looks at uncertainty estimates and explanations as competing notions, missing out on any potential, non accuracy-related, benefits that might be gained by combining the two together. Furthermore, while there is evidence suggesting that a user's own self-confidence influences attitude towards explanations (Suresh et al., 2021; Liao and Varshney, 2021), the magnitude of this effect in the context of decision-making tasks is still unclear, in contrast to other settings, where more in-depth insights are available (De Vries et al., 2003; Yang et al., 2017). In Chapter 7, we present a behavioural experiment to close this gap, studying the impact of user confidence on the combined human-model accuracy, as well as for assessing the effect of combining explanations and uncertainty estimates. Our results indicate that explanations offer unique insights and significantly improve model understanding, compared to uncertainty estimates. Moreover, we provide evidence that self-confidence has a significant influence on a user's reliance, understanding, and trust towards a model.

- **Education in XAI.** As mentioned earlier, practitioners face various kinds of challenges when applying XAI techniques, most of them stemming from their incomplete understanding of the field. A way to address this issue would be to offer the affected parties sufficient education to help them understand and appropriately apply the right techniques. However, there is a stark lack of academic resources on XAI, such as university level courses, while online articles discussing related things, do not provide for a holistic, systematic approach. In fact, there is only a single academic course on XAI, offered by Harvard University (Lakkaraju and Lage, 2019), as well as some tutorials (Samek and Montavon, 2020; Camburu and Akata, 2021), but they are usually intended for researchers. In Chapter 8, we present a course designed for introducing students and professional to the field of XAI. This work served as a guideline for implementing and delivering an actual MSc course, offered by the University of Edinburgh, putting our proposals into action. Finally, upon completion, students provided us with very positive feedback.

## 1.4 Thesis Structure

In Chapter 2 we provide a concise introduction to sum product networks (SPNs), and discuss the previous work relevant to Chapters 3, 4, 5. In Chapter 6 we outline the XAI techniques and previous work relevant to Chapters 7, 8. The following chapters are based on the accompanying publications or submitted manuscripts:

- Chapter 1:

  I. Papantonis and V. Belle, "Transparency: Why do we care?". In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence*, 2023.

- Chapter 3:

  I. Papantonis and V. Belle, "Interventions and counterfactuals in tractable probabilistic models", in *Knowledge Representation  Reasoning Meets Machine Learning, Workshop at NeurIPS*, 2019.

  I. Papantonis and V. Belle, "Transparency in Sum-Product Network Decompilations", 2022. (under submission)

- Chapter 4:

  I. Papantonis and V. Belle, "Closed-form results for prior constraints in sum-product networks", *Frontiers in Artificial Intelligence*, 4, 2021.

- Chapter 5:

  I. Papantonis and V. Belle, "Principled diverse counterfactuals in multilinear models", in *Workshop on Explainable AI in Finance, ICAIF*, 2021.

- Chapter 7:

  I. Papantonis and V. Belle, "Why not both? Self-Confidence and the Complementary Effect of Uncertainty and Explanations in Decision-Making Tasks", 2023. (under submission)

- Chapter 8:

  V. Belle and I. Papantonis, "Principles and practice of explainable machine learning", *Frontiers in Big Data*, 39, 2021.

  A. Bueff, I. Papantonis, A. Simkute, and V. Belle, "Explainability in machine learning: a pedagogical perspective", *Arxiv preprint*, 2022.

Finally, in Chapter 9 we conclude this thesis and discuss future directions.

# Part I

# *Transparency and TPMs*

# Chapter 2

# Related work - Tractable Probabilistic Models

In this chapter we are going to introduce sum-product networks (SPNs), as well as review the literature that is going to be relevant to Chapters 3, 4, 5. Moreover, part of the results in Chapter 3 concern probabilistic sentential decision diagrams (PSDDs), which is another class of TPMs, however since they are relevant to that chapter only, we introduce them in Chapter 3. Furthermore, to facilitate the presentation, the related work is divided in two chapters, 2 and 6, where the former covers tractable models, and the latter focuses on explanations and social aspects of AI. We believe that this division will bring the relevant material closer to where they are indeed needed, making for a better overall presentation.

## 2.1 Sum Product Networks

One of the key limitations of traditional Bayesian networks is that inference may be exponential, meaning that often approximate inference techniques must be employed, which require additional computational resources, and may potentially lead to inaccurate results. Furthermore, another consequence is that since learning the parameters of a BN involves inference as an intermediate step, learning might be intractable as well. These challenges have led to the development of tractable probabilistic models, as an alternative way to encode a distribution.

The main idea was introduced in (Darwiche, 2003), where the author considers the problem of representing the distribution of a set of discrete variables as a polynomial. For example, if a binary variable $X$ follows a Bernoulli distribution with probability of success equal to $p$, then the polynomial $F(\mathbf{1}_{X=1}, \mathbf{1}_{X=0}) = p \cdot \mathbf{1}_{X=1} + (1-p) \cdot \mathbf{1}_{X=0}$, where $\mathbf{1}_{X=1}, \mathbf{1}_{X=0}$ are indicators for the events $X = 1, X = 0$, respectively, computes the probability of every event by setting the corresponding variables equal to their values:

- $\Pr(X = 1) = F(1, 0) = p$
- $\Pr(X = 0) = F(0, 1) = 1 - p$

It is not very hard to see that any discrete distribution can be rewritten as a polynomial by explicitly representing every state. For example, the joint distribution of two binary variables, $X_1, X_2$ can be written as:
$F(\mathbf{1}_{X_1=1}, \mathbf{1}_{X_1=0}, \mathbf{1}_{X_2=1}, \mathbf{1}_{X_2=0}) = \Pr(X_1 = 1, X_2 = 1)\mathbf{1}_{X_1=1} \cdot \mathbf{1}_{X_2=1} + \Pr(X_1 = 0, X_2 = 1)\mathbf{1}_{X_1=0} \cdot \mathbf{1}_{X_2=1} + \Pr(X_1 = 1, X_2 = 0)\mathbf{1}_{X_1=1} \cdot \mathbf{1}_{X_2=0} + \Pr(X_1 = 0, X_2 = 0)\mathbf{1}_{X_1=0} \cdot \mathbf{1}_{X_2=0}$.

More generally, let $\mathbf{X}$ be a set of discrete variables, $\mathbf{x}$ represent a state of the variables in $\mathbf{X}$, and $\Phi(\mathbf{x})$ the probability of state $\mathbf{x}$. The polynomial $\sum_{\mathbf{x}} \Phi(\mathbf{x}) \prod(\mathbf{x})$, where $\prod(\mathbf{x})$ is the product of the indicators corresponding to state $\mathbf{x}$[1], is called the *canonical network polynomial* of the distribution. Having this polynomial comes with some very appealing advantages regarding inference:

- Marginalizing a variable requires a single computation of this polynomial.
- Conditioning on an event requires computing the network polynomial twice.

This means that exact inference involves at most 2 computations of the network polynomial, instead of invoking expensive subroutines. Of course, on the other hand, this polynomial is exponential in the number of variables, so it is itself expensive to compute. Having said that, the idea behind sum-product networks (SPNs) is to identify ways to express the network polynomial in a more compact form, i.e. deriving a *factorized network polynomial*, since once computing it becomes efficient, exact inference becomes efficient as well.

More specifically, SPNs are rooted directed graphs that explicitly represent the computations in a network polynomial. A SPN $S$ over variables $\mathbf{X}$ is made of alternating layers of sum and product nodes, with all leaf nodes corresponding to univariate distributions. Any edge exiting a sum node has a non-negative weight assigned to it. The value of a product node is the product of its children, while the value of a sum node is a weighted sum of its children, $\sum_{u_j \in Ch(u_i)} w_{ij} S_j(\mathbf{x})$, where $Ch(u_i)$ is the set containing the children of node $u_i$, and $S_j$ is the sub-SPN rooted at node $u_j$. We can define an SPN, as follows:

- Any tractable univariate distribution is a SPN.
- The product of two SPNs with disjoint set of variables is also a SPN.
- The weighted sum of two SPNs with the same set of variables is a SPN.

SPNs can greatly reduce the time and space complexity of inference, by taking advantage of context-specific independence relationships or by reusing intermediate computations (Poon and

[1] For example, if $\mathbf{x} = (X_1 = 0, X_2 = 1)$, then $\prod(\mathbf{x}) = \mathbf{1}_{X_1=0} \cdot \mathbf{1}_{X_2=1}$

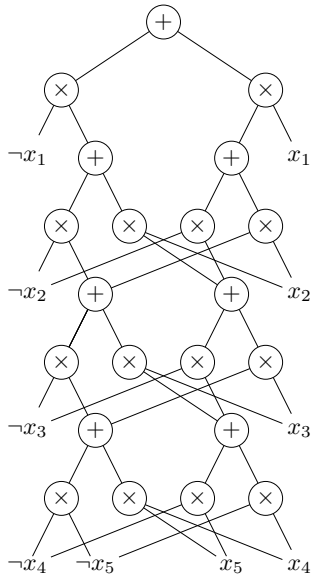


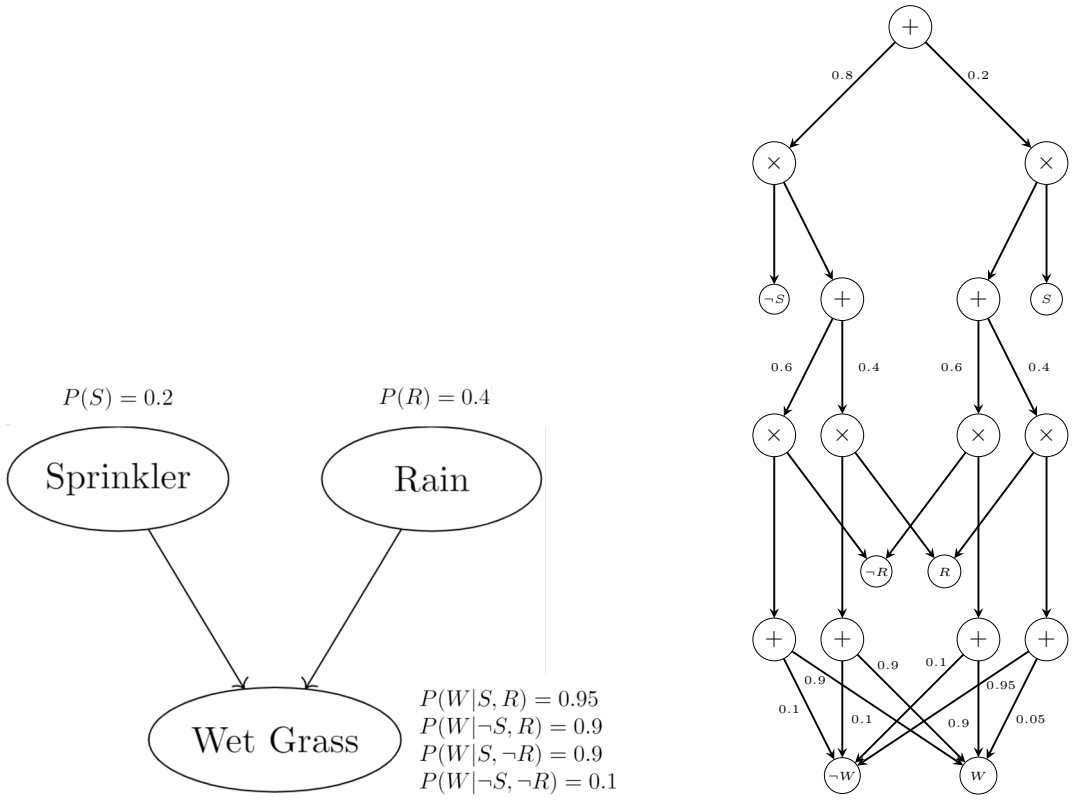

Figure 2.1: A SPN representing a uniform distribution over the states of five binary variables with an even number of 1's, adapted from (Poon and Domingos, 2011). For simplicity, weights are omitted from the SPN.

Domingos, 2011). As a matter of fact, SPNs are strictly more tractable as BNs (Zhao et al., 2015), meaning that any BN with tabular conditional probability tables (CPTs) can be transformed to a SPN in polynomial time, but not the opposite. Furthermore, there are certain distributions that can be encoded as compact polynomial sized SPNs, that cannot be compactly represented as BNs. For example (Poon and Domingos, 2011), show that a uniform distribution over the states of five binary variables with an even number of 1's requires an exponential number of components, when represented as a BN (since any component corresponds to a complete state), while a SPN achieves the same task in size linear in the number of variables. The reason behind this massive improvement is that SPNs allow for reusing intermediate computations, in contrast to BNs, as seen in Figure 2.1, where we denote the events $X_i = 1, X_i = 0$ with $X_i, \neg X_i$, respectively, to make the presentation more compact.[2]

Arguably, one of the most popular algorithms for compiling a BN into a SPN is the *variable elimination reverse topological ordering* (VErto) one presented in (Darwiche, 2003), which is based on variable elimination. The procedure starts by first specifying a topological ordering of the nodes in the BN[3], and then iteratively eliminating each variable from the CPTs it appears, resulting in a SPN representing the same distribution as the initial BN. For example, Figure 2.2a shows the well-known example of a BN representing the relationship between the events of raining ($R$), a sprinkler being on ($S$), and the grass being wet ($W$) (Alpaydin, 2020). According to this BN, both the weather conditions (raining or not) and the status of the sprinkler (on/off) have an influence on whether the grass is wet, while the corresponding probabilities indicate the likelihood of each event. In order to compile it to a SPN, the first step

[2]We are going to use this convention throughout Chapters 3, 4, 5.

[3]A topological ordering of the variables $X_1, X_2, \cdots, X_n$ in a BN is a permutation $X_{\sigma(1)}, X_{\sigma(2)}, \cdots, X_{\sigma(n)}$, such that for any $k = 1, 2, \cdots, n$ all the parents of $X_{\sigma(k)}$ appear within $X_{\sigma(1)}, \cdots, X_{\sigma(k-1)}$.




(a) A BN

(b) The corresponding SPN

Figure 2.2: A BN and the corresponding SPN that results from the compilation algorithm in (Darwiche, 2003).

is to specify a topological ordering, such as $S, R, W$ (alternatively, $R, S, W$ is another valid ordering). Eliminating variables in this order, the resulting SPN is shown in Figure 2.2b. Both the BN and SPN represent the same distribution, while when traversing the SPN from root to leaf, every path meets the variables in the way specified by the topological ordering used during the compilation process.

A central notion in SPNs is that of the *scope* of a node, which refers to the set of variables over which the node is defined. Intuitively, each node computes a quantity that depends only on the variables in its scope, and changes in the remaining ones do not affect its output. More formally, the scope of a node $n$ is defined as:

$$Scope(n) = \begin{cases} \{X\}, & \text{if } n \text{ is an indicator variable } \mathbf{1}_{X=0} \text{ or } \mathbf{1}_{X=1} \\ \cup_{c \in Ch(N)} Scope(c), & \text{otherwise} \end{cases}$$

where $Ch(n)$ denotes the children of a node, i.e. the set of all nodes that are connected to $n$ with edges exiting from $n$.

At this point, it should be noted that not every SPN represents a valid distribution, since it is possible to construct SPNs corresponding to network polynomials with missing terms (Poon and Domingos, 2011). However, *completeness* and *decomposability* are two properties that guarantee that a SPN indeed represents a valid distribution:

- A sum node, $n$, is complete if all of its children have the same scope, meaning that $Scope(j) = Scope(k)$ for all $j, k \in Ch(n)$. In this case, the outcome of the sub-SPN rooted at node $n$ is $S_n(\cdot) = \sum_{j \in Ch(n)} w_{nj} S_j(\cdot)$, where $\sum_j w_{nj} = 1$. Intuitively, this property ensures that if all of the nodes in $Ch(n)$ are valid distributions, then node $n$ is a valid distribution, too, since a convex combination of distributions is a distribution itself.
- A product node, $n$, is decomposable if all of its children have disjoint scopes, meaning that $Scope(j) \cap Scope(k) = \emptyset$ for all $j, k \in Ch(n)$. In this case, there is a partition of $Scope(n) = \{\mathbf{X}_1, \cdots, \mathbf{X}_{|Ch(n)|}\}$ such that we have that $S_n(\mathbf{X}_1 \cup \cdots \cup \mathbf{X}_{|Ch(n)|}) = \prod_{j \in Ch(n)} S_j(\mathbf{X}_j)$. Intuitively, this property utilizes the fact that the product of distributions over disjoint sets is a valid distribution, so if all of the children of a product node are valid distributions, then this node is a valid distribution itself.

Finally, *selectivity* is another useful property, since it allows SPNs to compute the *most probable explanation* with a single computation. A sum node $n$ is selective if for any assignment $\mathbf{x}$, only one of its children has a non-zero output. We conclude this section by noticing that SPNs resulting from the VErto algorithm satisfy all these properties, since they are complete, decomposable, and selective (Darwiche, 2003).

### 2.1.1 Graphical transformations of SPNs

As already discussed, every BN can be represented as a SPN, so a natural question is whether the inverse is also true. This was initially studied in (Zhao et al., 2015), where the authors propose an algorithm to achieve this goal, by interpreting sum nodes as latent random variables. Given a SPN defined over variables $X_1, X_2, \cdots, X_k$, the exact procedure is as follows:

- Introduce a set of latent variables, $Z_1, Z_2, \cdots, Z_n$, one for each sum node in the SPN.
- Create an empty BN over the original observable variables, $X_1, X_2, \cdots, X_k$, and the new latent variables, $Z_1, Z_2, \cdots, Z_n$.
- Draw edges from each $Z_i$ to the original variables belonging in the scope of the sum node for which $Z_i$ was created for.

A similar approach was proposed subsequently in (Peharz et al., 2016), where the difference is that the authors take into account the hierarchical structure of SPNs, so instead of only

connecting each latent variable with the original variables in its scope, they allow for connections between latent variables, $Z_i \rightarrow Z_j$, as long as the sum node corresponding to $Z_j$ is in the scope of the one corresponding to $Z_i$. More recently, a new transformation was proposed in (Butz et al., 2020), where this time the authors make the extra assumption that the SPN under examination is compiled from a BN (using the VErto algorithm). Furthermore, the authors argue that when some latent variables have the same children, it is possible to collapse them into a single latent variable. Figure 3.2, in Chapter 3 gives an example of the differences between these approaches.

An advantage of being able to transform SPNs back to BNs (apart from making SPNs transparent) is that it opens the door for studying causal queries using the SPN. In fact, performing causal inference using SPNs has been the topic of an emerging line of research. In (Zečević et al., 2021a), the authors propose a way for utilizing SPNs in order to estimate *interventional distributions*[4], using the SPN as a functional mapping between BN structures and distributions. Furthermore, in (Zečević et al., 2021b), the authors show that SPNs can be used to represent both observational and interventional distributions, although each requires a dedicated SPN. Finally, in (Darwiche, 2022), a different approach is presented, where a BN is compiled into a SPN, and then both the original BN and the resulting SPN are used in order to compute either causal or non-causal queries.

### 2.1.2 Constraints in SPNs

Although the problem of constraint satisfaction has been studied extensively in BNs, little is known about the plausibility of incorporating independence constraints in SPNs. To the best of our knowledge, the only relevant result is presented in (Peharz et al., 2014), where the authors discuss how in order to represent the structure $X_1 \rightarrow X_3 \leftarrow X_2$ as a SPN, which implies that $X_1 \perp X_2$, it is necessary to enforce equality between certain parameters (as shown in Figure 4.1, in Chapter 4). However, this observation is only applicable to this rather simple case, while it also requires the underlying BN structure to be known a priori. This leaves open the question of what happens in the general case, for example when the underlying structure is not known or when the SPN contains context-specific relationships.

On the other hand, it is well-known that it possible to enforce probabilistic independence constraints to BNs by adjusting either their structure or their parameters (Meek, 1995), leading to multiple works that take advantage of this property. For example, the authors in (De Campos et al., 2009) address the problem of structure learning in BNs in the presence of probabilistic constraints reflecting expert knowledge. Furthermore, in (Feelders and Van der Gaag, 2006), the authors propose a way to learn the parameters of a BN by taking into account qualitative constraints describing relationships between variables. Ancestral constraints, enforcing parent-child relationships, have also been explored, for example in (Chen et al., 2016a). An

[4]More details about such distributions in Chapter 3

alternative line of research, can be found in (Dechter et al., 1991; Dechter, 1999), where the authors consider the problem of incorporating temporal constraints.

### 2.1.3 Counterfactuals in SPNs

Counterfactuals have a long standing history within philosophy (Lewis, 1974; Ruben, 1990), as well as within the causal modelling community (Pearl, 2009b). Furthermore, they have found many applications into XAI, where they have gained significant traction in recent years, partly because there is evidence suggesting that non-technical audience feels more comfortable interpreting such explanations over alternatives, such as propositional rules (Binns et al., 2018). Furthermore, counterfactuals inherently convey a notion of "closeness" to the actual world, in the sense that they allow for detecting a set of minimal changes that can alter a model's decision. When it comes to BNs, counterfactuals are usually studied using Pearl's *do-calculus* (Pearl, 2009b), which requires knowing the specific functional relationships between all variables in the model. This is a highly non-trivial task, often requiring human experts to hand-craft the final model. Due to this challenge, an alternative approach for generating counterfactuals was proposed in the seminal work of Wachter et al. (Wachter et al., 2018), based on Lagrange multipliers, assuming the classifier is differentiable. This method does not require such an in-depth model specification, while it is also compatible with BNs that exhibit context-specific independencies. In contrast, extending Pearl's framework to such BNs is still ongoing (Tikka et al., 2019).

Building on top of the results in (Wachter et al., 2018), Russel (Russell, 2019) proposes a modified framework, based on *mixed integer programming* (MIP), to generate counterfactuals for linear models. As the author notes, this approach resolves some technical issues of (Wachter et al., 2018), while it also provides a principled way for generating diverse counterfactuals, since utilizing only a single counterfactual can be overly restrictive, impeding a better model understanding. The MIP approach to generating counterfactuals has been explored in a series of additional works as well. In (Cui et al., 2015; Tolomei et al., 2017) the authors propose such a method, especially designed for tree ensemble models. The resulting optimization problems are guaranteed to output a counterfactual instance (or possibly an infinite set of counterfactuals), however they are only applicable to tree models, and they do not support incorporating a priori diversity constraints, i.e. constraints that force a set of features to take on certain values or satisfy inequality conditions. This is important for many applications, since it is often the case that certain features are immutable (their values cannot be changed), such as a person's height. Alternative MIP formulations can be found in (Kanamori et al., 2020, 2021), which are applicable to linear models, however it is again unclear how to incorporate a priori constraints, as well as whether it is possible to apply these methods to classifiers that are based on SPNs.

Apart from the aforementioned approaches, the problem of generating counterfactual instances has been considered from alternative angles as well. In a recent line of work (Shih et al., 2018;

Shi et al., 2020; Choi et al., 2020), a different framework for producing counterfactuals is presented, based on utilizing intermediate architectures, such as OBDDs (Bryant, 1992). At the core of these works lies the idea of compiling a classifier into a structure that supports counterfactual generation in polynomial time. An advantage of this approach is that such models usually support answering a number of different queries in polynomial time, not only counterfactuals. That being said, the compiled model can be exponentially larger than the original one, while at the same time this approach supports only a single distance function, thus posing limitations on the expressiveness and flexibility of the resulting counterfactuals.

# Chapter 3

# Graphical representations of TPMs

## 3.1 Introduction

In recent years, there has been an increasing interest in studying causality-related properties in machine learning models (Bau et al., 2018; Besserve et al., 2018). Broadly speaking (Pearl, 2019), the motivation stems from extending the query and reasoning capabilities over probabilistic domains. That is, in standard probabilistic models, one is simply interested in *conditioning on observations* $\Pr(y \mid x)$: e.g., what is the likelihood of being tall given that you play basketball? Causal reasoning allows us to reason about *interventions* $\Pr(y \mid do(x))$: e.g., what is the probability of a person being tall given that he/she is made to play basketball?. *Counterfactual queries* allow us to reason about alternate worlds $\Pr(y \mid x^*)$: e.g., how tall would a person be if he/she was playing football instead of basketball?

A fundamental challenge underlying stochastic models, however, is the intractability of inference (Cooper, 1990). This has led to the paradigm of *tractable probabilistic models* (TPMs), where conditional or marginal distributions can be computed in time linear in the size of the model. Although initially limited to low tree-width models (Bach and Jordan, 2002), recent TPMs such as sum product networks (SPNs) (Poon and Domingos, 2011; Gens and Domingos, 2013) and probabilistic sentential decision diagrams (PSDDs) (Kisa et al., 2014) are derived from arithmetic circuits (ACs) and knowledge compilation approaches, more generally (Darwiche, 2002; Choi and Darwiche, 2017), which exploit efficient function representations and also capture high tree-width models. These models can also be learnt from data (Gens and Domingos, 2013; Kisa et al., 2014), leveraging the efficiency of inference.

Both of these models are closely related to Bayesian networks (BNs), as any BN can be compiled into a SPN (Darwiche, 2003) or a PSDD (Shen et al., 2016). However, their internal representation makes it very challenging to identify relationships and dependencies among the variables, in contrast to BNs, where it is immediate to uncover all the conditional independencies within a set of variables. In other words, BNs are *transparent* models, while

SPNs and PSDDs act as *black-boxes*. Earlier works address this issue by developing de-compilation algorithms that transform TPMs back to BNs in order to make the underlying relationships clear (Zhao et al., 2015; Peharz et al., 2016; Butz et al., 2020), hoping that it would also enable TPMs to be used in causal inference applications. In fact, on studying the relationship between SPNs and BNs (Zhao et al., 2015), the authors conclude with:

> *The structure of the resulting BNs can be used to study probabilistic dependencies and causal relationships between the variables of the original SPNs.*

In this chapter, we begin by studying the causal utility of the BNs generated by the methods in (Zhao et al., 2015; Peharz et al., 2016; Butz et al., 2020). Following that, we study the problem of de-compiling SPNs and PSDDs into graphical models, in a way that moves them closer towards the spectrum of transparency. More precisely, we present the following contributions:

- We prove that existing SPN to BN de-compilations result in graphs that support only trivial causal queries, providing an answer to the claims in (Zhao et al., 2015).
- We propose a new de-compilation algorithm that leads to an exact SPN to BN de-compilation, for SPNs compiled from BNs using the VErto algorithm, thus enhancing the transparency of SPNs by accurately uncovering their internal representations.
- We establish a novel relationship between PSDDs and chain graphs (CGs) that both enhances the PSDDs' transparency, as wells as allow for equipping them with a form of causal semantics.

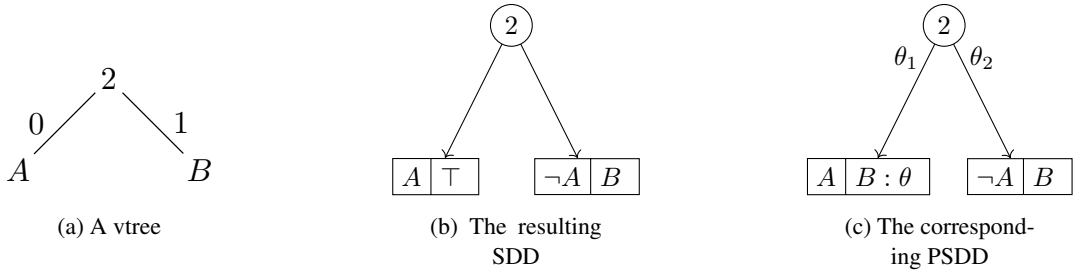
## 3.2 Background

### 3.2.1 PSDDs

Probabilistic sentential decision diagrams (PSDDs) are parameterized sentential decision diagrams (SDDs), which provide a language for representing propositional formulas in a way that allows for a series of useful quantities to be efficiently computed. The main idea is to represent a propositional formula in the form $(p_1 \wedge s_1) \vee \cdots \vee (p_k \wedge s_k)$, where the $p_i$'s are called primes, and the $s_i$'s are called subs. The primes are required to form a partition, while the variables appearing in the primes and the subs must not overlap. For example, a SDD representation of the expression $A \vee B$ is

$$A \vee B \equiv (A \wedge \top) \vee (\neg A \wedge B) \tag{3.1}$$

Clearly, the primes, $A, \neg A$ form a partition, while the primes and subs involve non-overlapping sets of variables.




(a) A vtree (b) The resulting SDD (c) The corresponding PSDD

Figure 3.1: The SDD and PSDD that are induced by formula the 3.1 and the vtree in 3.1a.

In order to determine which variables are going to appear within the primes and which within the subs, SDDs utilize a *variable tree* (vtree), which is a binary tree with leaves that are in one-to-one correspondence with the variables in the SDD. The variables that can be reached through the tree's left child appear in the primes, while those reached through the right one appear in the subs. For example, following the vtree in Figure 3.1a, the formula (3.1), results into the SDD in Figure 3.1b. *Decision nodes* are denoted with circles, and they compute the disjunction of their children. Moreover, the number within a decision node indicates the vtree node that determines the primes and the subs. For example, the SDD root contains the number 2, so (looking at the tree in Figure 3.1a) the primes should contain $A$, and the subs $B$, which is indeed the case. Finally, the *base*[1] of a SDD node, denoted with $[\cdot]$, corresponds to the formula it represents after removing terms that are tautologically true or false. In our example, the base of the root is $[(A \wedge \top) \vee (\neg A \wedge B)] = A \vee B$.

We are now ready to formally define SDDs, as in (Darwiche, 2011). $S$ is a SDD that respects vtree $v$ iff:

- $S = \top$ or $S = \bot$. The corresponding bases are $[\top] = True$ and $[\bot] = False$.
- $S = X$ or $S = \neg X$ and $v$ is a leaf with variable $X$. Base: $[X] = X$ and $[\neg X] = \neg X$
- $S = (p_1 \wedge s_1) \vee \cdots \vee (p_k \wedge s_k)$, $v$ is an internal vtree node, $p_1, \cdots, p_n$ are SDDs respecting $v$'s left child, $s_1, \cdots, s_n$ are SDDs respecting $v$'s right child, and $p_1, \cdots, p_n$ form a partition. Base: $[(p_1 \wedge s_1) \vee \cdots \vee (p_k \wedge s_k)] = \bigvee_{i=1,\ldots,k} [p_i] \wedge [s_i]$.

PSDDs build on top SDDs by parameterizing them so they induce a valid distribution, as follows: each prime $p_i$ is assigned a non-negative parameter $\theta_i$ such that $\sum_{i=1}^{k} \theta_i = 1$ and $\theta_i = 0$ if and only if $s_i = \bot$. Additionally each terminal node containing $\top$ is replaced by $X : \theta$, where $X$ is the variable in the corresponding vtree node, and $\theta$ is a parameter such that $0 < \theta < 1$. Using this notation, let $n$ be a PSDD node normalized for a vtree node $v$, then $n$ defines a distribution over the variables in $v$ as follows (Kisa et al., 2014):

- If $n$ is terminal node, and the corresponding vtree node, $v$, has a single variable $X$, then

[1] In the original paper, the term *semantics* is used, however, in the context of a PSDD the term "base" is more common.

| $n$ | $\Pr_n(X)$ | $\Pr_n(\neg X)$ |
|---|---|---|
| $X : \theta$ | $\theta$ | 1 - $\theta$ |
| $\perp$ | 0 | 0 |
| $X$ | 1 | 0 |
| $\neg X$ | 0 | 1 |

- If $n$ is a decision node and $v$ has left children $\mathbf{X}$ and right children $\mathbf{Y}$, then $\Pr_n(\mathbf{x}, \mathbf{y}) = \Pr_{p_i}(\mathbf{x}) \cdot \Pr_{s_i}(\mathbf{y}) \cdot \theta_i$, for the unique $i$ for which $\mathbf{x} \models p_i$. $\Pr_{p_i}(\cdot), \Pr_{s_i}(\cdot)$ denote the distribution of the PSDD nodes corresponding to the sub-PSDDs $p_i, s_i$, respectively.

Figure 3.1c, shows the PSDD corresponding to the SDD in Figure 3.1b. By construction, PSDDs enforce properties analogous to completeness and decomposability (Kisa et al., 2014), thus enabling tractable inference in the same manner as SPNs do. On top of that, PSDDs allow for integrating logical constraints in a distribution by taking advantage of the underlying SDD, so that only assignments satisfying the constraints have non-zero probability. This has found applications in domains where structural properties need to be enforced, such as when computing valid routes on a map (Shen et al., 2018).

### 3.2.2 Causal Inference

Traditionally, causal analysis has been based on structural equation models (SEMs) (Pearl, 2009a), which provide for an effective way to encode dependencies between variables, as well as for studying causal queries. Probabilistic relationships are represented using a BN, modeling the joint distribution, while the specific mechanism that governs the functional relationships between the various variables are given as a set of structural equations, $\{F_V : V \in \mathbf{V}\}$, where $\mathbf{V}$ is the set of the model's variables. A consequence of having access to these equations is that in order to define a probability distribution over the BN, it is sufficient to define it over the set of *exogenous* variables, $\mathbf{U}$, which contains all variables, $X$, with $\mathbf{PA}_x = \emptyset$, where $\mathbf{PA}_x$ denotes the set of $X$'s parents. This is because all the remaining variables are functions of those in $\mathbf{U}$.

Interventional distributions, i.e. the distribution of a set of variables, after a second set of variables is forced to attain certain values, are of central importance in causal inference. Assuming an intervention on a variable $X$, denoted by either $do(X = x)$ or $do(x)$, the joint distribution of the remaining variables, $\mathbf{V}_{-x}$, under this intervention, is $\Pr(\mathbf{V}_{-x}|do(X = x)) = \frac{\Pr(\mathbf{V}_{-x}, X=x)}{\Pr(X=x|\mathbf{PA}_x)}$. However, graphical criteria are extensively used in order to express interventional distributions in terms of conditionals and marginals (Pearl, 2009a). We introduce one of the most prominent such criterion in Section 3.6, as we utilize it in our proofs.

Another class of queries, that highlight the power of causal inference, are counterfactuals, expressions of the form $\Pr(Y = y|do(X = x), \mathbf{E} = \mathbf{e})$, meaning "Given that we observed

evidence **e**, in the factual world, what is the probability of $Y$ being equal to $y$, had $X$ been equal to $x$?". These statements can be handled using the following steps (Pearl, 2009b):

i **Abduction:** Update the distribution $\Pr(\mathbf{U})$ by incorporating the evidence, to obtain $\Pr(\mathbf{U}|\mathbf{e})$.

ii **Action:** Perform the intervention $do(X = x)$ and substitute $x$ into all equations involving $X$.

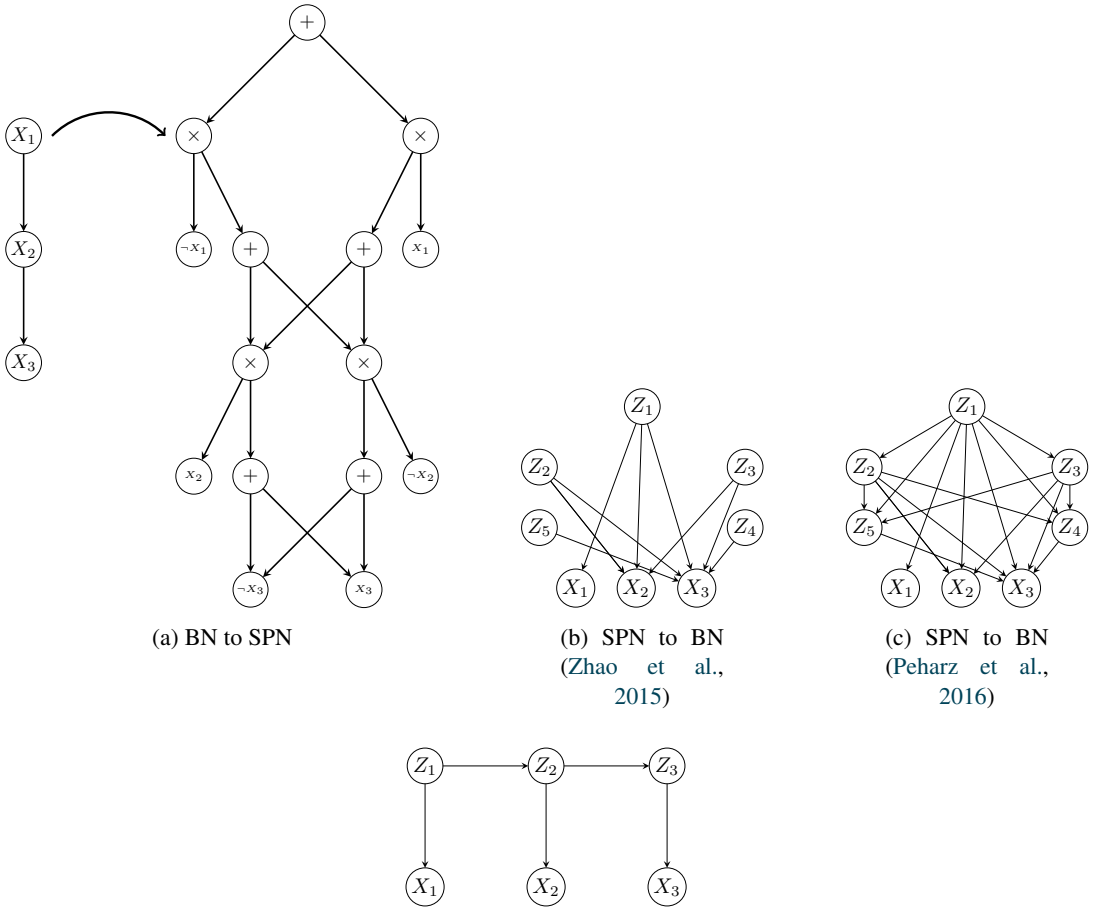
iii **Prediction:** Use the updated distribution, graph, and equations to compute the probability of $Y = y$.

Chain graphs (CGs) form another class of graphical models that has received considerable attention in causal analysis applications. While BNs contain only directed edges, CGs allow for undirected edges as well, resulting in graphical models capable of representing exponentially more distributions than BNs (Pena, 2007). The causal interpretation of these graphs was studied in (Lauritzen and Richardson, 2002), where the authors investigated ways to perform interventions in CGs. Having said that, there is no established general framework for studying counterfactuals in CGs, however the specific topology of the graphs studied in this chapter, allows for treating such queries in the same way as BNs.

## 3.3 Graphical representations of SPNs

### 3.3.1 Causal Utility of Existing Decompilations

All the existing SPN de-compilation algorithms, discussed in Chapter 2 (Zhao et al., 2015; Peharz et al., 2016; Butz et al., 2020), result in BNs that are defined over both the latent and the observable variables. Figures 3.2b, 3.2c, 3.2d, depict the result of applying each of these algorithms, respectively, to the SPN in Figure 3.2a, which was compiled from a simple chain BN with 3 variables. Looking at them it is clear that in terms of transparency the introduction of the latent variables leads to transformations that result in uninformative BNs that do not retain any of the original information in the initial BN in 3.2a. This situation seems rather problematic, since even when a SPN is compiled from a simple and informative BN, existing transformations might fail to recover it. Furthermore, looking at Figure 3.2, an intuitive observation is that if the initial BN represented causal relationships between $X_1, X_2, X_3$, the decompiled BNs attribute every correlation to latent factors. This means that even if we assume we start with a BN that represents a set of causal assumptions, compiling the BN to a SPN, and then decompiling it using any of the existing transformations, results in a complete loss of information. The following result formalizes this intuitive observation, by showing that the fact that there is no edge coming out of the observable variables implies that all interventional distributions are trivial.




(a) BN to SPN

(b) SPN to BN (Zhao et al., 2015)

(c) SPN to BN (Peharz et al., 2016)

(d) SPN to BN (Butz et al., 2020)

Figure 3.2: The final BNs that result from each transformation

**Proposition 3.1.** *Let $\mathcal{B}$ be a BN, $\boldsymbol{V}$ the set of its nodes, and $\boldsymbol{X} \subseteq \boldsymbol{V}$ such that no node in $\boldsymbol{X}$ has an edge coming out of it, then* $\Pr(\boldsymbol{V}_{-\boldsymbol{X}}|do(\boldsymbol{X})) = \Pr(\boldsymbol{V}_{-\boldsymbol{X}})$.

This immediately leads to the following:

**Theorem 3.2.** *The BN, $\mathcal{B}$, that results after transforming an SPN using the procedure described in (Zhao et al., 2015), (Peharz et al., 2016), or (Butz et al., 2020), satisfies the property* $\Pr(\boldsymbol{V}_{-\boldsymbol{X}}|do(\boldsymbol{X})) = \Pr(\boldsymbol{V}_{-\boldsymbol{X}})$*, for any $\boldsymbol{X} \subseteq \boldsymbol{V}$, where $\boldsymbol{V}$ is the set of the observable variables.*

The above result shows that existing interpretations do not result in BNs that can accommodate for interesting causal queries. This limitation stems from the fact that sum nodes are interpreted as latent variables, which leads to every probabilistic dependency been attributed to unobserved confounders, not to direct interaction between the variables. In turn, it is not surprising that all interventions are trivial, since the resulting BNs hold no meaningful information about the relationships between the variables.


(a) Children

(b) Grandchildren

Figure 3.3: Examples of indicator children and grandchildren

### 3.3.2 A New Decompilation Approach

The results in the previous section identify some of the implications of not being able to accurately decompile a SPN, in causality-related applications. However, even if a SPN represents purely associational, non-causal, relationships, the resulting BNs may fail to represent the dependencies between the variables, hindering the SPN's transparency. It is now reasonable to wonder whether this is due to an inherent limitation of SPNs, or an artifact of the existing transformations. One might argue that a SPN is merely a computational representation of the joint distribution of a BN, meaning that its purpose is to capture the functional characteristics of the joint distribution and provide an efficient way of computing it. Looking at SPNs from this angle, existing decompilation algorithms just reflect the various functions that SPNs define internally in order to perform the necessary computations. For example, Figure 3.2b represents the fact that the SPN in Figure 3.2a consists of 5 sub-SPNs; one defined over all $X_1, X_2, X_3$, two over $X_2, X_3$, and two over just $X_3$.

However, in the remaining of this section, we show that under certain conditions it is possible to decompile SPNs to far more informative BNs, that contain no artificial latent variables, rather they only include the original SPN variables. This involves utilizing alternative interpretations of the SPN sum nodes, as well as making use of the SPN parameters. This is a novel aspect of our approach, since none of the existing decompilation algorithms take into account the SPN parameters, although they contain information that cannot be retrieved by looking at the SPN structure alone (Peharz et al., 2014). More precisely, we assume that a BN $\mathcal{B}$ is an *I-map* for the distribution it represents (i.e. the independence relationships that can be inferred from the BN, are indeed satisfied by the distribution), and we consider the problem of decompiling the SPN that results from compiling $\mathcal{B}$ using the VErto algorithm, back into $\mathcal{B}$. This is the exact same setting as in (Butz et al., 2020), where the proposed algorithm can perform an exact decompilation in certain cases. Having said that, we propose an algorithm that *always* results in an exact decompiled BN.

As shown in Section 3.3.1, the assumption that every sum node corresponds to a latent variable is very restricting, having detrimental effects on the transparency and causal utility of the

decompiled BN. Although all existing decompilation algorithms share this assumption, this is not the only way of interpreting sum nodes. As a matter of fact, a meaningful probabilistic interpretation can be given to any sum node of a SPN that *represents* a variable (Paris et al., 2020). A sum node, $S$, represents a variable, $V$, if it has as many children as the number of states of $V$, and an indicator corresponding to one of $V$'s states can be reached either immediately (i.e., it is a child of $S$) or after one intermediate layer (i.e., it is a grandchild of $S$). For example, in both Figures 3.3a, 3.3b the root represents $X_1$, since it has as many children as $X_1$'s states, and each of them reaches a unique indicator either immediately (3.3a) or through a product node (3.3b). As shown in (Paris et al., 2020), if a sum node represents $V$, then it encodes the conditional distribution of $V$ given the context set by its ancestors. For example, in Figure 3.4b, the sum node in the red double circle (corresp. the blue double circle) models the distribution of $X_2|X_1 = 0$ (corresp. $X_2|X_1 = 1$). Analogously, in Figure 3.4c, the sum node in red models the distribution of $X_3|X_1 = 0, X_2 = 0$, since it represents $X_3$, while starting from the root, the path leading to this sum node contains indicators corresponding to the context $X_1 = 0, X_2 = 0$.

An important consequence of having a sum node representing a variable is that it is no longer necessary to introduce latent variables to define conditional distributions. Furthermore, we can now make two crucial observations about the qualitative properties of SPNs that result from the VErto algorithm. To this end, let $\mathcal{B}$ be a BN, and $\mathcal{SPN}_{\mathcal{C}}$ be the corresponding compiled SPN, then the following properties hold:

- **Property 1** Every sum node in $\mathcal{SPN}_{\mathcal{C}}$ represents a variable (Darwiche, 2003; Paris et al., 2020).
- **Property 2** Let $V$ be a variable in $\mathcal{B}$, and $\mathbf{S}$ be the set of sum nodes representing $V$ in the SPN, $\mathbf{S} = \{S|S \text{ is a sum node, Represent}(S) = V\}$, where for any sum node, $S$, Represent($S$) denotes the variable represented by $S$. Then, since $\mathcal{SPN}_{\mathcal{C}}$ is compiled using a topological ordering, each of the paths that start from the root and end in one of the nodes in $\mathbf{S}$ meets sum nodes representing the same variables. For example, looking at Figure 3.4b, $X_2$ is represented by the nodes in the red and blue double circles. Starting from the root, all paths leading two these nodes include just the root, which represents $X_1$. In addition, the double circles in Figures 3.4c, 3.4d correspond to all the nodes representing $X_3$. It is not difficult to see that all paths ending in one of these nodes, include the root (representing $X_1$) as well as one of the nodes that represent $X_2$. This is a direct consequence of having a topological ordering.

At this point we should remind ourselves that the goal of any decompilation algorithm is to recover the parent-child relationships represented by the underlying BN, $\mathcal{B}$. Having this in mind, properties 1 and 2 imply that for every variable, $V$, $\mathcal{SPN}_{\mathcal{C}}$ defines its conditional distribution by conditioning on all of its SPN ancestors, $X_1, \cdots, X_k$, i.e. the variables that appear earlier than $V$ in $\mathcal{SPN}_{\mathcal{C}}$, denoted by $\mathbf{C}_S$. This resembles the conditional distributions of

$V$ in $\mathcal{B}$, however this is defined by conditioning only on its parents, $P_1, \cdots, P_m$, denoted by $\mathbf{C}_B$, not all of its ancestors. Regardless, the crucial observation here is that since $\mathcal{SPN}_\mathcal{C}$ was compiled following a topological ordering respecting $\mathcal{B}$, then $\mathbf{C}_B \subseteq \mathbf{C}_S$ and furthermore all variables in $\mathbf{C}_S \setminus \mathbf{C}_B$ are non-descendants of $V$ in $\mathcal{B}$.

In general, distributions that factorize according to a BN respect the *local Markov property* (Peters et al., 2017), which states that a variable is independent of its non-descendants given its parents. In our case, since $\mathbf{C}_B$ contains exactly the parents of $V$ in $\mathcal{B}$, the aforementioned property makes sure that for any set, $\mathbf{ND}$, comprised of non-descendants of $V$ in $\mathcal{B}$, we have that $\Pr(V|\mathbf{C}_B) = \Pr(V|\mathbf{C}_B, \mathbf{ND})$. Connecting this with the observations in the previous paragraph, we immediately see that $\Pr(V|\mathbf{C}_S) = \Pr(V|\mathbf{C}_B)$, since $\mathbf{C}_B \subseteq \mathbf{C}_S$ and $\mathbf{C}_S \setminus \mathbf{C}_B \subseteq \mathbf{ND}$.

This remark implies that the task of decompiling $\mathcal{SPN}_\mathcal{C}$ is equivalent to inferring which of the variables in $\mathbf{C}_S$ are non-descendants of $V$ in the underlying BN. This is because since for any variable, $V$, it is straightforward to identify the set $\mathbf{C}_S$, so if the set of non-descendants $(\mathbf{C}_S \setminus \mathbf{C}_B)$ is also known, then their difference is equal to $\mathbf{C}_S \setminus (\mathbf{C}_S \setminus \mathbf{C}_B) = \mathbf{C}_B$, which uncovers the parents of $V$ in $\mathcal{B}$.

We are now ready to give an intuitive description of the decompilation process: looking at the ancestors of any variable in $\mathcal{SPN}_\mathcal{C}$, $V$, we can immediately identify $\mathbf{C}_S$, which is a superset of $\mathbf{C}_B$. Furthermore, since $V$'s conditional distribution must respect $\mathcal{B}$, the local Markov property guarantees that only variables in $\mathbf{C}_S \setminus \mathbf{C}_B$ (i.e. non-descendants in $\mathcal{B}$) can be removed from $\mathbf{C}_S$ without affecting the distribution. This means that once we establish a way to remove redundant variables from $\mathbf{C}_S$, then the ones that are going to remain by the end of the elimination process are going to be exactly the parents of $V$ in $\mathcal{B}$. It should be noted that this interplay between the local Markov property in $\mathcal{B}$ and the conditioning sets in $\mathcal{SPN}_\mathcal{C}$ is novel to the presented approach, and is not found in any existing decompilation algorithm, including the one presented in (Butz et al., 2020), which has exactly the same scope as our work. All previous algorithms rely on structural SPN properties alone, where the presented approach utilizes information about $\mathcal{B}$, yet in a way that does not require $\mathcal{B}$ to be known beforehand, rather only involving properties that follow from its existence.

The only thing remaining is to come up with a way for identifying the variables in $\mathbf{C}_S \setminus \mathbf{C}_B$. However, this can be easily done by using the following proposition:

**Proposition 3.3.** *Let $Y, X_1, X_2, \cdots, X_n$ be binary random variables. If there is a $X_k$, such $\Pr(Y|X_1, X_2, \cdots, X_k = 0, \cdots, X_n) = \Pr(Y|X_1, X_2, \cdots, X_k = 1, \cdots, X_n)$, then $Y \perp X_k | X_1, X_2, \cdots, X_{k-1}, X_{k+1}, \cdots, X_n$.*

Finally, Proposition 3.3 can be invoked implicitly by considering appropriate *induced trees up to a sum node $S'$*, which generalize standard *induced trees* (Zhao et al., 2016).

**Definition 3.4.** Let $\mathcal{S}$ be a complete and decomposable SPN over variables $X_1, \cdots, X_n$, and $T = (T_V, T_E)$ be a subgraph of $\mathcal{S}$. $T$ is called an induced subtree up to node $S'$ of $\mathcal{S}$ if:

**Algorithm 1** SPN to BN decompilation

**Require:** A SPN over $X_1, \cdots X_n$, compiled using the VErto algorithm, $\mathcal{SPN}_{\mathcal{C}}$
  $\mathcal{B} \leftarrow$ the empty BN over $X_1, \cdots X_n$
  $\mathit{Not\text{-}visited} \leftarrow \{X_1, \cdots X_n\}$
  **while** $\mathit{Not\text{-}visited} \neq \emptyset$ **do**
    Pick a variable $X_k \in \mathit{Not\text{-}visited}$
    $\mathbf{Rep}_{X_k} \leftarrow \{S | S$ is a sum node, Represent$(S) = X_k\}$
    $\mathit{Ancestors}_{X_k} \leftarrow \{X_m | \exists\, S$: Represent$(S) = X_m, S$ is above nodes in $\mathbf{Rep}_{X_k}\}$
    **if** $\mathit{Ancestors}_{X_k} = \emptyset$ **then**
      $\mathit{Not\text{-}visited} \leftarrow \mathit{Not\text{-}visited} \backslash \{X_k\}$
      continue
    **end if**
    **for** every variable $X_m \in \mathit{Ancestors}_{X_k}$ **do**
      $\mathbf{Rep}_{X_m} \leftarrow \{S | S$ is a sum node, Represent$(S) = X_m\}$
      **for** every $S_m \in \mathbf{Rep}_{X_m}$ **do**
        $\mathit{Trees}^{S_m} \leftarrow \{T | T \in \mathit{Subtrees}^S(\mathcal{SPN}_{\mathcal{C}})$ for a $S \in \mathbf{Rep}_{X_k}, S_m \in T\}$
        $\mathit{PairedTrees}^{S_m} \leftarrow \{(t_0, t_1) | t_0, t_1 \in \mathit{Trees}^{S_m}, \mathbf{1}_{X_m=0} \in t_0, \mathbf{1}_{X_m=1} \in t_1,$
          otherwise they contain the same indicators$\}$
        **for** every $(t_0, t_1) \in \mathit{PairedTrees}$ **do**
          $S_0 \leftarrow$ the end node in $t_0$
          $S_1 \leftarrow$ the end node in $t_1$
          **if** $S_0 \neq S_1$ **then**
            Add $X_m \rightarrow X_k$ to $\mathcal{B}$
            break
          **end if**
        **end for**
      **end for**
    **end for**
    $\mathit{Not\text{-}visited} \leftarrow \mathit{Not\text{-}visited} \backslash \{X_k\}$
  **end while**
  return $\mathcal{B}$

- $\mathit{Root}(\mathcal{S}) \in T_V$.
- If $v \in T_V$ is a sum node, then exactly one child of $v$ in $\mathcal{S}$ is in $T_V$, and the corresponding edge is in $T_E$.
- If $v \in T_V$ is a product node, then all children of $v$ in $\mathcal{S}$ are in $T_V$, and the corresponding edges are in $T_E$
- $S' \in T_V$, and once $S'$ is reached, the tree expansion stops.

An induced tree up to a sum node $S'$ results from traversing a SPN top-down, such that for every sum node only one of its children is included in the tree, for every product node all of its children are in the tree, and as soon as $S'$ is reached the procedure terminates. For a SPN $\mathcal{S}$, we denote the collection of all induced subtrees up to node $S'$ as $\mathit{Subtrees}^{S'}(\mathcal{S})$. The terminal node models the conditional distribution of the variable represented by $S'$, given the context implied by the indicators in the tree. For example, the sum node in the double red circle in

Figure 3.4d is the terminal node of the induced tree in red, and it models $\Pr(X_3|X_1 = 0, X_2 = 1)$, since this tree contains indicators for $X_1 = 0, X_2 = 1$. In turn, if $S_0, S_1$ represent the same variable $V$, comparing $T_0 \in Subtrees^{S_0}(\mathcal{S}), T_1 \in Subtrees^{S_1}(\mathcal{S})$, where $T_0, T_1$ differ only in the state of an indicator made for a single variable $X$, implicitly invokes Proposition 3.3. If $S_0 \neq S_1$, then the equality in 3.3 does not hold, meaning $X$ cannot be removed from the conditioning set, so by the local Markov property $X \in \mathbf{C}_B$, i.e. $X$ must be a parent of $V$ in $\mathcal{B}$. Alternatively, if all such nodes are identical, then $X$ can be removed from the conditioning set without affecting the distribution, which means that $X \in \mathbf{C}_S \setminus \mathbf{C}_B$. This way, Algorithm 1 utilizes a unique blend of information about both the structure and parameters of $\mathcal{SPN}_\mathcal{C}$ to enable an exact decompilation.

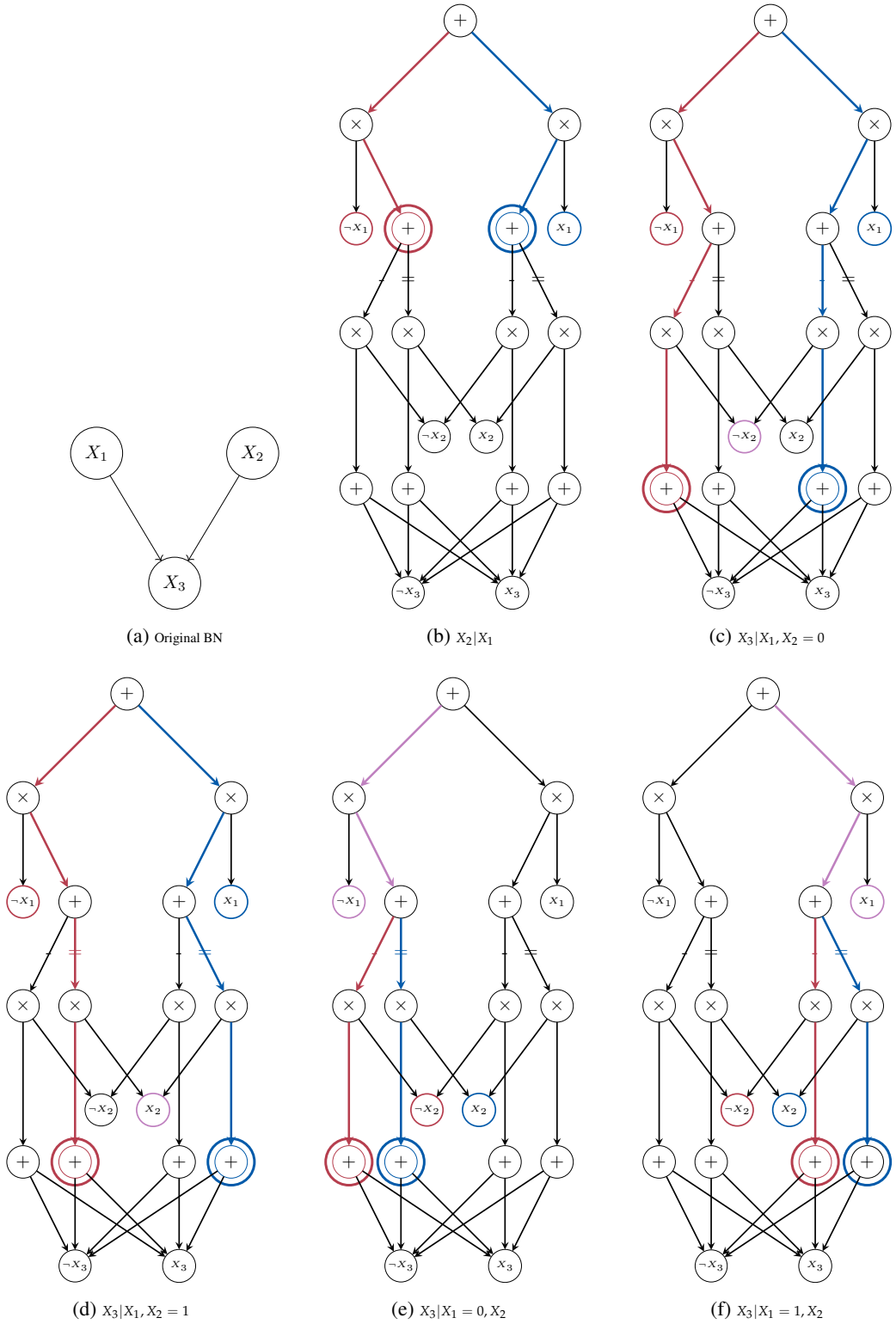
The following theorem is the culmination of our analysis, proving the validity of Algorithm 1.

**Theorem 3.5.** *Let $\mathcal{B}$ be a BN that is an I-map for the distribution it represents, and let $\mathcal{SPN}_\mathcal{C}$ be the SPN that results from compiling $\mathcal{B}$ using the VErto algorithm. Then applying Algorithm 1 to $\mathcal{SPN}_\mathcal{C}$ outputs exactly $\mathcal{B}$.*

**Example:** Figure 3.4 demonstrates the general process with an example. The original BN is shown in Figure 3.4a, while the remaining figures show the compiled SPN, along with the trees that have to be examined following Algorithm 1. It is well known that compiling the BN in 3.4a results in a SPN where the sum nodes in double circles in Figure 3.4b must have identical parameters (Peharz et al., 2014), which is why the corresponding edges are marked. Apart from that, the parameters of the remaining sum nodes are in general distinct, since they are not bound to satisfy any constraint.

Starting with $X_1$, we see that $\mathbf{C}_S = \emptyset \Rightarrow \mathbf{C}_B = \emptyset$, so $X_1$ has no parents in the underlying BN. Next, moving on to $X_2$, we have that $\mathbf{C}_S = \{X_1\}$. Looking at Figure 3.4b, we need to compare the sum node in red (which models the distribution $\Pr(X_2|X_1 = 0)$) to the node in blue (which models $\Pr(X_2|X_1 = 1)$). Since these nodes have identical parameters, we have that $\Pr(X_2|X_1 = 0) = \Pr(X_2|X_1 = 1)$, so by Proposition 3.3 we conclude that $\Pr(X_2|X_1) = \Pr(X_2)$, which combined with the local Markov property makes sure that $X_1 \notin \mathbf{C}_B \Rightarrow \mathbf{C}_B = \emptyset$, so neither $X_2$ has a parent in the underlying BN. It only appears as if $X_1$ influences the distribution of $X_2$ because all layers of a SPN are connected, but this is not really the case. Moving on to $X_3$, we see that $\mathbf{C}_S = \{X_1, X_2\}$. Figures 3.4c, 3.4d highlight the trees that need to be considered to decide whether $X_1$ belongs to $\mathbf{C}_B$. Comparing the corresponding sum nodes, we see that they model distinct distributions (since they do not not have equal parameters), meaning that $X_1 \in \mathbf{C}_B$, so the edge $X_1 \to X_3$ is added to the BN. Finally, we have to repeat this process for $X_2$, which is shown in Figures 3.4e, 3.4f. Again, since the final sum nodes are not identical, the edge $X_2 \to X_3$ is added, exactly recovering the BN in Figure 3.4a.

The above example shows how Algorithm 1 combines information about the structure and parameters of the compiled SPN in order to perform an exact decompilation. In contrast, even for relatively simple cases, like those in Figures 3.2a, 3.4a, all existing algorithms fail to




(a) Original BN (b) $X_2|X_1$ (c) $X_3|X_1, X_2 = 0$

(d) $X_3|X_1, X_2 = 1$ (e) $X_3|X_1 = 0, X_2$ (f) $X_3|X_1 = 1, X_2$

Figure 3.4: Illustration of applying Algorithm 1. Indicators in red are reached traversing the red paths, while indicators in blue are reached via blue paths. Indicators and edges in purple are belong to both red and blue paths. Edges marked with the same symbol have equal parameters.

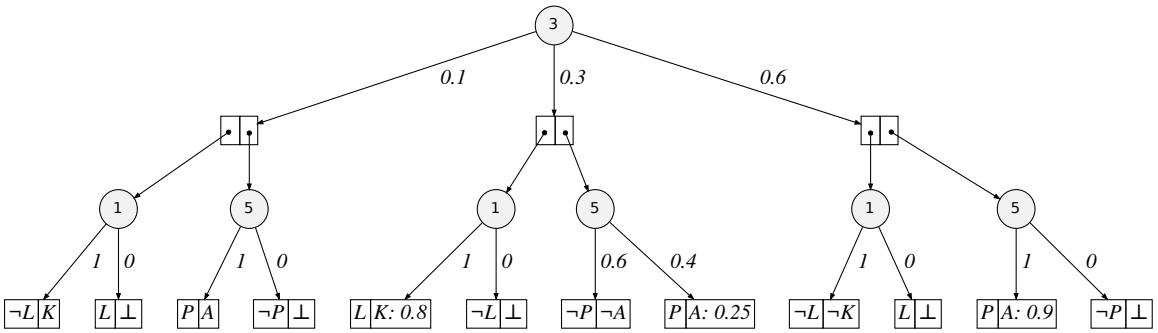



Figure 3.5: A PSDD over variables $A, L, K, P$, originally in (Kisa et al., 2014).

perform this task. This limitation has had a direct impact on a number of related works, such as the one presented in (Darwiche, 2022), where the author has to maintain both a BN and the corresponding compiled SPN in order to study interventional queries, despite the fact that all computations are performed using just the SPN. This is because maintaining only the SPN would result in losing all the information about the way variables are connected, hindering any subsequent causal analysis. Nevertheless, using Algorithm 1 it is no longer necessary to maintain the original BN, since all this information can be recovered at any point. This means that the compiled SPN can be used to compute both standard observational queries, as well as interventional ones (by first decompiling it to a BN and then following (Darwiche, 2022)). This observation is also related to the results in (Zečević et al., 2021b), where the authors argue that observational and interventional queries each require a distinct SPNs. However, SPNs compiled from BNs demonstrate that it is possible for a single SPN to answer both kinds of queries, mitigating the need to maintain separate SPNs.

## 3.4 A novel PSDD graphical representation

This section is concerned with the problem of representing PSDDs in a way that uncovers their internal representations and improves their transparency. Although this is a similar endeavour to the one in the previous section, this time the analysis will be based on a key feature of PSDDs, that is the ability to establish a deterministic dependency between a node and its children. More precisely, if $n = (p_1 \wedge s_1) \vee \cdots \vee (p_k \wedge s_k)$ is a PSDD node, then its base can be written as $[n] = \bigvee_{i=1,\ldots,k} [p_i] \wedge [s_i]$. This means that the value of node $n$ is uniquely determined by the values of its children. This is in contrast to SPNs, where each node defines a distribution but has no intrinsic value, since a PSDD node, $n$, both defines a distribution **and** represents the formula $[n]$.

The idea behind the proposed algorithm is to interpret $[n]$ and $[p_i], [s_i],\ i = 1, \ldots, k$ as having a parent-child relationship, in order to generate a CG that represents the internal connections in a PSDD. Consequently, if a PSDD is defined over variables $\mathbf{V}$, the goal of the proposed transformation is not to capture the connections between these variables, but to make clear in

which ways they are combined into more complex expressions by the PSDD. In a sense this is similar to the SPN de-compilations in (Zhao et al., 2015; Peharz et al., 2016; Butz et al., 2020), but instead of having latent variables with no meaningful interpretation, this time each of the newly introduced variables corresponds to a propositional formula over $\mathbf{V}$. Algorithm 2 presents the details of the transformation. It takes as input the propositional formula represented by the PSDD and recursively decomposes it to generate a CG.

**Algorithm 2** PSDD to CG

**Require:** The SDD base $\phi = c_1 \vee \cdots \vee c_n$, where $c_i = p_i \wedge s_i$, over variables $x_1, \ldots, x_k$
Create a variable corresponding to the whole expression, $v_0 = \phi$
Create a variable, $v_i$, for each $c_i$
Create an arrow from $v_i$ to $v_0$, $i = 1, \ldots, n$
**for** each $c_i = p_i \wedge s_i$ **do**
  Create a variable $v_i^p$ for $p_i$ and a variable $v_i^s$ for $s_i$
  Create an arrow from $v_i^j$ to $v_i$, for $j \in \{p, s\}$
**end for**
Repeat this process recursively, until the literal are reached
Once this procedure is over, connect the literals with undirected edges, so they form a chain

This transformation results in a model containing directed edges, with the exception of the edges between the observable variables (i.e. the literals), which are undirected. This choice, of keeping these edges undirected, reflects the following:

- There exists some correlation among these variables.
- No assumption is made about the source of it.

In the absence of additional information regarding the direction of the dependence between the variables, including undirected edges allows for a general treatment of the subject, without resorting to additional assumptions. Of course, in the presence of additional related information, the undirected edges could be replaced with directed ones. On the other hand, the alternative strategy of not including any edges between the observable variables, would encode the assumption that they are independent, which is rather strong and limiting.

We can now define a probability distribution over the CG resulting from Algorithm 2. From now on, the newly introduced variables will be referred to as *augmented variables*, denoted with $\mathbf{A}$, while the set of the original variables will be denoted with $\mathbf{V}$. Using this notation, a joint distribution over the augmented and original variables can be defined as follows:

- If $A \in \mathbf{A}$, and $\mathbf{PA}_A$ denotes the set of its parents, then:

$$\Pr(A = 1 | \mathbf{PA}_A) = \begin{cases} 1 & \text{if assignments in } \mathbf{PA}_A \text{ render } A = 1 \\ 0 & \text{otherwise} \end{cases}$$

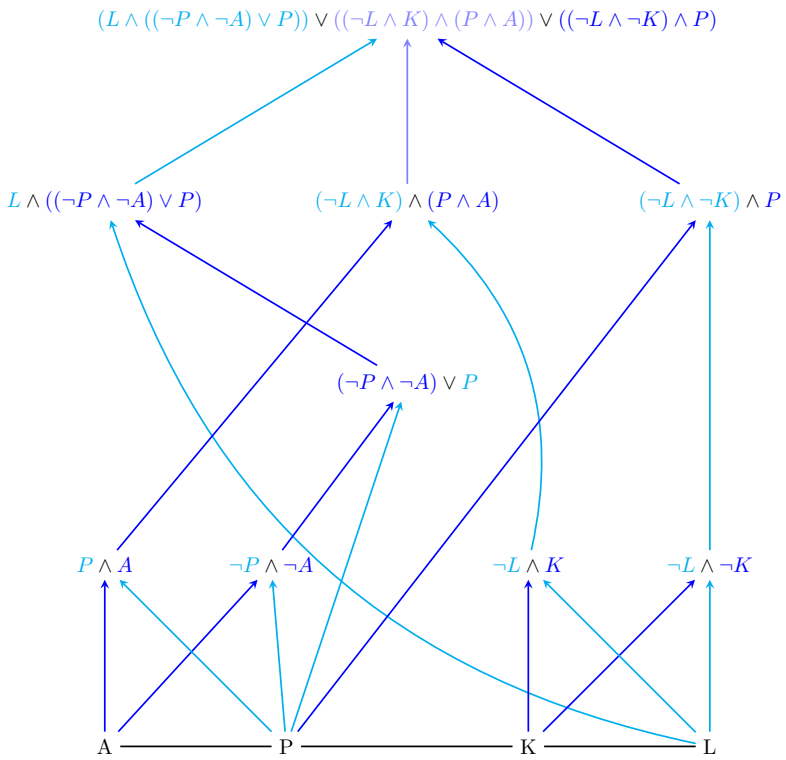



Figure 3.6: The resulting CG. Colored edges connect parent nodes (created for the same colored clause), to their children.

- $\Pr(\mathbf{V}) = \Pr_{PSDD}(\mathbf{V})$, where $\Pr_{PSDD}(\mathbf{V})$ is the PSDD distribution.

It is not difficult to see that the above construction yields a valid distribution over the CG:

**Theorem 3.6.** *Let $\mathcal{G}$ be the CG resulting from Algorithm 2, and $\boldsymbol{A}, \boldsymbol{V}$ be the sets of augmented and original variables, respectively. Then their joint distribution over $\mathcal{G}$ factorizes as:*

$$\Pr(\boldsymbol{V}, \boldsymbol{A}) = \Pr_{PSDD}(\boldsymbol{V}) \cdot \prod_{A \in \boldsymbol{A}} \Pr(A | \boldsymbol{PA}_A)$$

**Example:** The following example showcases how to construct a CG using Algorithm 2. The PSDD in Figure 3.5 corresponds to a problem considered in (Kisa et al., 2014). L (Logic), K (Knowledge Representation), P (Probability), and A (Artificial Intelligence), represent university courses. Students must enroll to them, while also obeying the following constraints: $P \vee L$, $A \Rightarrow P$, $K \Rightarrow (A \vee L)$ (where implication means if they enroll in LHS, then they must enroll in RHS).

The underlying SDD corresponds to the following propositional formula:

$$((\neg L \wedge K) \wedge (P \wedge A)) \vee (L \wedge ((\neg P \wedge \neg A) \vee P)) \vee ((\neg L \wedge \neg K) \wedge P) \tag{3.2}$$

Algorithm 2 takes (3.2) as input and constructs a CG, as follows: It first creates a node corresponding to the whole expression. Then, since (3.2) is composed of three disjunctions, three new nodes are created, with edges pointing from them to their disjunction. This procedure goes on recursively, until the literals, here $A, L, K, P$, are reached (see Figure 3.6).

By construction, $\mathbf{V} = \{A, P, K, L\}$, while $\mathbf{A}$ contains the following variables:

$$A_1 = P \wedge A, A_2 = \neg P \wedge \neg A, A_3 = \neg L \wedge K, A_4 = \neg L \wedge \neg K, A_5 = P \wedge A_4,$$
$$A_6 = A_1 \wedge A_3, A_7 = P \vee A_2, A_8 = L \wedge A_7, A_9 = A_5 \vee A_6 \vee A_8$$

### 3.4.1 Counterfactuals

The transformation presented in the previous section makes the internal connections in PSDDs clear and easily accessible. In this section we show that by taking advantage of the resulting CG it is possible to establish novel connections between PSDDs and concepts from causal inference. Intuitively, since the value of each variable in the CG is deterministically decided by its parents, we can consider the parents as causing their children to attain their value, thus opening the door to study counterfactual statements involving the PSDD's internal rules. This means that on top of serving as a transparent representation of a PSDD, the final CG allows for expanding the PSDD semantics beyond standard conditioning, enabling us to assess how probable it would have been for a rule to be satisfied (or not) had some conditions been different.

Furthermore, although it is not clear how to study counterfactuals for general CGs, the specific form of the ones resulting from Algorithm 2 allows for mitigating this limitation. This is because apart from the variables in $\mathbf{V}$, the remaining graph is directed, so it can be treated just like a regular BN. This observation is reflected in the following theorem, stating that interpreting $\mathbf{V}$ in the CG resulting from Algorithm 2, the same way as the exogenous variables, $\mathbf{U}$, in a BN, leads to a valid counterfactual distribution ($X|_{Y=y}$ denotes substituting all appearances of $Y$, in $X$, with $y$):

**Theorem 3.7.** *Let $\mathcal{G}$ be the result of Algorithm 2, then a counterfactual query of the form $\Pr(A = a|do(X = x), E = e)$, where $A \in \boldsymbol{A}$, $X, E \in \boldsymbol{A} \cup \boldsymbol{V}$, is equal to:*

$$\sum_{\boldsymbol{V}:\boldsymbol{V}\models A|_{X=x}=a} \Pr^*(\boldsymbol{V}) = \frac{\Pr(A|_{X=x} = a, E = e)}{\Pr(E = e)} = \Pr(A|_{X=x} = a|E = e),$$

*where $\Pr^*(\boldsymbol{V}) = \frac{\Pr(\boldsymbol{V})}{\Pr(E=e)}$ if $\boldsymbol{V} \models E = e$, and $0$ otherwise.*

**Example (Cont.):** Going back to our previous example, suppose we have observed that the branching rule $A \wedge P$ was not satisfied; meaning there is a student who did not enroll to both $A$ and $P$. What is the probability that the rule would have been satisfied, had the student enrolled to $P$? Formally, we ask for the probability of $\Pr(A_1 = 1|do(P = 1), A_1 = 0)$. According to

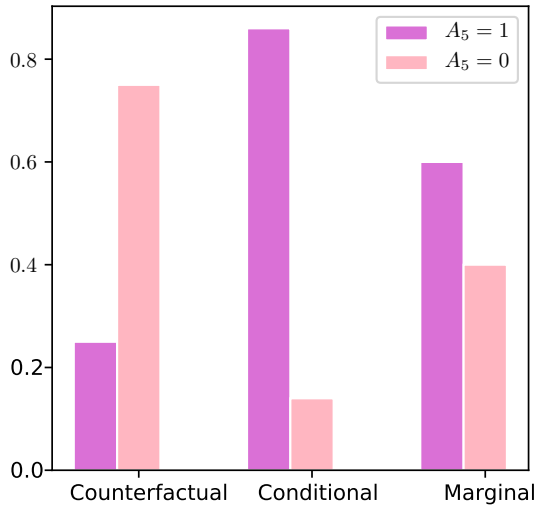



Figure 3.7: Comparison of distributions

Theorem 3.7, the updated distribution is $\Pr(A, L, K, P|A_1 = 0) = \frac{\Pr(A,L,K,P)}{\Pr(A_1=0)}$ if $A = 0$ or $P = 0$, and 0 otherwise. Furthermore, following the intervention $P = 1$, the updated equation of $A_1$ is equal to $A_1 = A$, meaning that the desired probability can be easily computed as $\Pr^*(A_1 = 1) = \Pr^*(A = 1) = \frac{\Pr(A=1,P=0)}{\Pr(A_1=0)}$.

The same process can be repeated for any query; for example, suppose there is a student not satisfying the property "he/she has enrolled to $P$, while not enrolling to neither $L$ or $K$", what is then the probability of him/her satisfying this property, had he/she enrolled to $K$? This time the probability of interest is $\Pr(A_5 = 1|do(K = 0), A_5 = 0)$, which reduces to $\Pr(L = 0, K = 1, P = 1)/0.4 = 0.1/0.4 = 0.25$, by repeating the same steps, as before. Figure 3.7 compares this counterfactual distribution to the conditional, $\Pr(A_5|K = 0)$, and the plain marginal, $\Pr(A_5)$, where all computations are based on the distribution represented by the PSDD in Figure 3.5.

The above provides some insights into the way counterfactuals of the form $\Pr(E = e'|do(X), E = e)$, such as $\Pr(A_1 = 1|do(P = 1), A_1 = 0)$, are computed. Looking at the previous calculations, this probability is equal to $\frac{\Pr(A=1,P=0)}{\Pr(A \wedge P=0)}$, so although the intervention forced $P = 1$, the probability in the numerator involves the event $P = 0$. An intuitive way to read the numerator is that it computes the probability that the variable $P$ getting another value (than the one intervened upon) was responsible for observing the evidence. In this case, since the intervention was $P = 1$, we ask how responsible was the event $P = 0$, for observing that $A \wedge P = 0$. The only compatible configurations with this scenario are $A = 1, P = 0$ and $A = 0, P = 0$. The result takes only the former into account, because in the latter it is $A = 0$, so it is not $P = 0$ that is solely responsible for observing the evidence, since $A \wedge P = 0$, regardless of $P$'s value.

## 3.5 Discussion and Conclusions

In this work, we studied ways to enhance the transparency of SPNs and PSDDs. For SPNs, it was shown that existing BN interpretations result in very restricted graphs that neither reveal the SPNs' internal representations nor allow for studying causal queries. To address this issue, we identified a set of sufficient conditions as well as an algorithm that incorporates multiple sources of information, in order to achieve an exact SPN to BN de-compilation. For PSDDs, a novel CG interpretation that uncovers their internal reasoning was established, thus allowing for expanding their semantics to include a notion of counterfactuals.

There is a number of interesting future research directions stemming from these results. For example, looking back at the derivation of Algorithm 1, we see that properties 1 and 2 are the only essential requirements to guarantee that an exact decompilation is possible. This implies that the algorithm should be applicable to any SPN satisfying them, not only those compiled using the VErto algorithm. An immediate consequence is that even when learning a SPN from data, instead of compiling it from a BN, as long as these two properties are enforced during training, it should be possible to generate a BN without any additional hidden variables that agrees with the SPN distribution. This points towards a very interesting research direction, where fully transparent models are learned directly from data in an end-to-end manner, combining the efficient inference of SPNs with the representational transparency of BNs, opening the door for utilizing SPNs in applications where model transparency is essential (Rudin, 2019). This is in stark contrast to existing approaches, which pose serious transparency limitations. Other interesting directions include studying potential ways to utilize the PSDD counterfactuals to quantify the importance of a feature (or internal rule), or appropriately adjusting Algorithm 1 so it can handle SPNs representing BNs with context-specific independence.

## 3.6 Proofs

**Rules of do-Calculus**

What follows is an essential graphical tool for deciding under what conditions we can reduce interventional queries to conditional ones. Here, $\mathcal{B}_{\overline{X}}$ denotes the graph obtained after deleting all the edges pointing to $X$, $\mathcal{B}_{\underline{X}}$ the one resulting after deleting all the edges emerging from $X$, and $\mathcal{B}_{\overline{X}\underline{Z}}$ refers to the graph where both edges incoming to $X$ and stemming from $Z$ are deleted.

**Definition 3.8. (Rules of do-Calculus)** Let $\mathcal{B}$ be a BN, and $\Pr(\cdot)$ the probability measure induced by it. If **X, Y, Z, W** are disjoint sets, then the following hold:

**Rule 1:** $\Pr(\mathbf{y}|do(\mathbf{x}),\mathbf{z},\mathbf{w}) = \Pr(\mathbf{y}|do(\mathbf{x}),\mathbf{w})$ if $(\mathbf{Y}\perp\mathbf{Z}|\mathbf{X},\mathbf{W})_{\mathcal{B}_{\overline{X}}}$.

**Rule 2:** $\Pr(\mathbf{y}|do(\mathbf{x}),do(\mathbf{z}),\mathbf{w}) = \Pr(\mathbf{y}|do(\mathbf{x}),\mathbf{z},\mathbf{w})$ if $(\mathbf{Y}\perp\mathbf{Z}|\mathbf{X},\mathbf{W})_{\mathcal{B}_{\overline{X}\underline{Z}}}$.

**Rule 3:** $\Pr(\mathbf{y}|do(\mathbf{x}),do(\mathbf{z}),\mathbf{w}) = \Pr(\mathbf{y}|do(\mathbf{x}),\mathbf{w})$ if $(\mathbf{Y}\perp\mathbf{Z}|\mathbf{X},\mathbf{W})_{\mathcal{B}_{\overline{X}\overline{Z(W)}}}$, where **Z(W)** is the set of **Z**-nodes that are not ancestors of any **W**-node in $\mathcal{B}_{\overline{X}}$.

**Proof of Proposition 3.1**

Using the 3rd rule of Pearl's do-calculus, it suffices to show that $(\mathbf{X}\perp\mathbf{U}\cup(\mathbf{V}\setminus\mathbf{X}))_{\mathcal{B}_{\overline{X}}}$. By assumption, no edges emanate from nodes in **X**, which implies that each of them will be isolated in $\mathcal{B}_{\overline{X}}$, so the desired independence holds, meaning that $\Pr(\mathbf{U},\mathbf{V}_{-\mathbf{X}}|do(\mathbf{X})) = \Pr(\mathbf{U},\mathbf{V}_{-\mathbf{X}})$. In addition, we have that $\Pr(\mathbf{V}_{-\mathbf{X}}|do(\mathbf{X})) = \sum_{\mathbf{U}}\Pr(\mathbf{U},\mathbf{V}_{-\mathbf{X}}|do(\mathbf{X})) = \sum_{\mathbf{U}}\Pr(\mathbf{U},\mathbf{V}_{-\mathbf{X}}) = \Pr(\mathbf{V}_{-\mathbf{X}})$. □

**Proof of Theorem 3.2**

All the proposed transformations result in bipartite BNs, with edges only from latent to observable variables. Since no edges stem out of the observable variables, the result follows by applying the above proposition. □

**Proof of Proposition 3.3**

Without loss of generality, we can assume that $k = n$, since otherwise we can just swap indices between $X_k$ and $X_n$. Applying the law of total probability, we get that

$$\Pr(Y|X_1,\cdots,X_{n-1}) = \sum_{X_n}\Pr(Y,X_n|X_1,\cdots,X_{n-1}) =$$
$$\Pr(Y|X_1,\cdots,X_{n-1},X_n=0)\cdot\Pr(X_n=0|X_1,\cdots,X_{n-1})+$$
$$\Pr(Y|X_1,\cdots,X_{n-1},X_n=1)\cdot\Pr(X_n=1|X_1,\cdots,X_{n-1}) =$$
$$\Pr(Y|X_1,\cdots,X_{n-1},X_n=0)\cdot\Pr(X_n=0|X_1,\cdots,X_{n-1})+$$
$$\Pr(Y|X_1,\cdots,X_{n-1},X_n=0)\cdot\Pr(X_n=1|X_1,\cdots,X_{n-1}) =$$
$$\Pr(Y|X_1,\cdots,X_{n-1},X_n=0)\cdot(\Pr(X_n=0|X_1,\cdots,X_{n-1})+\Pr(X_n=1|X_1,\cdots,X_{n-1})) =$$

$\Pr(Y|X_1, \cdots, X_{n-1}, X_n = 0)$

By symmetry, we also have that

$$\Pr(Y|X_1, \cdots, X_{n-1}, X_n = 1) = \Pr(Y|X_1, \cdots, X_{n-1}),$$

concluding the proof. □

**Proof of Theorem 3.5**

Let $V$ be a variable represented by some sum node in $\mathcal{SPN}_{\mathcal{C}}$, and $\mathbf{S}$ be the set of all sum nodes representing $V$, $\mathbf{S} = \{S|S \text{ is a sum node, Represent}(S) = V\}$. Properties 1 and 2 imply that all $S \in \mathbf{S}$ model the conditional distribution of $V$, given a configuration of its ancestors in $\mathcal{SPN}_{\mathcal{C}}$, $X_1, \cdots, X_k$. Furthermore, both $\mathcal{B}$ and $\mathcal{SPN}_{\mathcal{C}}$ represent the same distribution, which factorizes according to $\mathcal{B}$, so it satisfies the local Markov property. This means that $\Pr(V|X_1, \cdots, X_k) = \Pr(V|P_1, \cdots, P_m)$, where $P_1, \cdots, P_m$ are $V$'s parents in $\mathcal{B}$. This is because since $\mathcal{SPN}_{\mathcal{C}}$ was compiled using a topological ordering all of the $P_i$'s are within the set $\{X_1, \cdots, X_k\}$, and the remaining variables are non-descendants. Finally, Proposition 3.3 is used in order to identify the variables that should be removed from the conditional, i.e. the non-descendants of $V$ in $\mathcal{B}$. This is done in an implicit way, by just comparing certain induced trees, as follows:

- Pick a variable, $X_m$, that is represented by a sum node that is closer to the root than all nodes in $\mathbf{S}$. Any variable satisfying this property appears before $V$ in the underlying topological ordering.
- Pick a node, $S_m$, representing $X_m$, and let $Trees^{S_m} = \{T|T \in Subtrees^S(\mathcal{SPN}_{\mathcal{C}}) \text{ for some } S \in \mathbf{S}, S_m \in T\}$ be the set of all induced sub-trees that pass through $S_m$ and end in one of the nodes in $\mathbf{S}$
- Let $t_0, t_1 \in Trees^{S_m}$, such that they both contain exactly the same indicators for all variables, except from $X_m$. Instead, when it comes to $X_m$, $\mathbf{1}_{X_m=0} \in t_0, \mathbf{1}_{X_m=1} \in t_1$.
- Furthermore, let $S_0, S_1 \in \mathbf{S}$ be the end nodes of $t_0, t_1$, respectively. Then $S_0$ models the distribution of $V|\mathbf{x}_{1:k,-m}, X_m = 0$, while $S_1$ models $V|\mathbf{x}_{1:k,-m}, X_m = 1$, where $\mathbf{x}_{1:k,-m}$ are the states of the variables $X_1, \cdots, X_{m-1}, X_{m+1}, \cdots, X_k$, which are the same in both conditionals, since $t_0, t_1$ include the same indicators for each of these variables. By construction, the two conditioning sets only differ in the state of $X_m$.
- If $S_0 \neq S_1$, then by Proposition 3.3 we conclude $X_m$ and $V$ are not conditionally independent (since $\Pr(V|\mathbf{x}_{1:k,-m}, X_m = 0) \neq \Pr(V|\mathbf{x}_{1:k,-m}, X_m = 1)$), so $X_m$ must be a parent of $V$ in $\mathcal{B}$. On the other hand, if $S_0 = S_1$, we cannot reach a definite conclusion, so we consider a new pair of trees $t_0, t_1 \in Trees^{S_m}$ and repeat the same process. If after considering all such trees no two distinct sum nodes can be found, we then consider

another node representing $X_m$ and repeat the steps above. Finally, if this loop terminates without identifying two distinct sum nodes, then $\Pr(V|\mathbf{x}_{1:k,-m}, X_m = 0) = \Pr(V|\mathbf{x}_{1:k,-m}, X_m = 1)$ for all values of $\mathbf{x}_{1:k,-m}$, so again by Proposition 3.3 we can conclude that $X_m$ must not be a parent of $V$ in $\mathcal{B}$.

□

**Proof of Theorem 3.6**

This directly follows from the following observations:

- Any inconsistent joint assignment between variables in $\mathbf{V}$ and $\mathbf{A}$, e.g. $V_1 = 0, V_2 = 1, A_1 = V_1 \wedge V_2 = 1$, has a probability equal to zero, by definition.
- Any consistent joint assignment leads to all conditional distributions of variables in $\mathbf{A}$ being equal to 1, by definition, so it reduces to a probabilistic query involving only variables in $\mathbf{V}$. Now since the distribution of $\mathbf{V}$ is defined in terms of the PSDD, which represents a valid distribution, it is a valid distribution as well.

□

**Proof of Theorem 3.7**

The first step is to update the distribution of $\mathbf{V}$, given evidence $E = e$:

$$\Pr^*(\mathbf{V}) = \Pr(\mathbf{V}|E = e) = \frac{\Pr(\mathbf{V}, E = e)}{\Pr(E = e)}$$

At this point we should remind ourselves that every instantiation of $\mathbf{V}$, determines the values of all variables in the model, including $E$. This means, that in the above expression, all values of $\mathbf{V}$ that do not lead to $E = e$, result into the probability $\Pr(\mathbf{V}, E = e) = 0$. For the rest of the values we have that $\mathbf{V} \models E = e$, so $\Pr(\mathbf{V}, E = e) = \Pr(\mathbf{V})$.

The desired outcome follows, since if $A|_{X=x}$ is the updated equation determining the value of $A$ after performing an intervention, then, by the law of total probability, $\Pr(A|_{X=x} = a)$ is equal to the sum of the probabilities of all assignments of $\mathbf{V}$ that lead to $A|_{X=x} = a$. Utilizing the updated distribution of $\mathbf{V}$, $\Pr^*(\mathbf{V})$, to perform this calculation, we conclude that:

$$\Pr(A = a|do(X = x), E = e) = \sum_{\mathbf{V}:\mathbf{V}\models A|_{X=x}=a} \Pr^*(\mathbf{V}) = \sum_{\mathbf{V}:\mathbf{V}\models A|_{X=x}=a} \frac{\Pr(\mathbf{V}, E = e)}{\Pr(E = e)}$$
$$= \sum_{\mathbf{V}:\mathbf{V}\models A|_{X=x}=a \wedge \mathbf{V}\models E=e} \frac{\Pr(\mathbf{V}, E = e)}{\Pr(E = e)} = \sum_{\mathbf{V}:\mathbf{V}\models A|_{X=x}=a \wedge \mathbf{V}\models E=e} \frac{\Pr(\mathbf{V})}{\Pr(E = e)} = \frac{\Pr(A|_{X=x} = a, E = e)}{\Pr(E = e)}$$
$$= \Pr(A|_{X=x} = a|E = e)$$

□

# Chapter 4

# Constraints in SPNs

## 4.1 Introduction

Incorporating constraints is a major concern in data mining and probabilistic machine learning (Raedt et al., 2010; Friedman and Van den Broeck, 2019; Kisa et al., 2014), since a wide variety of problems require the prediction to be integrated with reasoning about various forms of constraints. This ranges from constraining the support of a distribution, such as when modelling routes on maps (Shen et al., 2018; Xu et al., 2018), to enforcing certain independence relationships, such as when approving loan predictions (Mahoney and Mohen, 2007). That is, when modelling routes, we may require the prediction model to respect the presence of physical paths between nodes on the map, in the sense of assigning zero probability to impossible or infeasible paths. Analogously, when approving loans, we may have conditional constraints for eliminating bias, e.g, the prediction should be independent of the applicant's ethnicity or gender.

While the problem of constraint satisfaction in BNs has been extensively studied, the relationship between constraints and TPMs remains an open question. In this chapter, we study the feasibility of incorporating various kinds of probabilistic constraints in SPNs. Although SPNs are closely related to BNs, the techniques that have been developed for the latter are not immediately applicable to the former, since their representations are dramatically different. In fact, this is related to the findings in Chapter 3, since whenever a SPN satisfies the identified sufficient conditions, it is possible to utilize the de-compiled BN, apply one of the existing BN constraint-related approaches, and then compile it back to a SPN. However, in this chapter we study the complementary situation, exploring whether it is possible to incorporate probabilistic constraints to arbitrary SPNs, even when they are not compiled from BNs. If the answer to this question is positive, then it means that SPNs exhibit some powerful transparent features even in the general case, allowing for integrating a priori expert knowledge.

Our focus is to explore whether the incorporation of probabilistic constraints in SPNs is at all feasible, on a theoretical level, as well as suggest an optimization based framework for carrying

out this task. Research towards this question has been extremely limited, since the only known result is that the constraints that are implied by the structure $X_1 \rightarrow X_3 \leftarrow X_2$ can be indeed encoded into a SPN (Peharz et al., 2014) by requiring certain parameters to be equal (see Figure 4.1). However, this does not address the general case where SPNs are not compiled from BNs, but are directly learned from data, instead. More precisely, we present the following contributions:

- We provide proofs that it is possible to enforce (unconditional) independence constraints, conditional independence constraints, as well as interventional ones to SPNs.
- Furthermore, we prove that all of them correspond to the SPNs' parameters satisfying a multivariate system of equations.
- Finally, we propose two different optimization schemes for training SPNs in the presence of such constraints.

These results extend SPN capabilities in a way that brings them closer to traditional BNs, moving them further towards the spectrum of transparent models. As far as suggesting an optimization framework, our approach for studying this question is inspired by recent approaches, where the problem of constraint satisfaction is addressed by adding regularization terms in the objective function (Xu et al., 2018; Marquez Neila et al., 2017). In the next section, we briefly review the main idea underlying this approach.

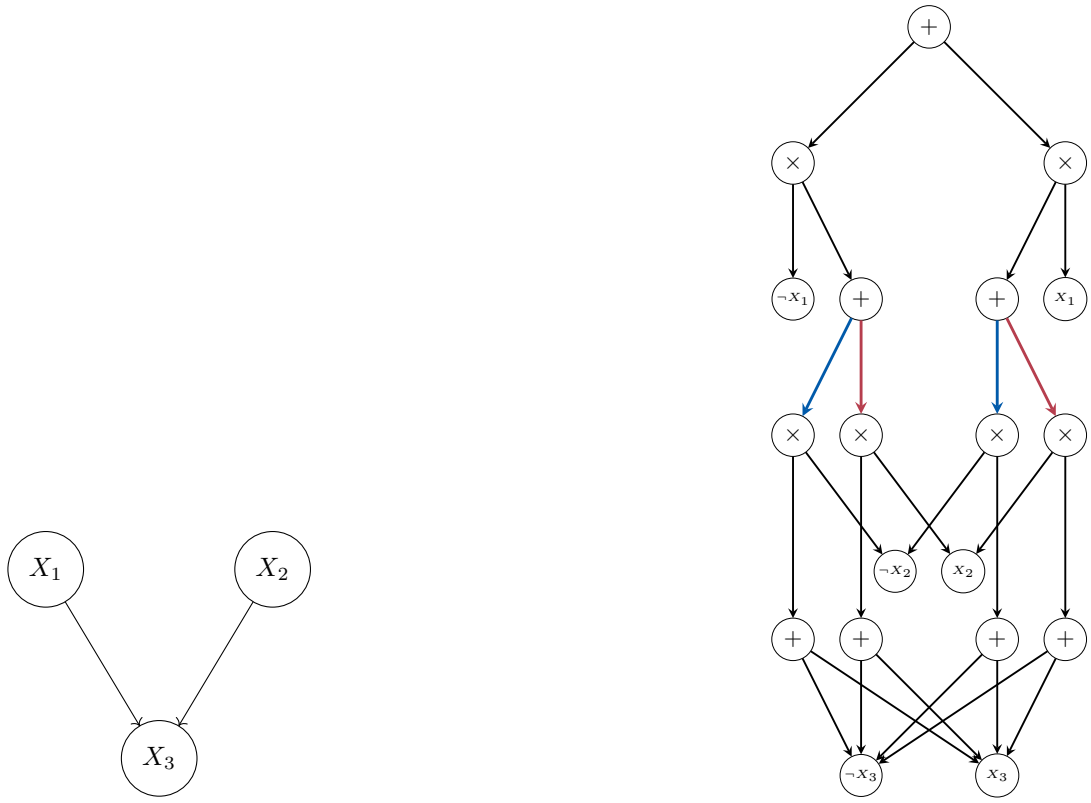
## 4.2 Background

### 4.2.1 Optimization

Constrained optimization is concerned with developing techniques allowing for optimizing functions under a set of constraints. For example, Figure 4.2 depicts the problem of minimizing a function, while requiring the solution to belong to the shaded area. One of the most common ways to address that, is to transform the objective function, so it takes the constraints into account. The problem of interest is to maximize the likelihood of a model (with a vector of parameters $\mathbf{w}$), $L(\mathbf{w})$, under constraints $C_i(\mathbf{w}) = 0,\ 1 \leq i \leq N$, so:

$$max_{\mathbf{w}} L(\mathbf{w}),\ \ s.t.\ C_1(\mathbf{w}) = 0, \cdots, C_N(\mathbf{w}) = 0$$

The transformed objective function, $\Lambda$, introduces a number of auxiliary variables, as many as the constraints, $\lambda_1, \cdots, \lambda_N$, and takes the following form $\Lambda(\mathbf{w}, \lambda_1, \cdots, \lambda_N) = L(\mathbf{w}) + \sum_{n=1}^{N} \lambda_n C_n(\mathbf{w})$. It can be shown that all of the solutions of the original problem correspond to stationary points of the new objective function (Protter and Morrey, 1985).




(a) BN (b) SPN

Figure 4.1: Examples of BN inducing an independence constraint, and the corresponding SPN. Imposing equality between parameters of same coloured edges enforces the independence constraint.

There are various numerical methods to solve this problem, such as projected gradient descent, where an initial vector $\mathbf{w}^{(0)}$ is updated incrementally, and then gets projected onto the surface defined by the constraints, until it converges to a solution of the problem. Furthermore, in cases where the objective function is in a special form, such as a quadratic polynomial, other approaches might be more efficient. See (Marquez Neila et al., 2017) for a more extensive discussion on the subject.

Alternative ways to address constraint optimization problems include recent advances, such as (Cotter et al., 2019a,b), where the optimization objective is formulated as a game between two players. Approaches like these can be readily incorporated within our framework, since we are going to make use of only differentiable constraints, as we will see in what follows.

Optimization problems like the above require all of the feasible solutions to satisfy the constraints. When this is the case, the constraints are referred to as *hard*. Alternative formulations of the problem could yield feasible solutions not satisfying the constraints. These constraints are called *soft*, because instead of demanding the solutions to adhere to them, we introduce a penalty term in the objective function, for each time they get violated. For example, if all of the $C_i(\mathbf{w}) = 0,\ 1 \leq i \leq N$ were treated as soft constraints, then after setting

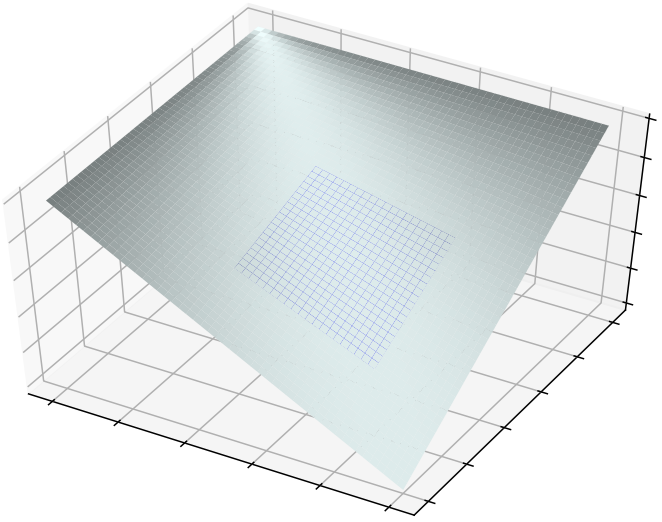


Figure 4.2: An example of optimizing a function, while constraining the solution to lie on the shaded part.

$\lambda_1, \cdots, \lambda_N$ to some value reflecting the cost of violating the corresponding constraint, the soft version of the problem would be to maximize the function $L(\mathbf{w}) + \sum_{n=1}^{N} \lambda_n C_n(\mathbf{w})$, so each time some $C_i$ is not equal to zero, it induces a penalty. In this case, all $\lambda_i$ are treated as hyperparameters, so they are specified before the optimization takes place. Furthermore, now we are interested in the maxima of this function, as opposed to the case of hard constraints, where we were interested in the stationary points of the transformed function.

## 4.3 Constraint satisfaction

In this section we establish a link between constraint satisfaction and the SPN's parameters, generalizing the results in (Peharz et al., 2014), where the authors discuss that in order to encode the triplet $X_1 \rightarrow X_3 \leftarrow X_2$ as a SPN, it is necessary to equate certain parameters (see Figure 4.1). The presented results are based on the fact that SPNs naturally exhibit a correspondence between parameters and probabilities. Although this is a simple observation, it is also crucial, since independence constraints can be usually expressed as an equality between probabilities. For example, if we want to incorporate the constraint that "$A$ is independent of $B$", we have to ensure that the equality $\Pr(A, B) = \Pr(A) \cdot \Pr(B)$ holds in the final model. This equality can be re-written in terms of the SPN parameters, allowing for uncovering a system of equations that guarantee that the constraint is satisfied. In what follows, we exploit this observation to prove it is feasible to incorporate various constraints in SPNs.

### 4.3.1 Conditional constraints

The first result concerns the case of constraining the likelihood so it enforces conditional independence between two variables. More precisely, assume variables $X_i, X_j$, and another

variable $X_k$, whose values we would like to condition on. Constraints similar to this appear in the fair AI literature (Zemel et al., 2013; Zafar et al., 2015; Hardt et al., 2016; Grgić-Hlača et al., 2016; Mary et al., 2019), where the objective is to eliminate bias, such as racial discrimination, from predictive models, by enforcing an appropriate set of conditions. For example, $X_i$ could represent the outcome of a loan application, $X_j$ could represent the applicant's ethnicity, and $X_k$ could be the applicant's salary. The goal of a conditional constraint, then, would be to make sure that the probability of granting a loan application is independent of the applicant's ethnicity, given the salary. It should also be noted that such properties can only be imposed during model training/calibration, since otherwise even the marginal distribution of a variable might have been biased, affected by information regarding the protected attribute, which leaked during training. This situation could arise under various circumstances, such as when applying a model to an imbalanced dataset (Mehrabi et al., 2021; Piotr Sapiezynski, 2017), in which case conditional constraints can impede such information leakage from happening.

As previously mentioned, SPNs allow for establishing a clear connection between probabilistic queries and the model's parameters. This is essential for our approach, since, in general, it is not clear how to achieve this connection. The following result establishes the relationship between conditional independence constraints and the parameters of an SPN.

**Theorem 4.1.** *Let $\mathcal{S}$ be a SPN representing the joint distribution of variables $X_1, \cdots, X_n$. Let $X_i, X_j, X_k$ be binary variables, then a conditional independence constraint of the form $X_i \perp X_j | X_k$ is equivalent to a quadratic multivariate system of equations on the SPN's parameters.*

At this point, it should be noted that although the above result is stated for binary SPNs, to make the presentation easier to follow, it holds for discrete SPNs in general, as shown in Section 4.8. Furthermore, by slightly modifying Theorem 4.1, it should be possible to enforce context-specific independence conditions in the SPN, such as $X_i \perp X_j | X_k = 1$. This showcases the flexibility of SPNs, since, to the best of our knowledge, there are no known results regarding whether context-specific independence can be incorporated into regular BNs.

### 4.3.2 Interventional constraints

Interventional distributions are used extensively in causal modelling, since they compute probabilities after an external intervention on a variable is performed, as discussed in the previous chapter. The new objective is to train a model while incorporating constraints of the form $\Pr(\mathbf{X}_{-i} | do(X_i = 0)) = \Pr(\mathbf{X}_{-i} | do(X_i = 1))$, where $\mathbf{X}_{-i}$ denotes the set of all model variables, excluding $X_i$. Constraints of this kind can have powerful implications regarding the causal mechanisms between $X_i$ and the rest of the variables.

The following is a well known formula connecting the interventional to the observational distribution of some variables (Pearl, 2009b), which will be utilized to prove Theorem 4.2:

$$\Pr(\mathbf{X}_{-i}|do(X_i = i)) = \frac{\Pr(\mathbf{X}_{-i}, X_i = i)}{\Pr(X_i = i|pa_{X_i})}$$

**Theorem 4.2.** *Let $\mathcal{S}$ be a SPN representing the joint distribution of variables $X_1, \cdots, X_n$. Let $X_i$ be a binary variable, then the constraint $\Pr(\boldsymbol{X}_{-i}|do(X_i = 0)) = \Pr(\boldsymbol{X}_{-i}|do(X_i = 1))$ is equivalent to a multivariate linear system of equations on the SPN's parameters.*

In section 4.8 we discuss how to extend this result to the general case.

### 4.3.3 Independence constraints

The last kind of constraints we are going to consider are those enforcing independence between variables. There are some already existing approaches, such as (Xu et al., 2018), allowing for incorporating rules expressed as propositional formulas within the model, in order for example to impose certain structure on the outcome variable, but doing the same with probabilistic ones still poses a major challenge. Using a similar reasoning as in the previous cases, it is possible to prove that incorporating independence constraints is analogous to the SPN's parameters satisfying a multivariate system of equations:

**Theorem 4.3.** *Let $\mathcal{S}$ be an SPN representing the joint distribution of variables $X_1, \cdots, X_n$. Let $X_i, X_j$ be two binary variables, then an (unconditional) independence constraint of the form $X_i \perp X_j$ is equivalent to a quadratic multivariate system of equations on the SPN's parameters.*

In section 4.8 we discuss how to extend this result to the general case.

## 4.4 System solutions

The results presented so far establish a relationship between various forms of constraints and multivariate systems of equations. Having said that, it stills remains to show that solutions to these systems exist. While the fact that parameter tuning can indeed be carried out effectively to enforce certain constraints in BNs (Meek, 1995) is reassuring, it is not enough to make sure that the systems identified in the previous sections have valid solutions. However, by utilizing their specific forms, it is possible to reach the conclusion that all of them can be solved indeed.

More specifically, a well known result coming from the field of commutative algebra states that a system of multivariate polynomials, where the number of unknown variables is greater than the number of equations in the system, can either have no solution or infinitely many solutions (Greuel et al., 2002; Cox et al., 2013; Sottile, 2011). Furthermore, looking at the equations that make up each of the systems in Theorems 4.1, 4.2, 4.3, which can be found in Section 4.8,

none of them has a constant term (i.e. the systems are *homogeneous*). This means that for each of these systems, setting all variables equal to zero yields a solution, which implies that the systems have in fact infinite solutions, which is in tune with the corresponding result for BNs (Meek, 1995). Consequently, as long as more than 8 variables appear in the system in Theorem 4.1, more than 2 variables appear in 4.2, and more than 4 variables appear in 4.3, all of them have an infinite number of solutions. However, since during optimization the primary objective is to maximize the likelihood, the value of the likelihood function can be used to select the solution that leads to the best performing model.

When it comes to solving these systems, on top of the standard numerical solvers, specialized algorithms have been developed to address such cases, like for example the cylindrical algebraic decomposition (Collins, 1975; Arnon et al., 1984). Finally, this section highlights the utility of Theorems 4.1, 4.2, 4.3, since on top of establishing a connection between SPN parameters and various kinds of constraints, by uncovering the exact form of the resulting systems they help us show that it is possible for all constraints to be enforced.

## 4.5 Revisiting existing constraints

Earlier in this chapter, it was briefly mentioned that the constraints implied by the BN structure in Figure 4.1a can be encoded into a SPN by making sure that certain parameters are equal, as in Figure 4.1b. In this section, we demonstrate how this fact can be obtained using the presented results. To this end, let a SPN, $\mathcal{S}$, having the same structure as in Figure 4.1b, and assume we would like to enforce the constraint $X_1 \perp X_2$. We will now show how the presented results naturally lead to the realization that this corresponds to an equality between parameters in blue, as well as between those in red. Looking at the SPN, it is not difficult to see that the implied factorization is:

$$\mathcal{S}(X_1, X_2, X_3, \neg X_1, \neg X_2, \neg X_3) = w_{11} X_1 \cdot (w_{21} X_2 \cdot (w_{31} X_3 + w_{32} \neg X_3) + w_{22} \neg X_2 \cdot (w_{33} X_3 + w_{34} \neg X_3)) \\ + w_{12} \neg X_1 \cdot (w_{23} X_2 \cdot (w_{35} X_3 + w_{36} \neg X_3) + w_{24} \neg X_2 \cdot (w_{37} X_3 + w_{38} \neg X_3))$$

Using this notation, we would like to prove that $w_{21} = w_{23}$ and that $w_{22} = w_{24}$. To this end, using Theorem 4.3, we need to enforce the conditions:

$$\begin{aligned} &\Pr(X_1, X_2) = \Pr(X_1) \cdot \Pr(X_2), \;\; \Pr(X_1, \neg X_2) = \Pr(X_1) \cdot \Pr(\neg X_2), \\ &\Pr(\neg X_1, X_2) = \Pr(\neg X_1) \cdot \Pr(X_2), \;\; \Pr(\neg X_1, \neg X_2) = \Pr(\neg X_1) \cdot \Pr(\neg X_2) \end{aligned} \tag{4.1}$$

All of these probabilities can be computed using the SPN, since for example, $\Pr(X_1, X_2) = \mathcal{S}(1, 1, 1, 0, 0, 1)$. Rewriting the equations in (4.1) in terms of the SPN's parameters yields the following:

**Algorithm 3** Training with soft constraints

**Require:** SPN structure $S$, dataset $D$, constraints $C_n$, hyperparameters $\alpha_n$, learning rate $\gamma$
Initialize SPN parameters, $\mathbf{w}$
**while** convergence criterion is not satisfied **do**
  Sample a $d \in D$
  $\mathbf{w}^{(k)} \leftarrow \mathbf{w}^{(k-1)} + \gamma(\nabla_{\mathbf{w}} \mathrm{S}(d) + \sum_n \alpha_n \nabla_{\mathbf{w}} C_n)$
  *NormalizeWeights(S)*
**end while**
return $S$

$$\begin{cases} w_{11}w_{21} = w_{11}(w_{11}w_{21} + w_{12}w_{23}) \\ w_{12}w_{23} = w_{12}(w_{11}w_{21} + w_{12}w_{23}) \\ w_{11}w_{22} = w_{11}(w_{11}w_{22} + w_{12}w_{24}) \\ w_{12}w_{24} = w_{12}(w_{11}w_{22} + w_{12}w_{24}) \end{cases} \Leftrightarrow \begin{cases} \dfrac{w_{11}w_{21}}{w_{12}w_{23}} = \dfrac{w_{11}(w_{11}w_{21} + w_{12}w_{23})}{w_{12}(w_{11}w_{21} + w_{12}w_{23})} \\ \dfrac{w_{11}w_{22}}{w_{12}w_{24}} = \dfrac{w_{11}(w_{11}w_{22} + w_{12}w_{24})}{w_{12}(w_{11}w_{22} + w_{12}w_{24})} \end{cases} \Leftrightarrow \begin{cases} \dfrac{w_{21}}{w_{23}} = 1 \\ \dfrac{w_{22}}{w_{24}} = 1 \end{cases}$$

The final equations clearly rediscover that the conditions $w_{21} = w_{23}$ and $w_{22} = w_{24}$ must hold, meaning there are infinitely many solutions, exactly as expected following the discussion in Section 4.4.

## 4.6 Optimization framework

The previous sections introduced the connection between the network polynomial of a SPN and various probabilistic queries. In this section we are going to discuss how to utilize these insights in order to train SPNs that incorporate probabilistic constraints. Many works (Gens and Domingos, 2013; Adel et al., 2015; Rooshenas and Lowd, 2014) attempt to learn both the structure and the parameters of a SPN, where the tuning of the parameters' values is usually achieved using a heuristic, such as the proportion of the training instances in a sum node. However, as noted in (Zhao et al., 2016), first learning the structure, using some of the aforementioned approaches, and then fitting the parameters, yields better results. In what follows we are going to follow the latter approach, assuming that the SPN structure is known, since the focus of this section is on the model's parameters.

As seen in Section 4.2.1, incorporating soft constraints is equivalent to adding new terms in the objective function. In our case, all of these terms are differentiable, since they are polynomials, so any standard optimization algorithm could be utilized to train the model. Algorithm 3 describes a pipeline for carrying out this procedure. Apart from including the extra terms in the objective function, it also allows for a hyperparameter, $\alpha$, so it is possible to adjust the relative importance of each constraint. Finally, since it might be the case that after each iteration the parameters of a sum node are not normalized (meaning they do not sum to 1), after each update

**Algorithm 4** Training with hard constraints

**Require:** SPN structure $S$, dataset $D$, constraints $C_n$, learning rate $\gamma$
Initialize SPN parameters, $\mathbf{w}$
**while** convergence criterion is not satisfied **do**
  Sample a $d \in D$
  $\mathbf{w}^{(k)} \leftarrow \mathbf{P}_{C_1,\ldots,C_n}(\mathbf{w}^{(k-1)} + \gamma \nabla_{\mathbf{w}} \mathrm{S}(d))$
  *NormalizeWeights(S)*
**end while**
return $S$

the SPN has to be normalized, for example following the approach described in (Peharz et al., 2015).

In contrast, if they are treated as hard constraints, projected gradient descent or approaches like the one developed in (Marquez Neila et al., 2017) should be used. Algorithm 4 is a variation of Algorithm 3, adapted to this case. The modification lies on the fact that after the weights are updated, then they are projected on the space defined by the constraints, using the $\mathbf{P}_{C_1,\ldots,C_n}(\cdot)$ operator, see (Marquez Neila et al., 2017; Zhao et al., 2016) for different projection techniques and their effect on the resulting solutions. All of the required constraints can be uncovered in analytical form, following the steps outlined in the proofs.

## 4.7 Conclusions

In this chapter, we provided proofs regarding the theoretical feasibility of incorporating various kinds of constraints to SPNs. By taking advantage of the network polynomial and the SPNs' multilinear structure, we were able to show that enforcing each of the considered constraint is equivalent to the parameters of the SPN satisfying a system of equations. Furthermore, we discussed two different optimization-based approaches to integrate the constraints to a SPN learned from data.

The obtained results demonstrate that SPNs possess some transparent features even in the general case, and are complimentary to the ones presented in Chapter 3. This opens the door for employing SPNs in applications where background information is available or when certain probabilistic assumptions need to be satisfied by the resulting model.

From here, there is a number of promising directions regarding future research. In this chapter, we only consider probabilistic constraints, so extending the results to account for propositional ones seems like a natural next step. In fact, PSDDs satisfy this property by construction, while also being tractable. This means that progress related to the feasibility of imposing probabilistic constraints to PSDDs, may lead to models that combine tractable inference with the flexibility to incorporate a wide range of a priori assumptions. Since PSDDs have a multilinear structure, just like SPNs, a variation of the techniques used in the presented proofs, could provide the first step in tackling this question.

Another promising direction comes from exploring ways to incorporate inequality constraints, for example that a certain configuration should have a higher likelihood than others. This task can be efficiently performed in the BN case (De Campos et al., 2009), but it remains an open question whether this is the case for SPNs as well. As with the equality constraints that were presented in this chapter, it is not a matter of immediately applying the same reasoning used in the BN case, except for when the conditions in Chapter 3 hold, however, taking advantage of the network polynomial in a similar way as in this chapter, could again facilitate proving the desired result.

## 4.8 Proofs

**Proof of Theorem 4.1**

Let $\mathcal{S}(\mathbf{x}) = \sum_{\mathbf{x}} f(\mathbf{x}) \prod_{n=1}^{N} \mathbf{1}_{x_n}$ be the canonical network polynomial of an SPN. The given conditional independence relationship corresponds to the equality $\Pr(X_i, X_j | X_k) = \Pr(X_i | X_k) \cdot \Pr(X_j | X_k)$, which can further be rewritten as follows:

$$\Pr(X_i, X_j | X_k) = \Pr(X_i | X_k) \cdot \Pr(X_j | X_k) \implies \frac{\Pr(X_i, X_j, X_k)}{\Pr(X_k)} = \frac{\Pr(X_i, X_k)}{\Pr(X_k)} \cdot \frac{\Pr(X_j, X_k)}{\Pr(X_k)} \tag{4.2}$$

$$\implies \Pr(X_i, X_j, X_k) \cdot \Pr(X_k) = \Pr(X_i, X_k) \cdot \Pr(X_j, X_k)$$

Next, we express the above probabilities in terms of $\mathcal{S}$ (where $X$ corresponds to the assignment $X = 1$, and $\neg X$ to $X = 0$):

$$\begin{aligned}
\Pr(X_i, X_j, X_k) = &\sum_{\mathbf{x}:x_i,x_j,x_k} f(\mathbf{x})\mathbf{1}_{x_i}\mathbf{1}_{x_j}\mathbf{1}_{x_k} + \sum_{\mathbf{x}:\neg x_i,x_j,x_k} f(\mathbf{x})\mathbf{1}_{\neg x_i}\mathbf{1}_{x_j}\mathbf{1}_{x_k} + \sum_{\mathbf{x}:x_i,\neg x_j,x_k} f(\mathbf{x})\mathbf{1}_{x_i}\mathbf{1}_{\neg x_j}\mathbf{1}_{x_k} \\
&+ \sum_{\mathbf{x}:x_i,x_j,\neg x_k} f(\mathbf{x})\mathbf{1}_{x_i}\mathbf{1}_{x_j}\mathbf{1}_{\neg x_k} + \sum_{\mathbf{x}:\neg x_i,\neg x_j,x_k} f(\mathbf{x})\mathbf{1}_{\neg x_i}\mathbf{1}_{\neg x_j}\mathbf{1}_{x_k} + \sum_{\mathbf{x}:\neg x_i,x_j,\neg x_k} f(\mathbf{x})\mathbf{1}_{\neg x_i}\mathbf{1}_{x_j}\mathbf{1}_{\neg x_k} \\
&+ \sum_{\mathbf{x}:x_i,\neg x_j,\neg x_k} f(\mathbf{x})\mathbf{1}_{x_i}\mathbf{1}_{\neg x_j}\mathbf{1}_{\neg x_k} + \sum_{\mathbf{x}:\neg x_i,\neg x_j,\neg x_k} f(\mathbf{x})\mathbf{1}_{\neg x_i}\mathbf{1}_{\neg x_j}\mathbf{1}_{\neg x_k}
\end{aligned}$$

$$\begin{aligned}
\Pr(X_i, X_k) &= \sum_{\mathbf{x}:x_i,x_k} f(\mathbf{x})\mathbf{1}_{x_i}\mathbf{1}_{x_k} + \sum_{\mathbf{x}:\neg x_i,x_k} f(\mathbf{x})\mathbf{1}_{\neg x_i}\mathbf{1}_{x_k} + \sum_{\mathbf{x}:x_i,\neg x_k} f(\mathbf{x})\mathbf{1}_{x_i}\mathbf{1}_{\neg x_k} + \sum_{\mathbf{x}:\neg x_i,\neg x_k} f(\mathbf{x})\mathbf{1}_{\neg x_i}\mathbf{1}_{\neg x_k} \\
\Pr(X_j, X_k) &= \sum_{\mathbf{x}:x_j,x_k} f(\mathbf{x})\mathbf{1}_{x_j}\mathbf{1}_{x_k} + \sum_{\mathbf{x}:\neg x_j,x_k} f(\mathbf{x})\mathbf{1}_{\neg x_j}\mathbf{1}_{x_k} + \sum_{\mathbf{x}:x_j,\neg x_k} f(\mathbf{x})\mathbf{1}_{x_j}\mathbf{1}_{\neg x_k} + \sum_{\mathbf{x}:\neg x_j,\neg x_k} f(\mathbf{x})\mathbf{1}_{\neg x_j}\mathbf{1}_{\neg x_k} \\
\Pr(X_k) &= \sum_{\mathbf{x}:x_k} f(\mathbf{x})\mathbf{1}_{x_k} + \sum_{\mathbf{x}:\neg x_k} f(\mathbf{x})\mathbf{1}_{\neg x_k}
\end{aligned}$$

The next step is to substitute these quantities into (4.2) and carry out the computations. In doing so, products of the form $\mathbf{1}_{x_k} \cdot \mathbf{1}_{\neg x_k}$, as well as $\mathbf{1}_{x_k} \cdot \mathbf{1}_{x_k}$, will arise. To deal with the former, it is enough to observe that the product of indicators corresponding to different states is always equal to zero, since a variable cannot be at two states at the same time, so $\mathbf{1}_{x_k} \cdot \mathbf{1}_{\neg x_k} = 0$. For the latter, since $\mathbf{1}_{x_k} \in \{0, 1\}$, and $0^2 = 0, 1^2 = 1$, it follows that $\mathbf{1}_{x_k} \cdot \mathbf{1}_{x_k} = \mathbf{1}_{x_k}$. With this in mind, substituting to (4.2) and equating the coefficients of same terms leads to the following system of equations:

$$\sum_{\mathbf{x}:x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:x_i,x_j,x_k} f(\mathbf{x}) = \sum_{\mathbf{x}:x_i,x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:x_j,x_k} f(\mathbf{x})$$

$$\sum_{\mathbf{x}:x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:\neg x_i,x_j,x_k} f(\mathbf{x}) = \sum_{\mathbf{x}:\neg x_i,x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:x_j,x_k} f(\mathbf{x})$$
$$\sum_{\mathbf{x}:x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:x_i,\neg x_j,x_k} f(\mathbf{x}) = \sum_{\mathbf{x}:x_i,x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:\neg x_j,x_k} f(\mathbf{x})$$
$$\sum_{\mathbf{x}:x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:\neg x_i,\neg x_j,x_k} f(\mathbf{x}) = \sum_{\mathbf{x}:\neg x_i,x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:\neg x_j,x_k} f(\mathbf{x})$$
$$\sum_{\mathbf{x}:\neg x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:x_i,x_j,\neg x_k} f(\mathbf{x}) = \sum_{\mathbf{x}:x_i,\neg x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:x_j,\neg x_k} f(\mathbf{x})$$
$$\sum_{\mathbf{x}:\neg x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:\neg x_i,x_j,\neg x_k} f(\mathbf{x}) = \sum_{\mathbf{x}:\neg x_i,\neg x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:x_j,\neg x_k} f(\mathbf{x})$$
$$\sum_{\mathbf{x}:\neg x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:x_i,\neg x_j,\neg x_k} f(\mathbf{x}) = \sum_{\mathbf{x}:x_i,\neg x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:\neg x_j,\neg x_k} f(\mathbf{x})$$
$$\sum_{\mathbf{x}:\neg x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:\neg x_i,\neg x_j,\neg x_k} f(\mathbf{x}) = \sum_{\mathbf{x}:\neg x_i,\neg x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:\neg x_j,\neg x_k} f(\mathbf{x})$$

Each of these equations correspond to a quadratic multivariate polynomial, since there are terms that are contained in both factors of each product. For example, the first equation has the product $\sum_{\mathbf{x}:x_k} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:x_i,x_j,x_k} f(\mathbf{x})$. It is not difficult to see that states having $X_i = 1, X_j = 1, X_k = 1$ are compatible with both sums, so the corresponding terms will appear in both of them, meaning that when multiplying the two sums, these terms will be squared. □

Looking at the proof, the equation in (4.2) induces a system with as many equations as the number of possible configurations of $X_i, X_j, X_k$. Since all of them were assumed to be binary, there are $2 \cdot 2 \cdot 2 = 8$ such configurations, exactly as many as the equations in the system. In the general case, where $X_i \in \{0, 1, \cdots, i-1\}, X_j \in \{0, 1, \cdots, j-1\}$ and $X_k \in \{0, 1, \cdots, k-1\}$, the resulting system contains $i \cdot j \cdot k$ equations, meaning that the number of equations scale linearly with respect to the range of each variables.

**Proof of Theorem 4.2**

The proof follows the same reasoning as before, so the first step is to rewrite the given constraint:

$$\begin{aligned}
&\Pr(\mathbf{X}_{-i}|do(X_i = 0)) = \Pr(\mathbf{X}_{-i}|do(X_i = 1)) \\
&\Rightarrow \frac{\Pr(\mathbf{X}_{-i}, X_i = 0)}{\Pr(X_i = 0|pa_{X_i})} = \frac{\Pr(\mathbf{X}_{-i}, X_i = 1)}{\Pr(X_i = 1|pa_{X_i})} \\
&\Rightarrow \Pr(\mathbf{X}_{-i}, X_i = 0) \cdot \Pr(X_i = 1|pa_{X_i}) = \Pr(\mathbf{X}_{-i}, X_i = 1) \cdot \Pr(X_i = 0|pa_{X_i}) \\
&\Rightarrow \Pr(\mathbf{X}_{-i}, X_i = 0) \cdot \Pr(X_i = 1, pa_{X_i}) = \Pr(\mathbf{X}_{-i}, X_i = 1) \cdot \Pr(X_i = 0, pa_{X_i})
\end{aligned}$$

The next step is to express these probabilities in terms of the network polynomial and substitute them to the above expression. These computations are repetitive and closely resemble those in the proof of Theorem 4.1, which is why we opted for omitting them. The important observation is that the result is a system of multivariate polynomials, in this case, too. To prove they are linear ones as well, it suffices to note that in both products $\Pr(\mathbf{X}_{-i}, X_i = 0) \cdot \Pr(X_i = 1, pa_{X_i})$

and $\Pr(\mathbf{X}_{-i}, X_i = 1) \cdot \Pr(X_i = 0, pa_{X_i})$, the set of parameters involved in the first factor is disjoint with the one appearing in the second factor, since the parameters that remain after setting $X_i = 0$ vanish when setting $X_i = 1$ (and vice versa). □

Following the discussion about extending the binary case to the general case, after Theorem 4.1, it should not be surprising that a similar argument holds true for Theorem 4.2 as well. The main observation is that enforcing an interventional constraint, for the non-binary case, $X_i \in \{0, 1, \cdots, i-1\}$, can be transformed into the following series of pairwise equalities: $\Pr(\mathbf{X}_{-i}|do(X_i = 0) = \Pr(\mathbf{X}_{-i}|do(X_i = 1), \Pr(\mathbf{X}_{-i}|do(X_i = 1) = \Pr(\mathbf{X}_{-i}|do(X_i = 2), \ldots, \Pr(\mathbf{X}_{-i}|do(X_i = i-1) = \Pr(\mathbf{X}_{-i}|do(X_i = i))$. Each equation involves $\mathbf{X}_{-i}$ and only two states of $X_i$, so (a slight modification of) Theorem 4.2 applies to each of them, meaning that each of these equalities induces a system of 2 equations. Putting everything together, the overall number of equations in the system is $2(i-1)$.

**Proof of Theorem 4.3**

To prove this result, again, the first observation is that the given constraint corresponds to the equality $\Pr(X_i, X_j) = \Pr(X_i) \cdot \Pr(X_j)$. Following that, all the probabilities involved can be expressed in terms of the network polynomial, $S$:

$$\Pr(X_i, X_j) = \sum_{\mathbf{x}:x_i,x_j} f(\mathbf{x})\mathbf{1}_{x_i}\mathbf{1}_{x_j} + \sum_{\mathbf{x}:\neg x_i,x_j} f(\mathbf{x})\mathbf{1}_{\neg x_i}\mathbf{1}_{x_j} + \sum_{\mathbf{x}:x_i,\neg x_j} f(\mathbf{x})\mathbf{1}_{x_i}\mathbf{1}_{\neg x_j} + \sum_{\mathbf{x}:\neg x_i,\neg x_j} f(\mathbf{x})\mathbf{1}_{\neg x_i}\mathbf{1}_{\neg x_j}$$
$$\Pr(X_i) = \sum_{\mathbf{x}:x_i} f(\mathbf{x})\mathbf{1}_{x_i} + \sum_{\mathbf{x}:\neg x_i} f(\mathbf{x})\mathbf{1}_{\neg x_i}$$
$$\Pr(X_j) = \sum_{\mathbf{x}:x_j} f(\mathbf{x})\mathbf{1}_{x_j} + \sum_{\mathbf{x}:\neg x_j} f(\mathbf{x})\mathbf{1}_{\neg x_j}$$

Next, we substitute these quantities to the constraint's equation, so we get that:

$$\begin{aligned}
&\sum_{\mathbf{x}:x_i,x_j} f(\mathbf{x})\mathbf{1}_{x_i}\mathbf{1}_{x_j} + \sum_{\mathbf{x}:\neg x_i,x_j} f(\mathbf{x})\mathbf{1}_{\neg x_i}\mathbf{1}_{x_j} + \sum_{\mathbf{x}:x_i,\neg x_j} f(\mathbf{x})\mathbf{1}_{x_i}\mathbf{1}_{\neg x_j} \\
&+ \sum_{\mathbf{x}:\neg x_i,\neg x_j} f(\mathbf{x})\mathbf{1}_{\neg x_i}\mathbf{1}_{\neg x_j} = \sum_{\mathbf{x}:x_i} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:x_j} f(\mathbf{x})\mathbf{1}_{x_i}\mathbf{1}_{x_j} \\
&+ \sum_{\mathbf{x}:x_i} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:\neg x_j} f(\mathbf{x})\mathbf{1}_{x_i}\mathbf{1}_{\neg x_j} + \sum_{\mathbf{x}:\neg x_i} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:x_j} f(\mathbf{x})\mathbf{1}_{\neg x_i}\mathbf{1}_{x_j} \\
&+ \sum_{\mathbf{x}:\neg x_i} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:\neg x_j} f(\mathbf{x})\mathbf{1}_{\neg x_i}\mathbf{1}_{\neg x_j}
\end{aligned}$$

Equating the coefficients we get the following system of equations:

$$\begin{aligned}
\sum_{\mathbf{x}:x_i,x_j} f(\mathbf{x}) &= \sum_{\mathbf{x}:x_i} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:x_j} f(\mathbf{x}) \\
\sum_{\mathbf{x}:\neg x_i,x_j} f(\mathbf{x}) &= \sum_{\mathbf{x}:\neg x_i} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:x_j} f(\mathbf{x}) \\
\sum_{\mathbf{x}:x_i,\neg x_j} f(\mathbf{x}) &= \sum_{\mathbf{x}:x_i} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:\neg x_j} f(\mathbf{x})
\end{aligned}$$

$$\sum_{\mathbf{x}:\neg x_i,\neg x_j} f(\mathbf{x}) = \sum_{\mathbf{x}:\neg x_i} f(\mathbf{x}) \cdot \sum_{\mathbf{x}:\neg x_j} f(\mathbf{x})$$

Each of these equations correspond to a quadratic multivariate polynomial, since in each equation the sums appearing on the right hand side have some terms in common. For example, looking at the first equation, the assignment setting all the variables equal to 1 is compatible with both summations, so the term $f(x_1, \cdots, x_n)$ appears in both of them. Clearly, by multiplying them we end up with a squared parameter. □

Theorem 4.3 can be extended to the general case, as well, following the same arguments as before. Assuming that $X_i \in \{0, 1, \cdots, m-1\},\ X_j \in \{0, 1, \cdots, k-1\}$, then enforcing an independence constraint results in a multivariate quadratic system of $m \cdot k$ equations.

At this point, looking at the proofs it might appear like the proposed framework can only be applied when having a non-factorized, canonical, polynomial representation. However, this is not the case, since upon a closer inspection, the only essential requirement is to be able to compute all the probabilities utilized in the proofs. SPNs can compute these probabilities regardless of whether they encode the canonical polynomial or one of its factorized representations. Of course, the more compact a SPN is, the more efficient is inference utilizing it, but it is always possible to infer these probabilities. After obtaining all the necessary quantities (each one requiring a single pass over the SPN), they can be substituted into the corresponding equations. Although the proofs are based on the canonical polynomial, in practical applications any of its factorized representations can be used, leading again to a system of equations, which is expressed entirely in terms of the SPN's parameters.

This reasoning applies to all the considered constraints, not only the independence ones, since the only essential requirement is to be able to substitute the probabilistic quantities appearing in a constraint with the corresponding SPN outcomes, thus obtaining a system involving only the SPN's parameters. As discussed in the preceding paragraph, all the necessary probabilities can be inferred when using a SPN. For example, enforcing a conditional constraint of the form $\Pr(X_1|X_2) = \Pr(X_1|\neg X_2)$, requires computing $\Pr(X_1, X_2), \Pr(X_1, \neg X_2), \Pr(X_2), \Pr(\neg X_2)$ (using a SPN) and substituting them into the equation
$\Pr(X_1, X_2) \cdot \Pr(\neg X_2) = \Pr(X_1, \neg X_2) \cdot \Pr(X_2)$, leading to an equation involving only the SPN parameters.

# Chapter 5

# Counterfactuals in SPNs

## 5.1 Introduction

In recent years explanations for machine learning (ML) models have gained a lot of prominence, especially in the context of safety critical applications. This is due to the black-box nature of many of the state-of-the-art models, which impedes a thorough understanding of their internal reasoning. One of the most important challenges is how can one be sure that the algorithm bases its decisions on the proper criteria, or that it does not discriminate against certain minority groups?

Bayesian networks (BNs) have been traditionally deployed in applications where such considerations are crucial, due to their ability to clearly represent relationships between variables, and incorporate causal information. One of the most celebrated properties of BNs is their ability to compute counterfactual quantities of the from "what would have been the value of $Y$, had $X$ been equal to $x$?", which has been extensively utilized in high-stakes applications, such as in finance (Dhar, 1998; Fatum and Hutchison, 2010), healthcare (Prosperi et al., 2020; VanderWeele, 2020), and criminal justice (Mishler et al., 2021; Sampson et al., 2006).

Having said that, one of the challenges of computing counterfactual quantities is that they require a very careful specification of the mechanisms that underlie the interactions of all the variables included in the model. This is a highly non-trivial task, usually involving domain-expert knowledge as well as hand-crafting the final models. However, ML models often involve a prohibitively high number of variables to allow for accurately specifying every interaction. Complications like this, have led to the emergence of a related line of research within the field of explainability in AI (XAI), where the objective is to identify simplified "computational" counterfactuals. The term computational reflects the underlying idea of this approach, where a classification model is treated as a function, and, given an instance, $X$, the objective is to find a counterfactual instance, $Y$, such that $X$ and $Y$ are as close as possible, but the model predicts a different class for each of them.

This line of research has led to the development of a general framework for producing counterfactual instances for any differentiable classification model, as described in (Wachter et al., 2018). Building on top of that, the authors in (Mothilal et al., 2020) proposed a method for generating diverse counterfactuals for differentiable models, while the work in (Russell, 2019) addressed some technical challenges, proposing a new framework that is based on mixed integer programming (MIP).

In this chapter, we extend the framework in (Russell, 2019) to the non-linear case, so it allows for generating computational counterfactuals for multilinear models. This model class includes Bayesian network classifiers (BNCs) (Friedman et al., 1997), as well as decision trees, and random forests. In order to address the BNC case we utilize discriminative SPNs, since they subsume all BNCs, allowing for studying the latter under a unified framework. More specifically, we present the following contributions:

- We show that by taking advantage of a model's multilinear structure, it is possible to formulate an integer linear program (ILP) that is guaranteed to generate valid counterfactuals.
- We demonstrate how one can seamlessly generate diverse counterfactuals using the presented framework.
- We demonstrate how to apply it to decision trees (DTs), random forests (RFs), as well as discriminative SPNs, possibly resulting into an infinite set of counterfactuals.
- We show that the presented framework generalizes other existing frameworks, while we also discuss how it can be easily adjusted to generate alternative forms of explanations.

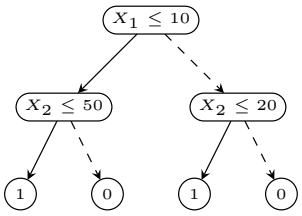
## 5.2 Background

In this section we are going to briefly introduce DTs, RFs, and discriminative SPNs, as these models are going to be used in the remaining of this chapter.

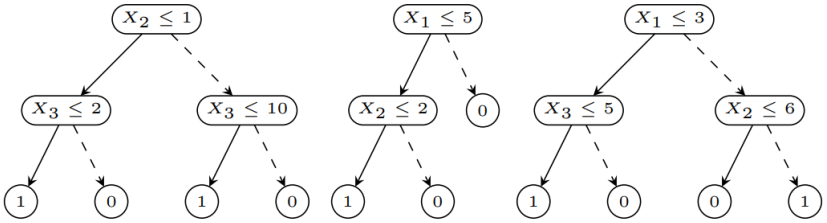
### 5.2.1 Decision Trees

Decision trees (DTs) are tree-like structures that contain a set of conditional control statements, such as $X \leq a$. Each assignment is consistent with exactly one root-to-leaf path, while control statements are arranged in a hierarchical manner, where intermediate nodes represent decisions and leaf nodes can be either class labels (for classification problems) or continuous quantities (for regression problems).

The majority of decision tree learning algorithms operate in a top-down manner, iteratively partitioning the whole dataset into smaller ones, and conditioning on the values of the feature




(a) A Decision Tree




(b) A Random Forest

Figure 5.1: Examples of decision trees and random forests. Solid lines are followed if the corresponding rule is satisfied, dashed lines if it is not.

that contains the most information. This has led to the development of a number of metrics that quantify the amount of information that is gained when splitting the dataset according to a specific feature, such as Gini impurity (Bishop, 2006). In turn, these metrics can be used in order to design algorithms that learn DTs from data, such as CART (Moore II, 1987).

An advantage of employing DTs is that their internal rule-based architecture is relatively easy to inspect, allowing for assessing the quality of the model. This is one of the major reasons DTs are usually utilized in cases where the model's understandability is essential. However, large DTs containing a lot of rules are not easy to interpret anymore, requiring additional explainability tools in order to reason about their internal behaviour (Arrieta et al., 2020).

### 5.2.2 Random Forests

DTs have been employed in various applications due to the transparency they exhibit, at least as long as they are kept at a reasonable size. However, one of their major limitations is their tendency to overfit a dataset, leading to high variance models that fail to maintain good performance when dealing with new data.

Random forests (RFs) aim at overcoming this challenge by combining multiple trees, resulting in more stable models with lower variance. The main insight underlying this approach is to sample with replacement from the training dataset in order to construct multiple new datasets, thus implementing the idea of bagging (Breiman, 1996). Following that, a decision tree is trained over each of these newly acquired datasets, leading to an ensemble of independent trees. Then, in prediction time, an aggregation measure, such as majority voting (for classification) or averaging (for regression), combines the predictions of each tree in order to generate the prediction of the whole forest.


(a) BNC (b) DSPN

Figure 5.2: BNC and DSPN representations of the same Naive Bayes model.

The procedure described above results in very expressive and accurate models, however this comes at the expense of interpretability, since the whole forest is far more challenging to explain, compared to single decision trees. This has led to the development of various techniques that attempt to explain the inner reasoning of a RF (Arrieta et al., 2020).

### 5.2.3 Discriminative Sum-Product Networks

Discriminative sum-product networks (DSPNs) are rooted directed graphical models, similar to the generative SPNs considered in the previous chapters. The only difference between the two is that DSPNs represent the conditional distribution of a variable, while generative SPNs represent the full joint distribution. Moreover, DSPNs enable tractable inference in the same manner as generative SPNs, by enforcing completeness and decomposability. DSPNs are preferred in classification problems, since although it is feasible to learn the full joint distribution using a generative SPN, and then employ it to compute any conditional distribution, directly learning the desired conditional distribution yields better results (Lafferty et al., 2001). This is because the joint distribution implicitly encodes all the corresponding conditionals, so it is a much more challenging task, compared to focusing on learning just a single conditional distribution. Furthermore, any BN representing the conditional distribution of a discrete variable (i.e. any BNC) can be represented as a DSPN, similarly to how any BN representing the joint distribution can be transformed into a generative SPN. Figure 5.2 depicts two equivalent representations of a naive Bayes model, made for the conditional distribution of $Y|X_1, X_2, X_3$, one as a BNC (Figure 5.2a), and one as a DSPN (Figure 5.2b).

## 5.3 Problem Derivation

This section introduces our approach for generating counterfactuals, inspired by (Russell, 2019), but addressing one of its key limitations; the range of models it applies to. Specifically, we extend the existing framework to multilinear models, such as DTs and BNCs, as well as ensembles thereof that utilize majority voting, such as RFs. In what follows all variables are

assumed to be binary, to allow for an easier presentation, however, in Section 5.3.4 an extension to the non-binary case is provided.

Before going any further, we define a quantity similar to the decision function, developed in (Shih et al., 2018), as follows:

**Definition 5.1.** Let $G : \mathbf{X} \to \{0,1\}$ be a binary classification function, and $P_0^G(\mathbf{X}), P_1^G(\mathbf{X})$ be multilinear polynomials of indicator variables, where all coefficient are equal to 1 and there is no constant term. Then $P_0^G(\mathbf{X}) \in \{0,1\}$ (respectively, $P_1^G(\mathbf{X})$) is called the 0-decision (resp. 1-decision) polynomial of $G$, iff $G(\mathbf{X}) = 0 \Leftrightarrow P_0^G(\mathbf{X}) = 1$ (resp. $G(\mathbf{X}) = 1 \Leftrightarrow P_1^G(\mathbf{X}) = 1$)

Decision polynomials provide for a multilinear representation of arbitrary binary classifiers. In (Shih et al., 2018), decision functions play a similar role, but there is no requirement for them to be multilinear. However, this additional condition is required in order to be able to derive an optimization problem in ILP format.

In the remaining of this section, some general results regarding decision polynomials are derived, which are going to provide the basis of the subsequent algorithms. The next proposition follows immediately from the definition, and will be used extensively throughout the rest:

**Proposition 5.2.** *Let $G : \boldsymbol{X} \to \{0,1\}$, and $P_0^G(\boldsymbol{X}), P_1^G(\boldsymbol{X})$ be the decision polynomials. Then $\forall \boldsymbol{x} \in \boldsymbol{X}$ $P_0^G(\boldsymbol{x}) + P_1^G(\boldsymbol{x}) = 1$*

The following statement is a simple observation that since each term of a decision polynomial is equal to either 0 or 1, in order for the polynomial to output 0, each term has to be equal to 0.

**Proposition 5.3.** *Let $G : \boldsymbol{X} \to \{0,1\}$, and $P_0^G(\boldsymbol{X}), P_1^G(\boldsymbol{X})$ be the decision polynomials. Assuming $P_0^G(\boldsymbol{X}) = T_1(\boldsymbol{X}) + T_2(\boldsymbol{X}) + \cdots + T_k(\boldsymbol{X})$, where each $T_i \in \{0,1\}$, then $P_0^G(\boldsymbol{X}) = 0 \Rightarrow T_1(\boldsymbol{X}) = T_2(\boldsymbol{X}) = \cdots = T_k(\boldsymbol{X}) = 0$. The same holds for $P_1^G(\boldsymbol{X})$.*

Proposition 5.3 implies that in order to make sure that a decision polynomial outputs 0, it is enough to make sure that each monomial equals 0. The next challenge is due to the fact that these monomials are products of indicator functions, not linear combinations thereof, which impedes us from formulating a linear optimization problem. A key insight for overcoming this difficulty is that since indicator functions can be equal to either 0 or 1, making sure that not all of them are equal to 1 is sufficient to guarantee that their product is equal to 0. The following proposition states a simple condition that leads to this outcome.

**Proposition 5.4.** *Let $X_1, X_2, \cdots X_k \in \{0,1\}$, then the following hold:*
$X_1 \cdot X_2 \cdots X_{k-1} \cdot X_k = 1 \Leftrightarrow X_1 + X_2 + \cdots + X_k = k$ *and*
$X_1 \cdot X_2 \cdots X_{k-1} \cdot X_k = 0 \Leftrightarrow X_1 + X_2 + \cdots + X_k \leq k - 1$.

At this point, Propositions 5.3 and 5.4 already provide for a set of constraints that are sufficient to ensure that a datapoint is classified as either 0 or 1. For example, if the goal is to generate an

**Algorithm 5** Counterfactuals for multilinear models

**Require:** A multilinear model $M$
Compile the $k$-DP of $M$, where $k \in \{0,1\}$
Form the objective function
**if** the counterfactual has to be classified in the $k$-class **then**
Add the constraints in Proposition 5.5 to the problem
**else**
Apply the inequality constraints in Proposition 5.4 to every term of the DP
**end if**

instance that belongs into the 1-class, then it is enough to consider the 0-decision polynomial and force it to output 0. This can be done by considering each of its terms, say $X_1 \cdot X_2 \cdots X_{k-1} \cdot X_k$, and adding the constraint $X_1 + X_2 + \cdots + X_k \leq k-1$. This procedure guarantees that all the terms will be equal to 0, so the solution to the problem, $\mathbf{X}$, satisfies $P_0^G(\mathbf{X}) = 0 \Rightarrow P_1^G(\mathbf{X}) = 1$, i.e. it is classified as 1.

However, having said that, storing both polynomials requires additional resources, while it could also be the case that one of them is significantly smaller than the other one, so it would be preferable to express the problem in terms of this polynomial to end up with a more compact optimization problem. A natural way to address this situation would be perform the complementary task of the one considered so far; forcing a decision polynomial to be equal to 1. For example, this could be achieved by defining a set of constraints that when satisfied force a term in the decision polynomial to be equal to 1, and the rest equal to 0. As a matter of fact, in the following proposition we show that it is indeed possible to uncover such a set of constraints:

**Proposition 5.5.** *Let $P_0(\boldsymbol{X}) = X_{11} \cdots X_{1k} + X_{21} \cdots X_{2m} + \cdots + X_{n1} \cdots X_{nl}$, where each $X_i \in \{0,1\}$, be the 0-decision polynomial of a model. Furthermore, let the constraints $X_{11} + X_{12} \cdots + X_{1k} \geq k \cdot \delta_1, X_{21} + X_{22} \cdots + X_{2m} \geq m \cdot \delta_2, \ldots, X_{n1} + X_{n2} \cdots + X_{nl} \geq l \cdot \delta_n, \sum_{i=1}^{n} \delta_i = 1$, where $\delta_i \in \{0,1\}$. If an assignment, $\boldsymbol{X}'$, satisfies these constraints, then $P_0(\boldsymbol{X}') = 1$. An analogous statement holds for $P_1(\boldsymbol{X})$.*

We have now developed most of the the necessary machinery to formulate a counterfactual generating optimization problem, shown in Algorithm 5. In the following subsections we address the first two points in Algorithm 5, providing ways to recover the DPs of DTs, RFs and BNCs, as well as discussing how to set the weights of the weighted $l_1$ norm, which is going to serve as our objective function. Furthermore we provide some adjustments that need to be made in order to take into account the characteristics of the aforementioned models.

### 5.3.1 Decision trees

In this section we are going to examine how to use the branching rules in a DT to generate a decision polynomial. To this end, we make use of the fact that any DT can be transformed into

an equivalent rule-based classifier (Quinlan, 1987), which allow us to derive a multilinear representation of a DT over the set of rules it naturally induces.

An example of the general process can be seen in Figure 5.1a, which contains a very simple decision tree. It is defined over two continuous variables, $X_1, X_2$, but it can also be seen as a function over its internal rules, $X_1 \leq 10, X_2 \leq 50, X_2 \leq 20$. To construct the 0-DP, we traverse the DT bottom-up and we include a term for each path that ends in a 0 leaf. Whenever a rule, $R$, in a path is satisfied, we make an indicator $\mathbf{1}[R]$. On the other hand, if a rule is not satisfied, we form the difference $1 - \mathbf{1}[R]$. Finally, the term is constructed by considering all rules in the path and multiplying the aforementioned quantities. The 1-DP is constructed by considering all paths that end into a 1 leaf and following the same process. The following are the decision polynomials for the DT in Figure 5.1a:

$$
\begin{aligned}
P_1^G(X_1, X_2) &= \mathbf{1}[X_1 \leq 10] \cdot \mathbf{1}[X_2 \leq 50] + (1 - \mathbf{1}[X_1 \leq 10]) \cdot \mathbf{1}[X_2 \leq 20], \\
P_0^G(X_1, X_2) &= \mathbf{1}[X_1 \leq 10] \cdot (1 - \mathbf{1}[X_2 \leq 50]) + (1 - \mathbf{1}[X_1 \leq 10]) \cdot (1 - \mathbf{1}[X_2 \leq 20]),
\end{aligned}
$$

where $\mathbf{1}$ is the indicator function.

In both polynomials, all monomials are monic, while there is no constant term. Furthermore, since each term corresponds to a single root-to-leaf path, each polynomial outputs either 0 (if the given assignment does not satisfy any path) or 1 (if a path is satisfied), so we conclude that they are indeed valid DPs. This process exemplifies the general reasoning, which remains unaltered, no matter how large a DT is.

Having the decision polynomials, we can now form a distance function and put all the pieces together. To this end, let $d = (X_1, X_2)$ be a factual datapoint of interest. Utilizing the weighted $l_1$ norm, $\| \cdot \|_{1,w}$, and the rule representation of the DT, the distance between two points can be defined as follows:

$$
\begin{aligned}
\|d - d'\|_{1,w} = w_1 |\mathbf{1}[X_1 \leq 10] - \mathbf{1}[X_1' \leq 10]| + w_2 |\mathbf{1}[X_2 \leq 20] - \mathbf{1}[X_2' \leq 20]| \\
+ w_3 |\mathbf{1}[X_2 \leq 50] - \mathbf{1}[X_2' \leq 50]|,
\end{aligned}
$$

where $w_1, w_2, w_3$ are constants. The last step is to remove the absolute values from the objective function. This is simple to do, since the values of the indicators $\mathbf{1}[X_1 \leq 10], \mathbf{1}[X_2 \leq 20], \mathbf{1}[X_2 \leq 50]$ are known quantities, and $0 \leq \mathbf{1}[\cdot] \leq 1$. The result is the objective function of the final optimization problem.

Going on with the example, let us also assume that $(X_1, X_2)$ satisfies $X_1 \leq 10, 20 < X_2 \leq 50$, so it is classified into the 1 class. Using the 0-DP and applying Proposition 5.5 (i.e. forcing it to be equal to 1), the final optimization problem is:

$$
min \;\; w_1(1 - \mathbf{1}[X_1' \leq 10]) + w_2 \mathbf{1}[X_2' \leq 20] + w_3(1 - \mathbf{1}[X_3' \leq 50]) \; s.t.
$$

$$\mathbf{1}[X_1' \leq 10] + (1 - \mathbf{1}[X_2' \leq 50]) \geq 2 \cdot \delta_1,$$
$$(1 - \mathbf{1}[X_1' \leq 10]) + (1 - \mathbf{1}[X_2' \leq 20]) \geq 2 \cdot \delta_2,$$
$$\delta_1 + \delta_2 = 1$$

The solution of this problem depends on the values of $w_1, w_2, w_3$, however, is guaranteed to be an instance that is classified as 0 by the DT. In fact, this process will result into an infinite set of counterfactuals, since if for example the solution turns out to be $\mathbf{1}[X_1' \leq 10] = 1, \mathbf{1}[X_2' \leq 20] = 0, \mathbf{1}[X_2' \leq 50] = 0$, then $X_1$ remains unaltered, but for $X_2$ we have that every element of the set $\{(X_1, X_2') : X_2' > 50\}$ is a valid counterfactual to $d$, with respect to the decision tree. This is an extension of the framework in (Russell, 2019), where the outcome was a single point.

Finally, we discuss the amount of constraints that has to be added to the optimization problem. As the DPs encode the root-to-leaf paths of the decision tree, the amount of constraints depends on the number of distinct root-to-leaf paths, $m$. The added flexibility of expressing the final problem using either of the two DPs, allows for efficiently handling situations that would be otherwise problematic. For example, if there is a DT having only one path that leads to a 0-leaf, and all the remaining ones lead to a 1-leaf, then everything can be encoded using the 0-DP in a highly efficient manner, using a single constraint, instead of $m - 1$ ones. This demonstrates that the worst-case scenario is when there is an equal number of 0-leaf and 1-leaf paths, in which case the cost of encoding the constraints is the same, no matter which DP is utilized. This means that in the worst case $O(\frac{m}{2})$ constraints would be necessary, each one involving $O(p)$ variables, where p is the length of the longest path in the tree.

### 5.3.2 Random Forests

In this section, we examine how to handle ensembles of multilinear models, using RFs as an example. Although the process is similar in spirit, incorporating information from multiple models poses an additional challenge. For example, looking at Figure 5.1b we can verify that the 1-DP of each tree is:

$$T_1 : P_1^{T_1}(X_1, X_2, X_3) = \mathbf{1}[X_2 \leq 1] \cdot \mathbf{1}[X_3 \leq 2] + (1 - \mathbf{1}[X_2 \leq 1]) \cdot \mathbf{1}[X_3 \leq 10],$$
$$T_2 : P_1^{T_2}(X_1, X_2, X_3) = \mathbf{1}[X_1 \leq 5] \cdot \mathbf{1}[X_2 \leq 2],$$
$$T_3 : P_1^{T_3}(X_1, X_2, X_3) = \mathbf{1}[X_1 \leq 3] \cdot \mathbf{1}[X_3 \leq 5] + (1 - \mathbf{1}[X_1 \leq 3]) \cdot (1 - \mathbf{1}[X_2 \leq 6])$$

As usual, each polynomial encodes all the 0 or 1 assignments of each individual tree, but how can we combine them all together so they encode the behaviour of the forest? A first remark is that in order to make sure that the model outputs, for example, 1, it is enough to enforce a constraint that at most one 0-decision polynomial outputs 1. This would mean that the outcome

of at least 2 out of the 3 decision trees is equal to 1, so the whole forest has an output of 1, assuming majority voting.

This kind of reasoning can be applied to ensembles will an arbitrary number of trees, and will be the base of extending the current framework. In fact, it turns out that this approach corresponds to a generalization of the one we presented for DTs, as shown at the end of this section. The following proposition provides for a way to infer whether a decision polynomial is equal to 1.

**Proposition 5.6.** *Let $P_0^G(\boldsymbol{X}) = X_{11} \cdots X_{1k} + X_{21} \cdots X_{2m} + \cdots + X_{n1} \cdots X_{nl}$, where each $X_i \in \{0,1\}$, be the 0-decision polynomial of a model. Furthermore, let the constraints $X_{11} + X_{12} \cdots + X_{1k} - k \leq \delta - 1, X_{21} + X_{22} \cdots + X_{2m} - m \leq \delta - 1, \cdots,$ $X_{n1} + X_{n2} \cdots + X_{nl} - l \leq \delta - 1$, where $\delta \in \{0,1\}$. If $\boldsymbol{X}, \delta$ satisfy all of them, then $\delta = 0 \Rightarrow P_0^G(\boldsymbol{X}) = 0$. An analogous statement holds for $P_1^G(\boldsymbol{X})$.*

Proposition 5.6 can be seen as an indicator of whether a DT outputs 0 or 1, and it can be used to control the behaviour of a RF. For example, assuming we utilize the 0-DPs to generate an instance that is classified as 1, we need to make sure that at least half of these 0-DPs output 0. By adding the constraints in Proposition 5.6 for every DT in the RF and demanding that at least half of the corresponding indicators are equal to 0, we enforce just that, so the majority of the DTs have an outcome equal to 1, meaning that the whole forest outputs 1.

As it was the case with DTs, Proposition 5.6 suffices to generate valid counterfactuals for RFs. However, the same considerations as before apply to the RF case, so it would be desirable to be able to express the optimization problem in terms of a single DP. As it turns out, it is possible to extend Proposition 5.5 so it can handle the RF case as well:

**Proposition 5.7.** *Let $T_1, T_2, \cdots, T_m$ be the DTs of a RF, $F$. For each $T_j$, consider $P_0^{T_j}(\boldsymbol{X})$ and add all the constraints appearing in Proposition 5.5, except for the last one, which is replaced by $\sum_{i=1}^{n} \delta_{ji} = \delta_j$, where $\delta_{ji}$ appears in the i-th constraint of the j-th tree and $\delta_j \in \{0,1\}$ is a newly introduced variable. Finally, add the constraint $\sum_{i=1}^{m} \delta_i > \lfloor \frac{m-1}{2} \rfloor$. If an assignment, $\boldsymbol{X}$, satisfies these constraints, then the majority of trees output 1, so $F$ outputs 1 as well.*

These results connect the behaviour of a single model to the behaviour of the ensemble, allowing to control the number of models that output a certain outcome. However, tree ensembles present an additional challenge that needs to be addressed; that is, we need to make sure that the solution of the optimization problem is consistent. In this setting, the term consistency is used in the sense that if the solution dictates that a condition of the form $X \leq \alpha$ holds, then all the conditions of the form $X \leq \beta$, where $\alpha \leq \beta$ hold as well. Furthermore, by the same reasoning, if a condition $X \leq \alpha$ does not hold, then no condition $X \leq \beta$, where $\alpha \geq \beta$ should hold. To this end, I define the following quantities:

**Definition 5.8.** Let $T_1, T_2, \cdots, T_n$ be DTs and $X$ one of the variables in their scope. We define:

- $F_x = \{X \leq a : X \leq a \in F(T_i), i \in \{1, 2, \cdots, n\}\}$, where $F(T_i)$ is the set of all the internal rules in $T_i$. In turn, $F_x$ is the set of all the rules among all the trees that involve variable $X$.

- Furthermore, let $X \leq a$ be an element of $F_x$, and define $F_x^+(X \leq a) = \{X \leq b : X \leq b \in F_x, b \geq a\}$, the set of rules involving $X$ where the threshold is larger than $a$, and $F_x^-(X \leq a) = \{X \leq b : X \leq b \in F_x, b \leq a\}$, the rules where the threshold is smaller than $a$.

The following proposition provides a way to achieve consistency by enforcing a set of constraints:

**Proposition 5.9.** *Let $T_1, T_2, \cdots, T_n$ be DTs that form a RF. Then, the constraints $\sum_{f_i \in F_x^+(X \leq a)} \mathbf{1}[f_i] \geq |F_x^+(X \leq a)| \cdot \mathbf{1}[X \leq a]$ and $\sum_{f_i \in F_x^-(X \leq a)} \mathbf{1}[f_i] \leq |F_x^-(X \leq a)| \cdot \mathbf{1}[X \leq a]$, guarantee that the final solution is consistent wrt the rule $X \leq a$.*

Looking at Proposition 5.9 we see that two constraints per rule are enough to guarantee consistency. We can now examine the number of constraints that are required in order to generate a counterfactual from a RF. Clearly, we have to include the counterfactual generating constraints as well as the consistency ones. The former, amounts to incorporating the DP of each tree in the forest. Based on our results for the DT case, assuming there are $N$ trees, $O(\frac{Nm}{2})$ constraints are required in the worst case, where $m$ is the maximum number of distinct paths among all $N$ trees. For the latter, we have to add two constraints per rule, meaning that $O(NF^*)$, where $F^* = max_{T_1, \cdots, T_N}(F_{T_i})$, constraints are required. Combining these together, in the worst case $O(N(\frac{m}{2} + F^*))$ constraints are needed to define a counterfactual generating problem, which scales linearly with respect to the number of trees in the forest.

We are now ready to demonstrate how to generate counterfactuals for RFs, by combining Proposition 5.6, Proposition 5.5, and Proposition 5.9. Returning to our running example, shown in Figure 5.1b, let $d = (X_1, X_2, X_3)$ be a datapoint that satisfies the conditions $X_1 \leq 3,\ X_2 \leq 1,\ X_3 \leq 2$, meaning that all 3 DTs classify $d$ as 1. Assuming we utilize the 1-DPs, the following optimization problem generates a set of solutions that are classified as 0 by the RF, by forcing the majority of the 1-DPs to output 0:

$$\begin{aligned}
&\min w_1(1 - \mathbf{1}[X_1 \leq 5]) + w_2(1 - \mathbf{1}[X_1 \leq 3]) + w_3(1 - \mathbf{1}[X_2 \leq 1]) \\
&+ w_4(1 - \mathbf{1}[X_2 \leq 2]) + w_5(1 - \mathbf{1}[X_2 \leq 6]) + w_6(1 - \mathbf{1}[X_3 \leq 2]) \\
&+ w_7(1 - \mathbf{1}[X_3 \leq 10]) + w_8(1 - \mathbf{1}[X_3 \leq 5]) \quad \text{s.t.}
\end{aligned}$$

$$\left.\begin{aligned}
&\mathbf{1}[X_2 \leq 1] + \mathbf{1}[X_3 \leq 2] - 2 \leq \delta_1 - 1,\\
&1 - \mathbf{1}[X_2 \leq 1] + \mathbf{1}[X_3 \leq 10] - 2 \leq \delta_1 - 1\\
&\mathbf{1}[X_1 \leq 5] + \mathbf{1}[X_2 \leq 2] - 2 \leq \delta_2 - 1,\\
&\mathbf{1}[X_1 \leq 3] + \mathbf{1}[X_3 \leq 5] - 2 \leq \delta_3 - 1\\
&1 - \mathbf{1}[X_1 \leq 3] + 1 - \mathbf{1}[X_2 \leq 6] - 2 \leq \delta_3 - 1,\\
&\delta_1 + \delta_2 + \delta_3 \leq 1
\end{aligned}\right\} \text{0 generating constraints}$$

$$\left.\mathbf{1}[X_1 \leq 5] \geq \mathbf{1}[X_1 \leq 3]\right\} X_1 \text{ consistency constraints}$$

$$\left.\begin{aligned}
&\mathbf{1}[X_2 \leq 2] + \mathbf{1}[X_2 \leq 6] \geq 2 \cdot \mathbf{1}[X_2 \leq 1],\\
&\mathbf{1}[X_2 \leq 6] \geq \mathbf{1}[X_2 \leq 2]\\
&\mathbf{1}[X_2 \leq 1] \leq \mathbf{1}[X_2 \leq 2],\\
&\mathbf{1}[X_2 \leq 1] + \mathbf{1}[X_2 \leq 2] \leq 2 \cdot \mathbf{1}[X_2 \leq 6]
\end{aligned}\right\} X_2 \text{ consistency constraints}$$

$$\left.\begin{aligned}
&\mathbf{1}[X_3 \leq 5] + \mathbf{1}[X_3 \leq 10] \geq 2 \cdot \mathbf{1}[X_3 \leq 2],\\
&\mathbf{1}[X_3 \leq 10] \geq \mathbf{1}[X_3 \leq 5]\\
&\mathbf{1}[X_3 \leq 2] \leq \mathbf{1}[X_3 \leq 5],\\
&\mathbf{1}[X_3 \leq 2] + \mathbf{1}[X_3 \leq 5] \leq 2 \cdot \mathbf{1}[X_3 \leq 10]
\end{aligned}\right\} X_3 \text{ consistency constraints}$$

Finally, in Section 5.6, we establish a connection between the optimization scheme for DTs and the one developed for RFs, showing that the latter is a generalization of the former.

### 5.3.3 Discriminative Sum-Product Networks

The last class of models we discuss are discriminative SPNs over binary variables, so we can obtain a unified framework that is readily applicable to any BNC. Retrieving the DPs of a DSPN is relatively straightforward, utilizing its interpretation as a collection of induced tree models (Zhao et al., 2016), which was introduced in Chapter 3. Each induced tree is derived by traversing the DSPN top-down, where for every sum (resp. product) node only one (resp. all) of its children is included in the final tree, as outlined in the first 3 conditions in definition 3.4. In the worst case, if a DSPN defined over $N$ variables exhibits no context-specific independence, then this process results into $2^N$ induced trees, one for each assignment (Zhao et al., 2016). However, if there are such relationships, then the DSPN can be much more compact, resulting in fewer and shallower trees. In any case, the worst case complexity is in tune with known complexity results regarding counterfactual generation for BNCs (Shih et al., 2018).

Assuming a DSPN representing a conditional distribution of the form $Y|X_1, \cdots, X_N$, each induced tree contains a set of indicators, which decides the state of the variables in the conditioning set (Peharz et al., 2014). This means that if a tree contains indicators $\mathbf{1}_{X_{i_1}=i_1}, \cdots, \mathbf{1}_{X_{i_k}=i_k}$, then it computes the probability of $Y|X_{i_1} = i_1, \cdots, X_{i_k} = i_k$. Looking at

all the induced trees and collecting the states of the variables in the conditioning set, it is relatively easy to construct the DPs, as follows:

- Let $T$ be an induced tree, and $X_{i_1} = i_1, \cdots, X_{i_k} = i_k$ be the state of the corresponding conditioning variables.
- Compute the odds $\frac{\Pr(Y=1|X_{i_1}=i_1,\cdots,X_{i_k}=i_k)}{\Pr(Y=0|X_{i_1}=i_1,\cdots,X_{i_k}=i_k)}$ . If this is greater than 1, it means that this particular state results in $Y$ being classified as 1, so a term corresponding to the conditioning set should be added to the 1-DP (where if $X_i = 1$ the term contains $X_i$, but if $X_i = 0$ the term contains $1 - X_i$). Otherwise, this term should be added to the 0-DP, instead.
- Repeat this process until all induced trees have been considered.

Once the DP(s) are known, then it is only a matter of enforcing the desired constraints, as per Algorithm 5.

### 5.3.4 Parameters, Prime Implicants, the Non-Binary Case, and Diversity

#### 5.3.4.1 Parameters

In this section, we discuss how to set the weights, $\mathbf{w}$, in the $\ell_1$ norm as well as some possible extensions. In the original work in (Wachter et al., 2018), the inverse of the median absolute deviation (MAD) of a feature was utilized:

$$\text{MAD}_k = \text{median}_{j \in D}(\mid X_{j,k} - \text{median}_{l \in D}(X_{l,k}) \mid) \tag{5.1}$$

where $D$ is the dataset, and $X_{i,k}$ denotes the value of feature $k$, in data point $i$.

As the authors argue, this comes with the advantage of being able to capture the intrinsic volatility of a feature, while also being robust to outliers. However, in our framework we consider binary variables, where using MAD is inappropriate, since it is always equal to 0. Instead, we follow the approach in (Russell, 2019), opting for using the inverse standard deviation of a variable.

#### 5.3.4.2 Generalizations

A special case worth mentioning arises when all weights are equal to 1. Then, the resulting distance, $\|\cdot\|_{1,\mathbf{1}}$, reduces to the Hamming distance (Shi et al., 2020), and the solution of the minimization problem reflects the smallest number of changes that are necessary to make the model change its output. As a matter of fact, this number has already gained significant attention within an emergent line of research regarding explainability approaches, where it is

known as the *robustness* of a classifier (Shi et al., 2020). This means that the presented framework can compute a model's robustness by just setting all weights equal to 1. However, it is much more flexible, since Hamming distance does not allow for assigning different weights to features, they are all treated as being equally important. In contrast, the distance function used in this work admits non-uniform feature weights, reflecting the relative importance of each term

Furthermore, although the focus of this chapter is on generating counterfactuals, it is possible to generate *prime implicant (PI) explanations* (Shih et al., 2018), by making a few minor adjustments. Unlike counterfactual explanations that compute a minimal set of changes enough to alter the model's decision, PI-explanations compute a minimal set of feature values that is enough to maintain the model's decision, no matter the values of the remaining features. Similarly, conditional PI explanations provide the same information, but with the additional constraint that the conditioning variables should be members of the PI set. In other words, they uncover the smallest super-set of the conditioning variables that guarantees that the model's outcome cannot be altered by changes in the remaining variables. The details of how to adjust the presented framework to generate PI-explanations, as well as the validity of the newly derived framework, are shown in Section 5.6.

#### 5.3.4.3 The non-binary case

So far, it has been assumed that all variables (or rules) are binary, but it is possible to extend the framework to the non-binary case, by utilizing a simple transformation. In general, let $X$ be a variable taking values in $\{0, 1, \cdots, k\}$. We can now introduce $k$ new binary variables, $X_0, X_1, \cdots, X_k$, such that $X_i = 1 \Leftrightarrow X = i$. Furthermore, we need to add the constraint $\sum_{i=0}^{k} X_i = 1$ to enforce that $X$ takes exactly one value. Employing this trick it is immediate to handle the non-binary case.

#### 5.3.4.4 Diverse Counterfactuals

One of the benefits of this proposal is that it is seamless to generate diverse counterfactuals, which is not possible for many of the existing techniques, but it is a benefit of employing ILP, as recognized by Russel (Russell, 2019). In the BNC case it is as simple as just setting the variables to their desired values, leaving everything else intact. For example, if we would like to generate a counterfactual satisfying $X = 1$, for some variable $X$, then we just need to add the constraint $X = 1$ on top of the constraints shown in Algorithm 5. In the DT and RF cases, since variables may be continuous, a user could ask for counterfactuals that satisfy conditions such as $a \leq X \leq b$, for a variable $X$ (or a set of variables). This is again easy to handle, since the condition $a \leq X \leq b$, is enough to decide the values of some of the constraints in $F_x$. Then, it is just a matter of plugging these values into the optimization problem (as constraints) and proceeding as normal, leaving the rest unchanged. In contrast, for most of the existing

| | Sex | Age | Race | Juvenile felonies | Prior crimes | Two year residivism | Outcome |
|---|---|---|---|---|---|---|---|
| Factual | Male | 33 | Caucasian | 0 | 2 | Yes | Low score |
| Counterfactual | Male | 33 | Caucasian | $>$ **0** | 2 | Yes | High score |
| Diverse counterfactual | **Female** | 33 | Caucasian | (=0) 0 | 2 | Yes | High score |
| Factual | Male | 21 | Black | 0 | 0 | Yes | High score |
| Counterfactual | Male | $\leq$ **19.5** | Black | 0 | 0 | Yes | Low score |
| Diverse counterfactual | Male | ($\geq$ 20) $>$ **21.5** | Black | 0 | 0 | Yes | Low score |
| Factual | Male | 27 | Black | 0 | 0 | No | Low score |
| Counterfactual | Male | 27 | Black | 0 | $>$ **4.5** | No | High score |
| Diverse counterfactual | Male | $<$ **21.5** | Black | 0 | (=0) 0 | No | High score |
| Factual | Male | 32 | Black | 0 | 0 | No | Low score |
| Counterfactual | Male | 27 | Black | 0 | $>$ **1.5** | No | High score |
| Diverse counterfactual | Male | 32 | **Caucasian** | $\geq$ **1** | ($\leq$ 1) 0 | No | High score |
| Factual | Male | 43 | Caucasian | 0 | 2 | No | Low score |
| Counterfactual | **Female** | 43 | Caucasian | 0 | 2 | No | High score |
| Diverse counterfactual | (Male) | 43 | Caucasian | 0 | $>$ **4** | No | High score |

Table 5.1: COMPAS dataset instances

approaches, incorporating a-priori constraints is a highly non-trivial task, usually achieved by utilizing a set of heuristics.

## 5.4 Experiments

In this section, we will demonstrate some of the advantages of utilizing the proposed framework. To this end, we examine three different case studies, based on the COMPAS, LSAT, and Congressional Voting Records datasets. For the first two, a DT and a RF, respectively, are employed, while the last one is based on a Naive Bayes Classifier, although any BNC can be used. In Tables 5.1, 5.2, counterfactual conditions are in bold, while diversity conditions are inside a parenthesis.

**COMPAS:** COMPAS is a popular algorithm for assessing the likelihood that a person will reoffend (recidivate) within two years from being released from prison. It has drawn significant attention within the fairness in AI community, due to the number of biases it exhibits, such as favoring white inmates against black ones (Dressel and Farid, 2018). The dataset contains the COMPAS training variables (age, race, sex, number of prior crimes, number of juvenile felonies), whether the inmate actually reoffended within a two-year period (two year residivism), as well as the final score generated by the algorithm. A DT was trained on this dataset to predict the risk of reoffending.

Table 5.1 shows the records of 5 inmates, where the first row represents the factual datapoint, the second an unconstrained counterfactual, and the third one is making use of the diversity constraints. Looking at the first one, we see that the unconstrained counterfactual is in fact an infinite counterfactual set, since any instance satisfying "juvenile felonies $>$ 0" is a valid counterfactual.

At this point, judging from the unconstrained counterfactual alone, it would be difficult to assess whether the model exhibits any bias. However, this is a case where diversity constraints

| | Sex | LSAT | Race | UGPA | Outcome |
|---|---|---|---|---|---|
| Factual | Male | 34 | White | 3 | Pass |
| Counterfactual | Male | $\leq$ **19.25** | White | 3 | Fail |
| Diverse counterfactual | Male | ($\geq$ 25) 34 | **Black** | $<$ **1.95** | Fail |
| Factual | Male | 36.5 | White | 3.2 | Pass |
| Counterfactual | Male | $\leq$ **20.75** | **Black** | $\leq$ **1.95** | Fail |
| Diverse counterfactual | Male | $\leq$ **19.25** | (White) White | $\leq$ **2.15** | Fail |
| Factual | Female | 43 | White | 2.8 | Pass |
| Counterfactual | Female | $\leq$ **26.75** | White | 2.8 | Fail |
| Diverse counterfactual | Female | ($\leq$ 20)$\leq$ **19.25** | White | $\leq$ **2.15** | Fail |
| Factual | Male | 35 | White | 2.7 | Pass |
| Counterfactual | Male | 35 | **Black** | $\leq$ **1.95** | Fail |
| Diverse counterfactual | Male | $\leq$ **19.25** | (White) White | 2.7 | Fail |
| Factual | Male | 33 | White | 3 | Pass |
| Counterfactual | Male | 33 | **Black** | $\leq$ **1.85** | Fail |
| Diverse counterfactual | Male | $\leq$ **19.25** | (White) White | 3 | Fail |

Table 5.2: LSAT dataset instances

can lead to valuable insights. Enforcing the "number of juvenile felonies = 0", results into a counterfactual that uncovers the model's hidden biased behavior, since it suggests that had the inmate been female, the model would predict a high score of reoffending.

**LSAT:** LSAT is another popular dataset in the fairness literature, since it exhibits a strong bias against black people, too. In this setting, the model has to predict whether students will pass the bar exam, based on their sex, age, law school admission test (lsat), and undergraduate gpa (ugpa). Table 5.2 shows 5 student records, along with the model's prediction. The first unconstrained counterfactual again corresponds to an infinite set, where making sure that lsat is less than 19.25 is enough to alter the model's prediction.

While looking at this counterfactual does not reveal any biases, the relative discrepancy between the factual value of lsat and the counterfactual condition (about 15 points), should be an indicator that constraining lsat closer to its factual value could expose biased behavior. While incorporating inequality constraints is in general very challenging, in the proposed framework it reduces to assigning specific values to some of the variables in the final optimization problem. As it turns out, enforcing that lsat is greater than 25 leads to a counterfactual that clearly showcases the bias in the model, since it suggests that information about the student's race can be used to alter the model's decision.

This case also demonstrates how such counterfactuals can be used to guide the inspection of the dataset in order to identify the reasons behind this behavior. Upon further exploration, we found that 96.7% of male, white students passed the bar, while the same percentage for male, black students was 77.8%. Furthermore, the number of white students in the dataset was about 21 times that of black ones. This shows that black, male students are severely under-represented, while the imbalance between successful/unsuccessful students in the two groups may prompt the model to assign significant predictive power to a student's race.

Having said that, utilizing the counterfactual it is possible to perform a more targeted analysis, to uncover imbalances that are not as apparent. To this end, the dataset was inspected for black,

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | Outcome |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Factual | + | + | + | - | - | - | + | + | + | - | + | - | - | - | + | + | Democrat |
| Prime implicants | | | ✓ | ✓ | ✓ | | | | | | | | | ✓ | | | Democrat |
| Conditional prime implicants | ✓ | | | ✓ | ✓ | | | | | | ✓ | | | ✓ | | | Democrat |
| Factual | + | + | - | + | + | + | - | - | - | + | - | + | - | + | - | + | Republican |
| Prime implicants | | | | ✓ | ✓ | | | | | | | ✓ | ✓ | ✓ | | | Republican |
| Conditional prime implicants | ✓ | | ✓ | ✓ | ✓ | ✓ | | ✓ | | | | | | ✓ | | | Republican |
| Factual | + | - | + | - | - | - | - | + | + | + | - | - | - | - | + | + | Democrat |
| Prime implicants | | | | ✓ | ✓ | | | | | | | | | ✓ | | | Democrat |
| Conditional prime implicants | ✓ | | ✓ | ✓ | ✓ | | | | | | | | | | | | Democrat |
| Factual | - | - | - | + | + | + | - | - | - | - | - | + | + | + | - | + | Republican |
| Prime implicants | | | | ✓ | ✓ | | | | ✓ | | | | ✓ | ✓ | ✓ | | Republican |
| Conditional prime implicants | ✓ | | ✓ | ✓ | ✓ | | | | | | | | ✓ | ✓ | | | Republican |
| Factual | - | - | + | - | - | + | + | + | + | + | + | - | - | - | + | + | Democrat |
| Prime implicants | | | | ✓ | ✓ | | | | | | | | | ✓ | | | Democrat |
| Conditional prime implicants | ✓ | | ✓ | ✓ | ✓ | | | | | | | | | ✓ | | | Democrat |

Table 5.3: Congressional Voting Records dataset instances

male students with $lsat < 19.5$ and $gpa = 3$, only to find out that all such students failed to pass the bar. However, for white, male students, with the same characteristics, half of them passed the bar. On top of this discrepancy, even the specific instances prompted biased behavior, since, for example, a black student with $lsat = 19$, $gpa = 3$, failed, while a white one with $lsat = 17.5$, $gpa = 3$, succeeded, encouraging the model to take racial information into account.

Following this analysis, it should come as no surprise that the RF picked up a corresponding bias, since by looking at the individual DTs we found 6 different paths that lead to a positive outcome for white, male students with $gpa < 3$, as opposed to only 1 for black students. This means that the RF is more "forgiving" towards white students with low gpa, compared to black ones with the same characteristics. Targeting these two specific subgroups was guided by the insights obtained from the diverse counterfactuals, which eventually led to the discovery of significant information regarding both the dataset and the model. In contrast, none of the approaches in (Cui et al., 2015; Tolomei et al., 2017; Kanamori et al., 2020, 2021) support diversity constraints, which would hinder performing the same kind of analysis.

**Congressional Voting Records:** This dataset contains the votes of the U.S. House of Representatives Congressmen on 16 key votes. This time, the problem is to predict whether a person is a Democrat or a Republican, based on these 16 votes. To this end, a Naive Bayes classifier was used, however the same analysis can be performed for any BNC. Table 5.3 shows how 5 particular congressman voted (where + represents voting for, and - voting against). This time, instead of computing counterfactuals, we present prime implicants explanations, as generated by the slightly adjusting the proposed framework (see Section 5.6).

Looking at the first instance, the unconditional prime implicants form a set of 4 elements, meaning that as long as the votes regarding topics 3, 4, 5, 14 remain the same, the model will always classify that person as a Democrat. Furthermore, to inspect the model in more detail, it is possible to compute conditional prime implicants. For example, requiring that the first vote remains the same, we see that the resulting explanation now has 5 elements, some of them not

present in the unconditional explanation. This result indicates there is some relationship among these variables, which could in turn motivate an additional analysis. As in the previous experiments, these kind of insights are gained by taking advantage of the flexibility of the presented framework, since it allows for generating multiple kinds of explanations, under only minimal modifications, as opposed to the approaches in (Cui et al., 2015; Tolomei et al., 2017; Kanamori et al., 2020, 2021).

## 5.5 Future work and conclusions

In this chapter, we present a framework for generating counterfactual explanations for (ensembles of) multilinear models. This extends the methodology in (Russell, 2019), as well as generalizes some of the results in (Shih et al., 2018). We show how to apply these results to DTs, RFs, and BNCs, however any multilinear model can be utilized, instead. This is in contrast to methods like, (Fernández et al., 2020), which is based on a modification of the CART algorithm, so it is only applicable to DTs and RFs. We show how the new framework permits more expressive distance functions, that incorporate the relative importance of each term, instead of treating all feature changes as equally important or feasible. Furthermore, the BNC case was based on discriminative SPNs, so the presented results further enhance the transparent characteristics of SPNs, equipping them with the ability to generate both counterfactuals and prime implicant explanations. Finally, we demonstrate how diversity constraints can facilitate inspecting a model for biased behaviour, highlighting their advantages over unconstrained counterfactuals.

There is a number of interesting research directions that can be motivated from this work. A first remark is that as can be seen from the complexity results, the worst case scenario is exponential, so there are cases where encoding a DP can be impractical. These situations showcase the importance of developing approximate representations of DPs, that correctly classify instances with high probability. This seems like a natural next step, especially considering the long-standing research line of approximate reasoning in BNs, as well as some recent attempts at approximate reasoning with DTs and RFs (Deng, 2014). An alternative way to address this issue, without resorting to approximations, could be to embed it into optimization frameworks, such as *column generation* (Michele Conforti, 2014), that can effectively handle large problems. Advances in these areas could facilitate generating out-of-the-box counterfactuals, leading to their wider adaptation in practical applications.

## 5.6 Proofs and Extensions

**Proof of Proposition 5.5**

Let $\mathbf{X}$ be an assignment that satisfies all the constraints. From the constraint $\sum_{i=1}^{n} \delta_i = 1$ we have that there is a unique $j$, such that $\delta_j = 1$. This $\delta_j$ appears in a constraint of the form $X_{j1} + X_{j2} \cdots + X_{jp} \geq p \cdot \delta_j \Rightarrow X_{j1} + X_{j2} \cdots + X_{jp} \geq p$. However, it also holds that $X_{j1} + X_{j2} \cdots + X_{jp} \leq p$, so putting these two expressions together we have that $X_{j1} + X_{j2} \cdots + X_{jp} = p \Rightarrow X_{j1} \cdot X_{j2} \cdots X_{jp} = 1$, by Proposition 5.4, which means that $P_0(\mathbf{X}) = 1$. □

**Proof of Proposition 5.6**

If $\delta = 0$, the constraints can be rewritten as:

$$
\begin{aligned}
X_{11} + X_{12} \cdots + X_{1k} &\leq k - 1 \\
X_{21} + X_{22} \cdots + X_{2m} &\leq m - 1 \\
&\cdots \\
X_{n1} + X_{n2} \cdots + X_{nl} &\leq l - 1
\end{aligned}
$$

By Proposition 5.4, this means that all terms in the 0-DP are zero, so $P_0(\mathbf{X}) = 0$. □

**Proof of Proposition 5.7**

The last constraint enforces that more that half of the $\delta_i$'s are equal to 1. The result follows, since each $\delta_i$ is an indicator of the corresponding DT's outcome, so for more than half of them their 0-DP is equal to 1, i.e. their outcome is 0, thus the whole RF classifies the final instance in the same category. □

**Proof of Proposition 5.9**

Let $X \leq a$ be a feature in a RF. The first case we are going to examine is when this rule is not satisfied, meaning that $\mathbf{1}[X_1 \leq a] = 0$. Then, the first constraint reduces to $\sum_{f_i \in F_x^+(X \leq a)} \mathbf{1}[f_i] \geq 0$, which always holds. The second constraint however becomes $\sum_{f_i \in F_x^-(X \leq a)} \mathbf{1}[f_i] \leq 0$, which means that $\sum_{f_i \in F_x^-(X \leq a)} \mathbf{1}[f_i] = 0$, forcing all rules within $F_x^-(X \leq a)$ to be false as well, thus guaranteeing consistency wrt to the feature $X \leq a$.

The other case we need to examine is when $\mathbf{1}[X_1 \leq a] = 1$. Then, the first constraint becomes $\sum_{f_i \in F_x^+(X \leq a)} \mathbf{1}[f_i] \geq |F_x^+(X \leq a)|$, which implies that all rules within $F_x^+(X \leq a)$ are also satisfied. The second constraint becomes $\sum_{f_i \in F_x^-(X \leq a)} \mathbf{1}[f_i] \leq |F_x^-(X \leq a)|$, which always holds. □

**Proof that the RF schema generalizes the DT one**

To prove this claim it suffices to consider a trivial RF, comprised of just a single tree, and show that the constraints in Section 5.3.2 reduce to the ones in Section 5.3.1. Without loss of generality, we can assume we are using the 0-DP to formulate the optimization problem, since the same argument applies to the other case as well. Let $T$ be a DT, and let us first consider the case of generating an instance that is classified as 1. To this end, the constraints in Proposition 5.6 will be utilized, treating $T$ as a trivial RF, $F$.

The 0-DP of $F$ is identical to the 0-DP of $T$, so
$P_0^F(X) = P_0^T(X) = X_{11} \cdot X_{12} \cdots X_{1k} + X_{21} \cdot X_{22} \cdots X_{2m} + \cdots + X_{n1} \cdot X_{n2} \cdots X_{nl}$.
Following the procedure in Proposition 5.6, we have to add the constraints:

$$\begin{cases} X_{11} + X_{12} \cdots + X_{1k} - k \leq \delta_1 - 1 \\ X_{21} + X_{22} \cdots + X_{2m} - m \leq \delta_1 - 1 \\ \qquad \cdots \\ X_{n1} + X_{n2} \cdots + X_{nl} - l \leq \delta_1 - 1 \\ \delta_1 \leq 0 \end{cases} \xRightarrow{\delta_1 = 0} \begin{cases} X_{11} + X_{12} \cdots + X_{1k} \leq k - 1 \\ X_{21} + X_{22} \cdots + X_{2m} \leq m - 1 \\ \qquad \cdots \\ X_{n1} + X_{n2} \cdots + X_{nl} \leq l - 1 \end{cases}$$

where the implication follows from the fact that since $\delta_1 \geq 0$, the constraint $\delta_1 \leq 0$ leads to $\delta_1 = 0$. Looking at the right-hand side, we see that these are exactly the constraints that result from Proposition 5.4, thus establishing the desired equivalence for this case.

Furthermore, we have to examine the case of generating an instance that is classified as 0, using the 0-DT. Again, treating $T$ as a trivial RF, Proposition 5.7 can be used to obtain the following set of sufficient constraints:

$$\begin{cases} X_{11} + X_{12} \cdots + X_{1k} \geq k \cdot \delta_1 \\ X_{21} + X_{22} \cdots + X_{2m} \geq m \cdot \delta_2 \\ \qquad \cdots \\ X_{n1} + X_{n2} \cdots + X_{nl} \geq l \cdot \delta_n \\ \sum_{i=1}^{n} \delta_i = \delta \\ \delta > 0 \end{cases} \xRightarrow{\delta = 1} \begin{cases} X_{11} + X_{12} \cdots + X_{1k} \geq k \cdot \delta_1 \\ X_{21} + X_{22} \cdots + X_{2m} \geq m \cdot \delta_2 \\ \qquad \cdots \\ X_{n1} + X_{n2} \cdots + X_{nl} \geq l \cdot \delta_n \\ \sum_{i=1}^{n} \delta_i = 1 \end{cases}$$

where this time the implication follows from the fact that $\delta \in \{0, 1\}$, so the constraint $\delta > 0$ leads to $\delta = 1$. Again, these are exactly the constraints in Proposition 5.5. Additionally, since there is only a single tree in forest, it is not necessary to include the consistency constraints, because inconsistencies only arise when combining multiple trees. This concludes the proof of the claim that the RF constraints generalize the DT ones.

**From Counterfactuals to Prime Implicants**

The procedure of computing the prime implicants of an instance is a simple modification of the one developed for computing counterfactuals. Let us assume the instance of interest is $\mathbf{X} = (X_1, X_2, \cdots, X_n)$. Furthermore, without loss of generality, we can assume that it is classified as 1 by the model. To compute the prime implicants of $\mathbf{X}$, we have to form the objective function (with all coefficients equal to 1) and all the constraints that are necessary so the solution to the optimization problem is classified again as 1. Finally, instead of minimizing this function, we have to maximize it. Intuitively, by doing so, we ask what is the largest set of features that can change values, without altering the model's decision.

The following argument proves that this process indeed results in computing the prime implicants of $\mathbf{X}$. Let us assume that the solution to this problem dictates that variables $X_{i_1}, X_{i_2}, \cdots, X_{i_k}$, should change values, while the variables in $\mathbf{Z} = \{X_1, X_2, \cdots, X_n\} \setminus \{X_{i_1}, X_{i_2}, \cdots, X_{i_k}\}$ should not. Then $\mathbf{Z}$ is equal to the prime implicants. To see this, let us assume that $\mathbf{Z}$ contains $m$ elements, and that the number of prime implicants of $\mathbf{X}$ are $l < m$. Then, as long as these $l$ variables maintain their values, the model will classify the datapoint as 1. In turn, this means that all the remaining $n - l$ variables can switch values, and the model's prediction will remain 1, which implies that these $n - l$ variables constitute a feasible solution of the optimization problem of the previous paragraph. Now this leads to a contradiction, since by assumption the optimal solution of that problem alters the values of $n - m$ variables, so any other solution must alter the values of at most that many variables, i.e. the inequality $n - m \geq n - l \Rightarrow l \geq m$ should hold, which is a contradiction.

# Part II

## *Transparency and Explanations*

# Chapter 6

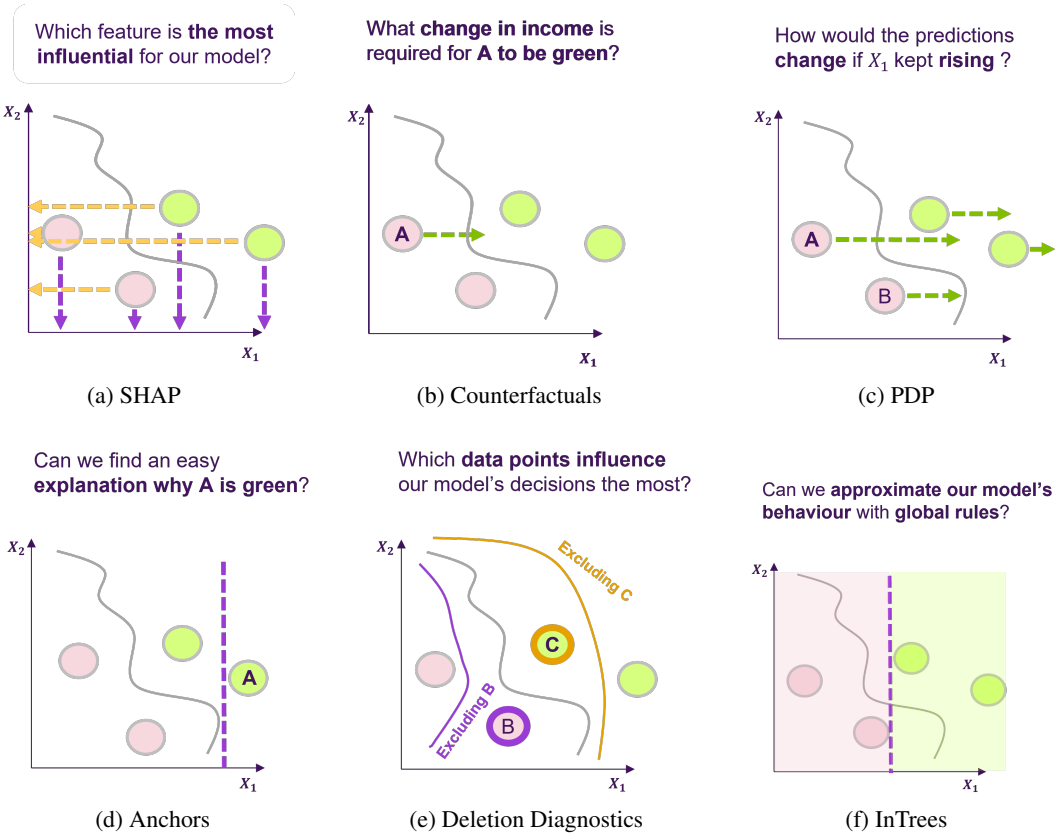
# Related work - Explanations and Trust

In this chapter we are going to review the XAI techniques and related work relevant to Chapters 7, 8.

## 6.1 XAI

The positive impact of AI/ML on society is coupled with the very significant challenge of being able to understand a model's decision-making process. This has given rise to the field of explainability in AI (XAI) which aims at developing tools that facilitate performing post-hoc analysis to discover useful insights. Some of the most popular forms of XAI generated explanation types are the following:

- **Visual explanations** aim at generating visualizations that facilitate the understanding of a model. Although there are some inherit challenges (such as the inability to grasp more than three dimensions), the developed approaches can help in gaining insights about the decision boundary or the way features interact with each other. Due to this, in most cases visualizations are used as complementary techniques, especially when appealing to a non-expert audience.
- **Explanations by simplification** refer to the approach of approximating a black-box model using a simpler model that is easier to interpret. The main challenge comes from the fact that the simple model has to be flexible enough so it can approximate the complex model accurately.
- **Feature importance explanations** attempt to explain a model's decision by quantifying the influence of each input variable. This results in a ranking of importance scores, where higher scores mean that the corresponding variable was more important for the model. These scores alone may not always constitute a complete explanation, but serve as a first step in gaining some insights about the model's reasoning.




(a) SHAP (b) Counterfactuals (c) PDP

(d) Anchors (e) Deletion Diagnostics (f) InTrees

Figure 6.1: Various XAI approaches

On top of that, XAI approaches can be also categorized based on their scope, meaning whether they explain a specific model prediction (*local explanations*), or the overall model behaviour (*global explanations*). Furthermore, they can be categorized based on their applicability, meaning whether they are designed to operate on specific models (*model-specific*), or they are general enough to be applied to any model (*model-agnostic*). Model-specific XAI techniques leverage the properties of a certain model architecture to design efficient algorithms that generate explanations without resorting to approximations. On the other hand, model-agnostic techniques have to be flexible enough, so they do not depend on the intrinsic architecture of a model, thus operating solely on the basis of relating the input of a model to its outputs. In the remaining of this section, some of the XAI techniques that will be needed in later chapters are reviewed, while Figure 6.1 provides a brief graphical summary thereof.

### 6.1.1 SHAP

SHAP (Shapley Additive exPlanations) (Lundberg and Lee, 2017) is a model agnostic method for explaining individual predications. SHAP learns local explanations by utilising Shapley Values (Shapley, 1952) from co-operative game theory, in order to measure feature attributions. The objective is to build a linear model around the instance to be explained and then interpret the coefficients as feature importance scores.

Shapley values provide a means to attribute rewards to agents conditioned on the agent's total contribution to the final reward. In a cooperative setting, agents collaborate in a coalition and are rewarded with respect to their individual contribution. In order to apply this technique to ML models, it is necessary to make adjustments so the problem is expressed in a game theoretic manner:

- *Setting/Game*: SHAP interprets the model prediction on a single input, $x$, as a game.
- *Agents/Players*: The different features of input $x$ are interpreted as being individual players.
- *Reward/Gain*: Measured by taking the model prediction on the input $x$ and subtracting the marginal predictions, i.e. predictions where some of the features are absent.

Having made these adjustments, the Shapley value of feature $i$ equals:

$$\phi_i = \sum_{S \subset F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]$$

where F is the set of all features, $f_{S \cup \{i\}}(x_{S \cup \{i\}})$ is the models decision when the features in $S \cup \{i\}$ are given as input, and $f_S(x_S)$ is the decision when only features in $S$ are given.

### 6.1.2 Counterfactuals

A counterfactual explanation is a statement that identifies how a given prediction would need to change for an alternate outcome to occur. Key to counterfactuals is the idea of "the closest possible world" which signifies the smallest possible change on a set of variables that suffices to alter a model's outcome (Lewis, 1973). For example, if a loan application is rejected, then providing a counterfactual (i.e. a successful application which is as similar as possible to the original one), makes it easier for a person to identify the important information that is relevant to their specific application. In a sense, counterfactuals highlight why a decision was not made, in contrast to other approaches that aim at explaining why a decision was made.

One of the most popular frameworks for generating counterfactuals for ML models is based on (Wachter et al., 2018), where the authors express the problem as:

$$\min_x d(x, x_i) \text{ s.t.}$$
$$f(x) = Y$$

where $d$ is a distance function, $x_i$ is the factual datapoint, $x$ is the counterfactual one, $f(\cdot)$ is the ML model, and $Y$ is the category we would like the counterfactual point to be classified into. For differentiable models, this problem can be solved using Lagrange multipliers, along with an optimization scheme, such as ADAM (Kingma and Ba, 2015).

### 6.1.3 PDPs

Another prominent means of explaining a ML model is using visualizations, especially when communicating explanations to a non-technical audience. A Partial Dependence Plot (PDP/PD) (Friedman, 2001) plots the average prediction as a feature's value changes. These plots can reveal the nature of the relationship between a feature and the output, for example, whether it is linear or exponential. PDPs present global explanations, as the method factors all instances and provides an explanation regarding the (marginal) global relationship between a feature and the model prediction. Assuming we are interested in examining the partial dependence of the model $f$ on a feature $s$, we have to compute:

$$f^*(x_s) = \sum_{i=1}^{N} f(x_s, \mathbf{X}_{-s}^{(i)})$$

where $N$ is the cardinality of the dataset, and $\mathbf{X}_{-s}^{(i)}$ is the $i$-th datapoint, excluding feature $s$. Furthermore, it is not difficult to extended this method to account for the partial dependence of a function on more than one features, however, this is usually done for one or two features.

### 6.1.4 Anchors

Rule-based classifiers have been traditionally utilized due to their transparency, since they are easy to inspect and understand. Anchors (Ribeiro et al., 2018) are a XAI technique that builds on this principle, aiming at generating simple if-then rules to describe a model's reasoning. They explain individual predictions locally by identifying a decision rule that "anchors" the prediction in question, thus they operate on instance level. A rule anchoring a prediction implies that changes to the remaining feature values do not impact the prediction. A rule's *coverage* is defined as the fraction on instances that satisfy the "if" part of the rule. Moreover, a rule's *precision* is the fraction of instances that satisfy both the "if" and the "then" part.

Formally, an anchor, $A$, is defined as the solution of the following problem:

$$\max_{\text{A s.t. } P(prec(A) \geq \tau) \geq 1-\delta} cov(A)$$

where $prec(A) = \mathbb{E}_{D(z|A)}(\mathbf{1}_{f(x)=f(z)})$ is the precision, $D(z|A)$ is the data distribution given the anchor, $f(z)$ is the ML model, $\mathbf{1}$ is the indicator function, and $cov(A) = \mathbb{E}_{D(z)}(A(z))$ is the coverage. In words, this optimization problem looks for if-then rules where the preconditions (the "if" part) contains conditions that are satisfied by as many instances as possible, while requiring that these points also satisfy the "then" part, with high probability. This way the resulting rules are not based on niche characteristics of the specific datapoint at hand, but are as generally applicable as possible.

### 6.1.5 Deletion Diagnostics

Deletion diagnostics is a technique which investigates the model as a function of its training data. It considers the impact of removing a particular training instance from the dataset on the final model (Cook, 1977). By removing an instance with significant influence from training, deletion diagnostics can help with model understanding. In this context, an instance is considered to be influential if its removal causes the parameters of the trained model to change significantly or results in notably different predictions on the remaining instances. This can aid in debugging by locating influential instances that are detrimental to the model's accuracy.

Influence functions (Koh and Liang, 2017) are a contemporary alternative to standard methods of deletion diagnostics, wherein the removal of an instance $i$ is approximated, and the model does not need to be retrained with instance $i$ removed. This makes it more efficient to estimate the influence of a datapoint on the final model, since, retraining a model every time an instance is removed is very computation intensive.

### 6.1.6 InTrees

InTrees (Deng, 2014) are a model-specific XAI method for tree ensembles, which take advantage of the tree architecture to produce interpretable explanations. It can be seen as a collection of multiple algorithms which aim to:

- Extract rules.
- Rank rules.
- Prune rules, removing irrelevant or redundant variable-value pairs from a rule.
- Select rules, choosing a compact set of relevant rules and dismissing redundant ones.
- Summarize the model by taking extracted rules and returning a simplified tree ensemble.

InTrees demonstrate how certain black-box architectures may contain pieces of information that can facilitate the model's understanding. It is worth noting that in the core of this technique lies the idea that although a tree ensembles might be opaque, each of its constituents is transparent, so they can be readily inspected.

## 6.2 Human-AI collaboration

The importance of establishing a transparent relationship between human users and automation on fostering an effective collaboration between the two parties has been consistently identified in prior literature. In (Bhatt et al., 2021) the authors called for utilizing diverse estimates that

convey multiple aspects of the underlying model uncertainty to promote transparency and help users comprehend the degree to which a model's predictions should be followed. Moreover, the findings in (Ashoori and Weisz, 2019) suggested that in high-stakes applications uncertainty estimates might not be enough, since the absence of explanations may lead to users entirely dismissing a model, regardless of its accuracy.

Motivated by such discussions, a growing body of recent empirical investigations focus on the relative effect of uncertainty and explanations on joint accuracy and trust. For example, the findings in (Zhang et al., 2020), suggested that simply providing participants with information about model confidence, i.e. the probability a model assigns to its predictions, is more effective than explanations in improving trust and joint accuracy, as well as that explanations were not successful in allowing participants disentangle between high and low confidence predictions. Moreover, the results in (Lai and Tan, 2019) demonstrated that the best joint accuracy was achieved when presenting information containing the model's prediction paired with the corresponding model confidence, in line with (Zhang et al., 2020). Pairing local feature importance explanations and model predictions was slightly less effective, while presenting explanations alone, led only to a minor improvement compared to the baseline.

Another related study is presented in (Bansal et al., 2021b), which explores whether combining model confidence and explanations can further improve the accuracy of the human-AI team. The resulting analysis showed that when both parties had comparable individual accuracy, then presenting participants with the model's prediction and confidence led to the ensemble achieving superior joint accuracy. The authors found no further improvement when pairing this information with explanations, concluding that the former strategy is as effective as the latter, while also being substantially simpler.

Moreover, in both (Bansal et al., 2021b; Zhang et al., 2020) it has been acknowledged that a user's self-confidence should have an effect on the joint human-AI accuracy in the context of decision-making tasks. Despite this being an intuitive remark, to the best of our knowledge, this idea has not been empirically verified. This is in contrast to alternative settings, such as when humans function as operators being in charge of deciding whether to perform a task manually or delegate it to a model, where the role of self-confidence has received considerable attention. In (Lee and Moray, 1994), the authors provided evidence that participants turned into automation only when their trust in its capabilities exceeded their own, otherwise they tended to performed a task manually. These findings are extended in (De Vries et al., 2003), where the results indicated that there exists a fundamental bias towards people trusting their own abilities, instead of the model. Moreover, in (Lewandowsky et al., 2000), it is shown that participants' self-confidence determined whether they retained control or not, strengthening the previous findings.

Apart from exploring the effect of explanations on accuracy in decision-making tasks, other surveys focus on alternative questions, such as the one in (Dodge et al., 2019), which explored the efficacy of explanations in helping human users detect unfair model behaviour.

Interestingly, the results revealed that local explanations were the most effective in exposing fairness discrepancies among individuals, while global ones instilled more confidence in the users that their understanding was correct. In addition, the study in (Wang and Yin, 2021), brought a new perspective by exploring the comparative effect of explanation styles on model understanding, across datasets of varying difficulty. The final results uncovered that the difficulty of the application significantly influenced the effect of explanations on model understanding, while also indicating that local explanations improved participants' objective understanding, and that global explanations improved their self-reported subjective understanding.

A methodological approach shared by many recent studies on trust in the human-AI collaboration, is that trust is measured using agreement and switching percentages. The former is the fraction of times that the user and the model agreed on their final predictions, while the latter is the percentage of times users switched their predictions to follow the model, assuming the two parties initially disagreed. This approach is in contrast with the predominant practice in the human factors and human-computer interaction communities, where trust is assessed based on either specialized trust measuring scales such as (Madsen and Gregor, 2000; Jian et al., 2000; Adams et al., 2003; Cahour and Forzy, 2009), sophisticated implicit behavioural measures (De Vries et al., 2003; Miller et al., 2016), or combinations thereof. Furthermore, focusing exclusively on the aforementioned percentages to measure trust poses a major methodological shift, since both of them are indicators of reliance (Miller et al., 2016; Lee and See, 2004), so this paradigm presupposes that trust can be indirectly inferred through reliance.

## 6.3 Education in XAI

The rapid emergence of XAI has provided a wide variety of explanations that allow for inspecting a model from multiple angles. Following these developments, a natural next step is to make sure that these tools are competently used by parties interacting with automated systems. In fact, this is also highlighted in the *IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems* (Shahriari and Shahriari, 2017), which contains contributions from 250 thought leaders throughout the world. Furthermore, considering that XAI has only recently emerged, many practitioners are not familiar with the developments in the field, thus developing academic resources contributes to the field's visibility/accessibility, while also promoting the engagement between the AI community and communities utilizing AI/ML systems (Bhatt et al., 2020b,a).

In line with such guidelines, a few technical tutorials on XAI have been developed, including (Samek and Montavon, 2020), where the authors focus on methods for explaining deep neural networks, and (Camburu and Akata, 2021), which puts emphasis on natural language explanations. However, it should be noted that such tutorials are usually presented in academic workshops and are intended to other researchers, not practitioners/data scientists. To alleviate

this issue, a XAI course designed for graduate students was offered by the Stanford University (Lakkaraju and Lage, 2019), centered around technical discussions regarding various XAI approaches.

Apart from the above, the work presented in (Khosravi et al., 2022), identifies ways in which XAI can be used to enhance educational processes. This is a complementary research direction, since it does not consider how to teach XAI in itself, but rather its utility as a component in educational systems. The authors go on and discuss some hypothetical cases, showcasing the benefits of incorporating XAI approaches into interactive interfaces. Although this work does not directly touch upon teaching/introducing the related XAI approaches, it presupposes that both students and educators are to some extent capable of interpreting and understanding the corresponding outputs, which necessitates the development of accessible material introducing the relevant XAI tools.

# Chapter 7

# Human-AI collaboration

## 7.1 Introduction

AI and ML models have already become an indispensable component in many applications, ranging from medical diagnosis to criminal justice. However, full automation is not always desirable, especially in high-stakes applications, for example due to ethical (Naik et al., 2022) or fairness (Mehrabi et al., 2021) concerns. Instead, in such cases, humans should be assisted by automated systems so that the two parties reach a joint decision, stemming out of their interaction. The advantage of this approach is that while it makes use of sophisticated AI systems, humans retain full agency over the final decision, limiting the adverse effect of potential poor model predictions. One of the primary objectives of this human-AI collaboration is to achieve high performance, a goal that requires human users to be able to decide when to follow the model's predictions, which is a multi-faceted objective, influenced by complex interactions between multiple factors (Lee and See, 2004; Hoff and Bashir, 2015; Adams et al., 2003).

Identifying such factors as well as the way they influence user behaviour and attitude towards a model has been an active research area for decades within the human factors and the AI communities, resulting in several behavioural theories describing the dynamics of the human-AI interaction (Lee and Moray, 1992; Linegang et al., 2006; Madsen and Gregor, 2000). A consistent point of convergence among these theories is that both model-related factors, such as the extent to which a model is perceived to be *reliable* and *understandable*, and user-related factors, such as their *self-confidence* in their abilities to carry out a task, play a crucial role in the formation of the human-AI relationship.

As far as model-related factors are concerned, the emergence of explainable AI has sparked a surge of empirical studies that explore the effect of different explanation styles on model understanding, or the capacity of explanations to allow users detect unfair model behaviour (Lai and Tan, 2019; Wang and Yin, 2021; Dodge et al., 2019; Lai et al., 2020). Moreover, with

respect to reliability, recent studies have contrasted the influence of model predictions, uncertainty estimates, and explanations on users' perceived model reliability, comparing their relative effectiveness on instilling trust and/or inducing a complementary performance benefit, where the joint human-AI accuracy is superior to the individual accuracy of either party (Zhang et al., 2020; Bansal et al., 2021b; Green and Chen, 2019; Lundberg et al., 2018). While this is an ongoing endeavour, there has been substantial evidence suggesting that uncertainty estimates are at least as effective as explanations in achieving these goals. Moreover, uncertainty estimates are arguably simpler to implement and communicate to diverse audiences, raising questions about the overall utility of explanations.

Having said that, surveys that consider both uncertainty estimates and explanations, usually view them as competing sources of reliability-related information. While this approach has the merit of providing a common ground upon which it is possible to compare the two, it reduces explanations to reliability indicators, even though their primarily function is to enhance understanding (Hoffman et al., 2018). In addition, while prior research suggests that information regarding reliability and understanding have complementary functions (Zuboff, 1988; Sheridan, 1989; Lee and Moray, 1992; Madsen and Gregor, 2000; Kelly, 2003), the aforementioned approach fails to capture this aspect and provide relevant insights. For example, uncertainty estimates may help users decide the extent to which to rely on a model, but they provide no justifications in cases where a model makes incorrect predictions, hindering model acceptance (Ashoori and Weisz, 2019). On the other hand, while explanations mitigate this issue, inferring a model's prediction and uncertainty based on explanations alone, requires substantial technical expertise, while also inducing a very high cognitive load, making it an inefficient strategy for practical applications (Kaur et al., 2020).

In addition to the above, users' self-confidence in their abilities to complete a certain task is another factor that influences multiple aspects of the human-AI relationship (Lee and Moray, 1992; Lewandowsky et al., 2000; De Vries et al., 2003; Lee and See, 2004). A number of empirical surveys have studied this effect in tasks where humans function as operators, deciding whether to perform a task manually or allocate it to a model, providing evidence that humans' self-confidence has a significant influence on trust and reliance. Despite such findings, in the context of joint decision-making, where humans are always in charge of taking a decision, and the model takes on an advisory role, self-confidence has received very little attention. This leaves a significant gap in empirical investigations, especially considering that many surveys in the domain explore questions concerning trust in automation.

Moreover, another point that warrants further consideration is the way trust is operationalized in recent surveys. In particular, trust is almost exclusively assessed through the lens of *agreement* and *switching percentages* (Zhang et al., 2020), as opposed to using specialized trust measuring scales, such as those developed in (Madsen and Gregor, 2000; Jian et al., 2000; Adams et al., 2003; Cahour and Forzy, 2009). Nevertheless, it is well established that both of these percentages measure reliance, not trust, and that they may fail to account for confounding variables, such as time constraints, inherent application risks, or users' own self-confidence

(Miller et al., 2016; Chancey et al., 2013). This is because although trust has been identified to mediate reliance on automation, trust is a broader attitude towards automation, while reliance is a behaviour that may potentially constitute a manifestation of trust (Ajzen, 1980; Lee and See, 2004). For example, it is possible for one to rely on a model without really trusting it, simply because one lacks the background to take an informed decision. On the other hand, it is also possible for users to base their decisions solely on their own knowledge, so any agreement with the model is only coincidental, not a manifestation of reliance or trust.

In this chapter we attempt to address these issues by conducting an empirical study to identify the effect of self-confidence and various types of model assistance on human-AI collaboration. In particular, we seek to identify how the joint accuracy of the ensemble is affected by users' confidence, as well as whether there are differences in user behaviour depending on the provided level of model assistance. Moreover, we seek to uncover potential non accuracy-related benefits of bringing together uncertainty estimates and explanations, looking for differences in terms of reliance, understanding, and trust towards the model. With this we aim to provide evidence that although uncertainty estimates may be as effective as explanations with respect to performance, the latter influence other key aspects, so pairing the two together induces a complementary effect by leveraging the simplicity of uncertainty estimates and the unique insights offered by explanations. More specifically we present the following contributions:

- We design and implement an online empirical study with human participants.
- We identify a complementary effect between uncertainty estimates and explanations, with the former being sufficient for improving performance, and the latter leading to significant improvements in both subjective and objective model understanding.
- We provide evidence that human self-confidence significantly influences the joint human-AI accuracy, while we also illustrate the pitfalls of not properly adjusting for this effect.
- We showcase how different uncertainty measures influence user behaviour.
- We show that both human and model confidence affect reliance, understanding, and trust.
- We demonstrate the limitations of using switching and agreement percentages as a proxy for trust.

## 7.2 Study Overview

In this study we design a salary prediction task, and we seek to answer questions along two principal axes. On the one hand, we follow the discussions in (Bansal et al., 2021b) and we seek to obtain deeper insights regarding the role of the interaction of human and model

confidence in influencing joint accuracy. As the authors note, this interaction should play an important part in regulating joint accuracy, however, there is no concrete evidence supporting this view. In this work we fill this gap, while further expanding on this idea, exploring not only how the joint accuracy is affected, but also how reliance, understanding, and trust are shaped as a result of this interaction. On the other hand, we seek to find evidence of added benefits of pairing uncertainty estimates with explanations. Recent surveys have consistently demonstrated that in terms of accuracy the former is at least as effective as the latter, suggesting that uncertainty alone is enough to promote an effective human-AI collaboration. However, it is still unclear whether combining uncertainty and explanations can yield alternative, non accuracy related benefits. In this work we look for differences with respect to model understanding, which is an important factor, linked to aspects such as model acceptance and long-term adoption (Adams et al., 2003). Similarly, motivated by the discussions in (Bhatt et al., 2021), we explore the effect of combining uncertainty measures of different scope on users' behaviour. In particular, we ask the following research questions:

**RQ1** How is joint predictive performance influenced by the interaction of human confidence, model confidence, and the degree of model assistance?

**RQ2** How are reliance, understanding, and trust towards the model affected by the same factors?

**RQ3** Does the combination of explanations and uncertainty measures offer non accuracy-related, complementary benefits?

**RQ4** How uncertainty estimates of varying scope influence user behaviour?

Studying these questions, we aim to assess the role of self-confidence in decision-making tasks, as well as how different combinations of information elicit differences in user behaviour. Moreover, we demonstrate the pitfalls of using switching and agreement percentages as a proxy for studying trust. In sum, our goal is to uncover concrete advantages of employing combinations of diverse information sources, promoting research that further expands on this topic. This is especially important considering that in naturalistic settings, stakeholders expect combinations of multiple sources of information. More specifically, we aim to test the following hypotheses:

**H1** Superior joint accuracy will be observed when humans have low self-confidence, and the model makes high confidence predictions. Moreover, pairing model prediction and confidence will be sufficient to induce this effect.

**H2** Participants provided with explanations will have better model understanding.

**H3** Reliance, understanding and trust towards the model will be affected by both human confidence and model confidence, as follows:

**H3.1** Reliance will be affected primarily by human confidence, and to a lesser extent by model confidence. Furthermore, we expect to find an increase in reliance when humans have low confidence and the model makes high confidence predictions .

**H3.2** Understanding will be similarly affected by both human and model confidence . In addition, we expect an increase in understanding when both parties have high confidence.

**H3.3** Trust will be affected primarily by human confidence, and to a lesser extent by model confidence. We also expect an increase in trust when both parties have high confidence.

**H4** The difference between uncertainty measures of distinct scopes (global vs local) will induce differences in user behaviour.

### 7.2.1 Experimental Design

#### 7.2.1.1 Participants

We recruited 112 participants from Amazon Mechanical Turk for our experiment. 49 participants were women, and 63 were men. 18 participants were between age 18 and 29, 45 between age 30 and 39, 23 between 40 and 49, and 26 were over 50 years old. Furthermore, our task was available only to USA residents, due to the fact that the selected dataset contained information that was relevant to the USA social context.

#### 7.2.1.2 Dataset

We designed a modified version of the task presented in (Zhang et al., 2020), where participants had to predict whether a person's annual salary was greater than 50000 dollars. However, since this task was based on the Adult dataset, which contains data from the 1994 Census[1], we needed to adjust the salary threshold to account for inflation. Considering that in this time span the US dollar has seen a cumulative price increase of 101.09%, the adjusted value became 100500, which was rounded up to 100000 dollars. The dataset contains 48842 instances, and each one is comprised of 14 features. Following the authors in (Zhang et al., 2020), we opted for using only the 8 most relevant ones, so participants were not overloaded with information. These features corresponded to a person's: age, employer, education, marital status, occupation, ethnic background, gender, as well as the hours-per-week spent working. We trained a gradient boosting decision tree model on 80% of this dataset, leaving the remaining 20% to test its final performance, which turnout out to be 82%.

[1]Link: https://archive.ics.uci.edu/ml/datasets/adult

#### 7.2.1.3 Task Instances

The reason we selected the Adult dataset, was that it contains instances of varying difficulty, where some of them are relatively easy to predict for lay users with no prior related experience, while others can be significantly harder. This allows for actively manipulating participants' self-confidence to study its effect on various aspects of the human-AI collaboration. In contrast, the authors in (Wang and Yin, 2021), utilized datasets that were either relatively easy or very hard for lay users. A limitation of this approach is that when participants perform a task for which they have no related knowledge or intuition, they are in a state of absolute ignorance promoting a blind reliance on the model, which is first very different from being uncertain and second very far from real-life situations. Indeed, it is highly improbable that a model will be employed by stakeholders having no knowledge/intuition regarding the application at hand. Instead, our prediction task is quite intuitive and mostly requires common sense knowledge, while also allowing for different degrees of confidence in one's predictions.

In order to select the actual task instances, we first set the threshold for low confidence model predictions at 65%, meaning that any prediction with probability not exceeding that number, was considered to be a low confidence model prediction. The corresponding threshold for high confidence predictions was set at 80%. We intentionally opted for a relatively large gap between the two thresholds in order to avoid the interval in-between where it is ambiguous whether a prediction should be seen as having low or high confidence. We then went through the resulting filtered dataset looking for instances of varying complexity, from a human's perspective. After completing this step, we needed to make sure that humans and model have comparable individual performances, to match the setting in (Bansal et al., 2021b). Following the suggestion in (Zhang et al., 2020), we used a stratified sampling approach, constraining the model accuracy to be 75%, since the unconstrained accuracy (82%) was very high for lay people. By the end of this procedure, we identified 56 instances, equally divided into the 4 configurations of human/model confidence: (Human - High & Model - High), (Human - High & Model - Low), (Human - Low & Model - High), (Human - Low & Model - Low). In order to verify that these instances were indeed effective both in inducing different states of human confidence and in allowing for comparable human-model performance, we recruited 15 participants from Amazon Mechanical Turk, asking them to provide a confidence score and prediction for each of these datapoints. Finally, we confirmed that our categorization was effective at inducing a different level of self-confidence to lay users ($Z = 8$, $p < 0.001$), as well as that the selected instances allowed for a comparable accuracy between participants and model (Average human accuracy $= 65.6\%$, 95% confidence interval $(54.2, 76.4)$)[2].

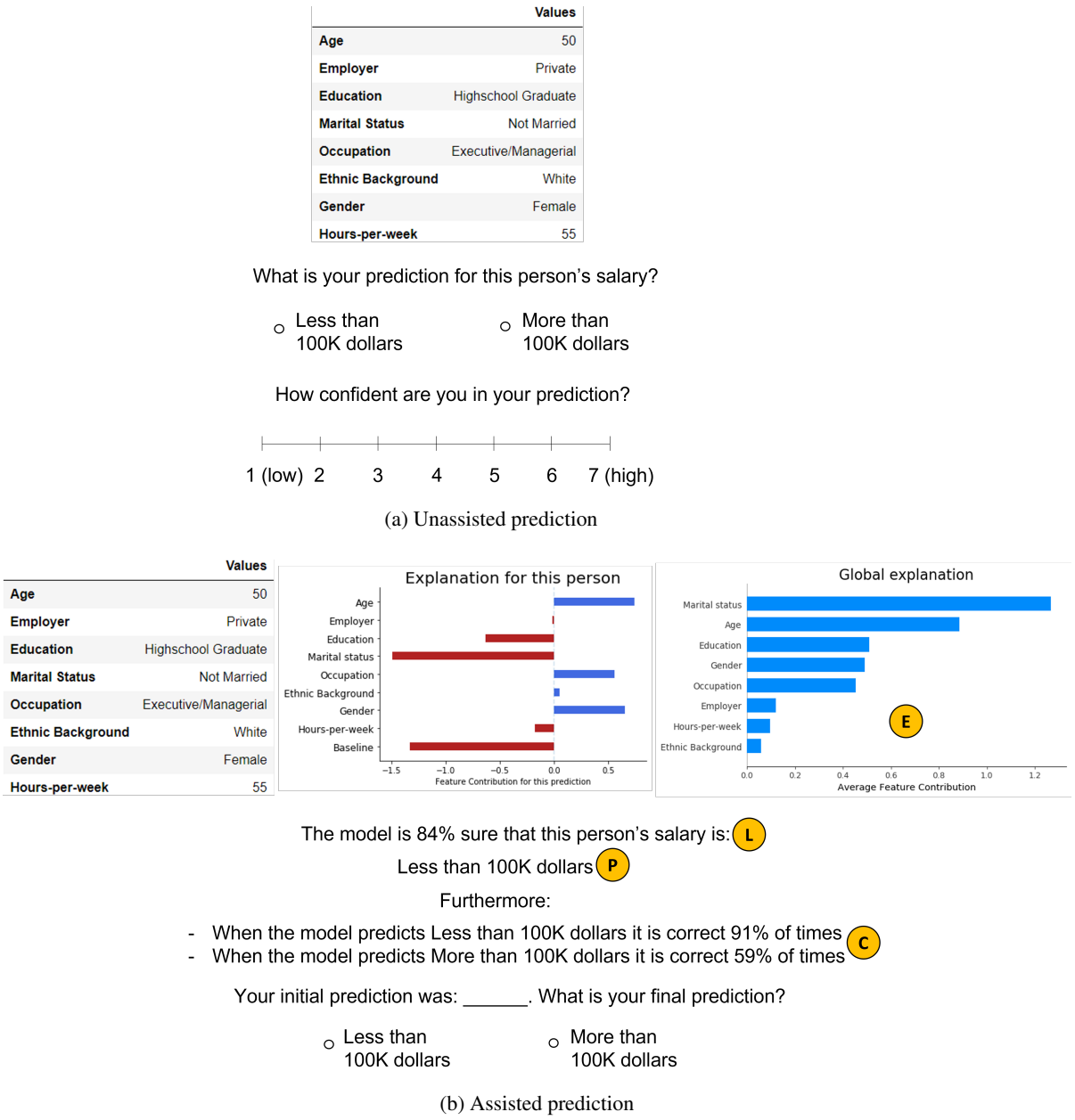
[2]For the former we used Wilcoxon's signed-rank test, while the latter was estimated using the bootstrap method.

#### 7.2.1.4 Design

In order to address our research questions, we designed a prediction task where in each trial participants needed to go through a three-step process. First, they had to inspect an instance, and provide an initial prediction about that person's salary, as well as an estimate of their confidence. Following that they were provided with varying levels of model assistance, depending on the condition (see below), and then they were asked to give their final prediction, where they were free to either maintain or change their initial one. Figure 7.1 shows an example of this procedure. Finally, participants needed to provide an estimate of how much they relied on the model for that prediction, how much they felt they understood its decision-making process, as well as to which extent they trusted the model's prediction. These three steps were repeated in each trial, and after completing the task participants were given a test comprised of 9 multiple choice questions, adapted from (Wang and Yin, 2021), to assess their objective understanding of the model.

In more detail, there were 4 experimental conditions, each one providing an increasing level of model support:

- **Prediction:** In this condition, after participants submitted their initial prediction and confidence score, they are shown only the model's prediction for the same instance. After inspecting it, they are asked to submit their final answer. This serves as the baseline condition, providing only minimal model assistance.
- **Local Confidence:** In this condition, participants were shown both the model's prediction and the corresponding model confidence, i.e the probability that the model assigned to each prediction.
- **Combined Confidence:** In this condition, participants were shown all the information that was available in the previous one, plus the recall for each class, , i.e. the fraction of times an instance is correctly identified by the model as being a member of the class. Here, recall acts as a global meta uncertainty estimate, providing information about the robustness of a model's own confidence. Combining these uncertainty measures should help participants gain a more refined picture of the model's performance, since knowing that a model is, say, 80% confident in its prediction, but predictions for this class are correct only 50% of the time, is more informative than just knowing the model's confidence.
- **Explanations:** In this condition, participants were shown all of the previous information, as well as a local and a global explanation. Based on the findings in (Wang and Yin, 2021), we employed feature importance explanations for both, due to their effectiveness in promoting a better model understanding. Local explanations showed how much each feature influenced the model to reach a particular prediction, while global ones displayed the average overall impact of each feature. All explanations were generated based on SHAP.




(a) Unassisted prediction

(b) Assisted prediction

Figure 7.1: (a) Participants needed to inspect a datapoint and provide their unassisted prediction/confidence. (b) The model assistance presented to participants, depending on condition. Participants in the **Prediction** condition were shown the datapoint and the information next to P. In **Local**, they were shown P + L. In **Combined**, they were shown P + L + C. Finally, in **Explanations**, they were shown all the information contained in this slide.

Participants were randomly assigned to one of the four conditions. Within subjects we manipulated model confidence and human confidence, such that participants in each condition were presented with an equal number of trials with each confidence combination. More precisely, each participant was presented with 4 instances of each of the following certainty combinations: (Human - High & Model - High), (Human - High & Model - Low), (Human - Low & Model - High), (Human - Low & Model - Low). Participants were also asked to provide their confidence in each of their predictions, which was used to confirm that our

manipulation was successful in inducing varying degrees of confidence in this sample too ($Z = 200$, $p < 0.001$).

In addition, we matched the number of instances with people earning less/more than 100K dollars within each certainty combination, such that two out of the four instances of each combination showed people gaining more than 100K dollars. Order of presentation of instances was random. Our dependent variables are accuracy, reliance, subjective understanding of the model, trust and objective understanding of the model.

#### 7.2.1.5 Procedure

Upon accepting to take part in the experiment, participants were presented with the task instructions, which matched the demands of each condition. In the **Explanations** condition, after participants read the instructions, they went through an introduction on explanations and the interpretation of the local and global explanation plots. Then, they were presented with three multiple-choice questions testing whether they conceptually understood the distinction between local and global explanations and whether they were able to correctly interpret the explanation plots. Participants in this condition needed to answer 2 or 3 questions correctly to be included in the sample.

Once the introduction was completed, participants in all conditions went through a familiarization phase, consisting of 12 trials. In each trial, participants first inspected a person's profile for whom the age, employer, education, marital status, occupation, ethnic background, gender and hours per week spent working were provided. Participants had to predict whether this person gains more or less than 100K dollars per year and to give their confidence in their prediction by clicking on a Likert scale ranging from 1 (low) to 7 (high). In the next slide, participants were provided with the model's assistance, which contained different kinds of information depending on the condition (see Section 7.2.1.4) and they were asked to give their final prediction, which could be the same or different from their initial one. Once both of these steps were performed, participants were shown the real life outcome for the person under consideration. The aim of our familiarization phase was two-fold. First, participants could understand better their task and develop some familiarity with the model's assistance (especially in the case of the **Explanations** condition, which contained a greater amount and a more diverse set of information) but more importantly, participants had the opportunity to gain some insight about the model's performance. In particular, given that participants were provided with the real-life outcomes, they were exposed to instances where the model erred, from which they could infer that following the model blindly would not be a fruitful strategy.

After the end of the familiarization phase, participants were informed that they were about to start with the main phase of the experiment, which consisted of 16 trials. In each of them, the two first steps were identical to the first two steps of the familiarization phase, that is participants inspected an instance, they provided their prediction and their confidence in their

prediction and in the next slide they were provided with the model's assistance and they were required to provide their final prediction for this instance. In the test phase, however, after submitting their final prediction, instead of inspecting the real life outcome, participants were asked to answer on a scale from 1 to 7 to which extent they agreed with the following statements, which we borrowed from the scales in (Cahour and Forzy, 2009; Adams et al., 2003):

- *I relied on the model to reach my final prediction.*
- *I understand the model's decision making process.*
- *I trust the model's prediction for this person.*

Finally, after going through all 16 trials, participants were presented with an exit survey of 9 multiple choice questions which assessed their objective understanding of the model, adapted from (Wang and Yin, 2021).[3] The aim of these questions was to address **H2**, since they allowed for comparing model understanding across conditions. This made possible to identify whether explanations offer any significant added benefits, compared to providing users with uncertainty estimates alone. The questions cover a wide spectrum of objectives related to understanding:

- **Global feature importance:** Participants were asked to select the most/least influential features the model utilizes to reach its predictions. (2 questions)
- **Local feature importance:** Participants were given a person's profile, and they were asked to select the most influential feature for this particular case. (1 question)
- **Counterfactual thinking:** Participants were presented with a person's profile, as well as a list of changes in the values of the features, and they were asked to select which of these changes would be sufficient to alter the model's prediction. (2 questions)
- **Model simulation:** Participants were given a profile, and they were asked to answer what they believed the model's prediction for this person would be. (2 questions)
- **Error detection:** Participants were shown a profile, as well as the model's prediction, and they were asked whether they find this prediction to be correct or not. (2 questions)

To make sure that participants were attentive, we included two attention checks in the experiment, where they were given instructions about which answer they should submit. Those who failed to pass the checks, were excluded from the analysis. The base payment was $3.20 for participants in the **Explanations** condition, and $3.00 for the rest of them, since the former required participants to go through an introduction on feature importance explanations. Moreover, to further motivate participants, we included two performance based bonuses; those

[3] All the question can be found in Appendix A.

who provided a correct final prediction on more than 12 of the 16 main trials were given an extra $0.30, and those who answered correctly more than 6 of the questions in the exit survey received a bonus of $0.10.

## 7.3 Results

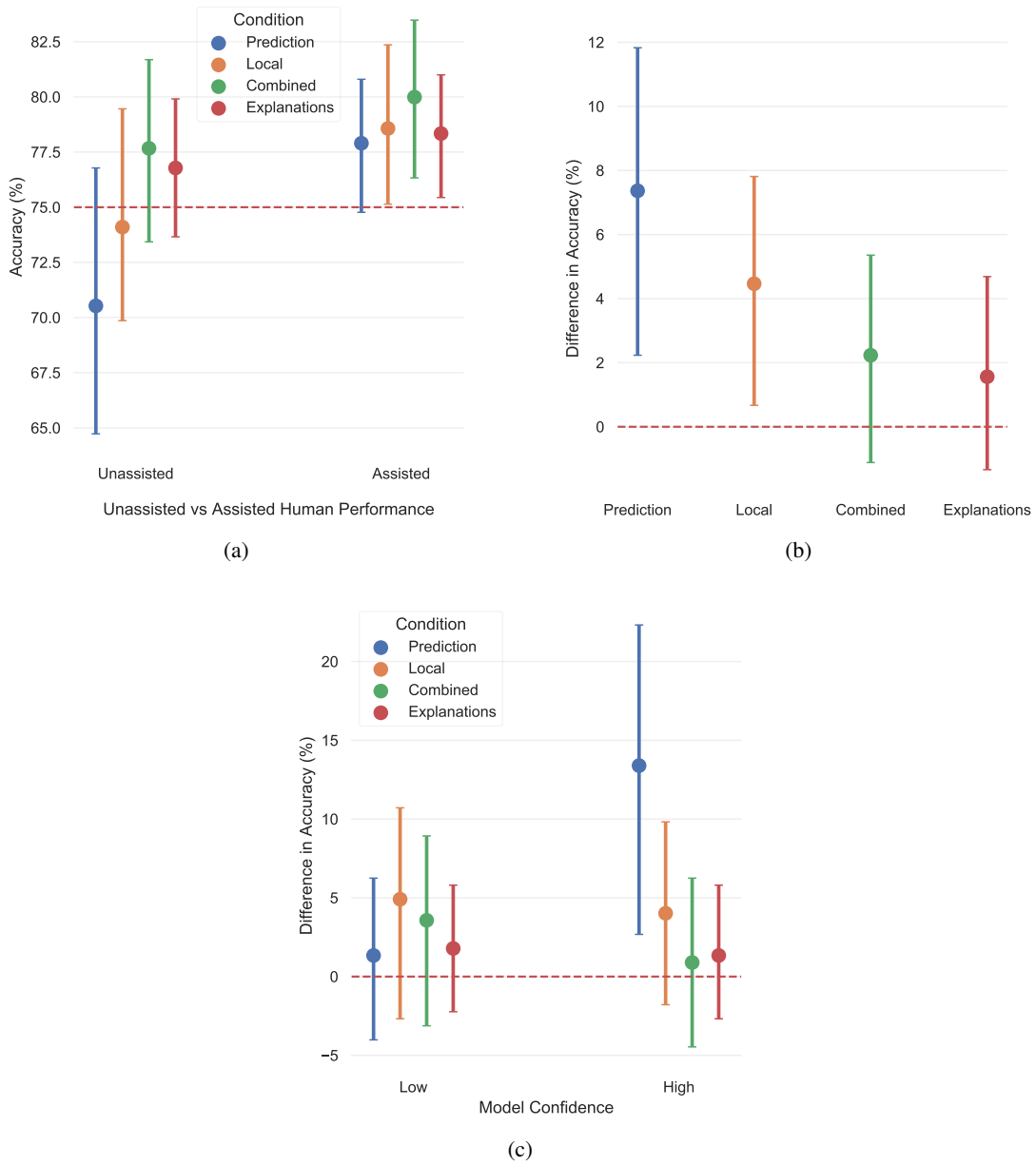



Figure 7.2: (a) Participants' unassisted and assisted accuracy. The red dotted line shows the model's accuracy. (b) Difference between participants assisted and unassisted accuracy, for each condition. (c) Difference in participants' accuracy as a function of the model's confidence.

In this section we present an analysis of our obtained data. All confidence intervals (CIs) were calculated using the non-parametric bootstrap estimation method (Efron and Tibshirani, 1986). Pairwise comparisons between conditions were performed using the Mann-Whitney U Test

(McKnight and Najab, 2010), while all other comparisons were conducted using Wilcoxon's signed-rank test (Woolson, 2007). Details about all CIs and comparisons can be found in Appendix B.

### 7.3.1 Performance

The first set of analyses examined the effect of human confidence, model confidence, and model assistance (condition) on human performance. To address this question, we began with comparing the individual accuracy of the two parties, so that we can then assess whether the ensemble achieved superior performance. To this end, we compared participants' accuracy before exposure to any model assistance (Unassisted Performance) to the model's accuracy. Figure 7.2a, depicts participants' unassisted performance per condition, along with a 95% confidence interval. Details about all CIs are presented in Appendix B. Figure 7.2a shows that 75% belongs to all CIs, so participants and model showed comparable performance in all conditions, thus recreating the setting in (Bansal et al., 2021b).

Then, we compared participants' performance after exposure to the model's assistance (Assisted Performance) to the model's accuracy. Figure 7.2a, shows the assisted performance, along with the corresponding 95% CIs. Participants' assisted performance was significantly better than 75% in all but the **Prediction** condition, suggesting that even the simple strategy of pairing model predictions with confidence, as in the **Local** condition, is beneficial to participants' performance, in line with the findings in (Bansal et al., 2021b). On the other hand, participants in the **Prediction** condition failed to surpass the model's performance, suggesting that predictions alone are not as effective in improving the joint performance, supporting the findings in (Lai and Tan, 2019).

Having established that the model's assistance helped the ensemble surpass the individual model accuracy, we continue by examining whether it surpassed participant's individual accuracy as well. Figure 7.2b, shows the 95% CIs of the difference between participants' assisted and unassisted performance, per condition. Participants' assisted performance was significantly better than their unassisted performance in the **Prediction** and **Local** conditions. On the contrary, the same comparison did not yield statistically significant results in the **Combined** and **Explanations** conditions, even though the point estimates were positive. This pattern can be explained, at least in part, by the fact that participants in the **Combined** and **Explanations** conditions already had better performance in their unassisted predictions compared to participants in the **Prediction** and **Local** conditions, leaving less room for improvement for them. Interestingly, when the point estimate of participants' unassisted accuracy was lower than the model's accuracy (conditions **Prediction** and **Local**), the ensemble surpassed the accuracy of both parties, however, when the point estimate was higher than 75% (conditions **Combined** and **Explanations**), it failed to significantly outperform participants' individual accuracy. In (Bansal et al., 2021b), participants' accuracy was always

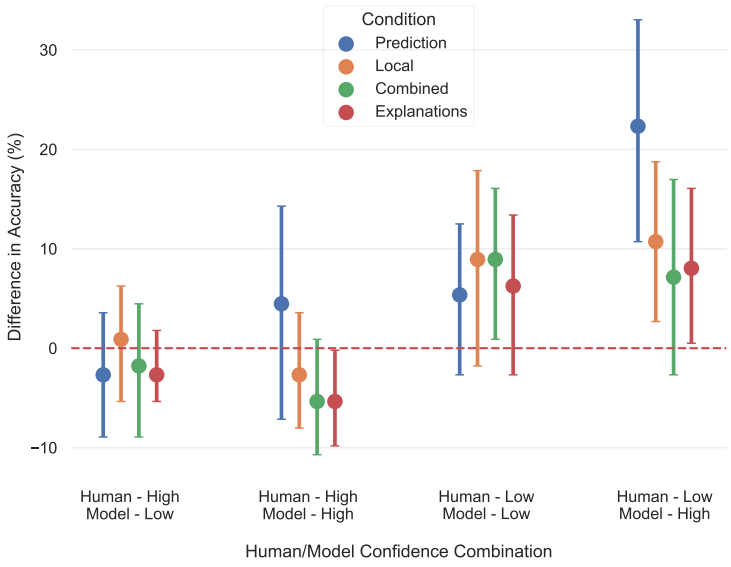



Figure 7.3: The difference between unassisted and assisted human performance, broken down by condition, human confidence, and model confidence. The red line shows the model's accuracy.

lower than the model's, so this might explain why the ensemble achieved superior accuracy in all tasks in their study.

Expanding on the above findings, we then isolated the effect of the different levels of model confidence (Low/High) on participants' accuracy (see Figure 7.2c). The resulting analysis showed that, with the exception of the **Prediction** condition, model confidence did not appear to modulate participants' performance. Note that the **Prediction** condition was the only one where participants had in fact no information about whether the model had low or high confidence, and taking into account the width of the corresponding CI, which suggests that there was substantial variation in participants' accuracy, this result might be due to noise in the data.

A careful inspection of the pattern of the results discussed so far leads to a seemingly paradoxical observation: Focusing on the **Local** condition, we found that the model's assistance significantly improved participants' performance, yet when we broke down this effect for the different levels of model confidence, neither high nor low confidence model predictions significantly improved participants' performance. This leads to the puzzling conclusion that when considering model assistance in general, it helped participants improve their accuracy, but when considering its assistance on low and high confidence predictions separately, the effect vanishes. This phenomenon is known as the *Simpson's paradox*, and it has been extensively studied in statistics, causal inference and philosophy (Wagner, 1982; Julious and Mullee, 1994; Hernán et al., 2011). In statistical terms, this indicates that there are important confounding variables and/or causal relationships, that have not been accounted for into the analysis. The emergence of this phenomenon in our analysis perfectly captured the

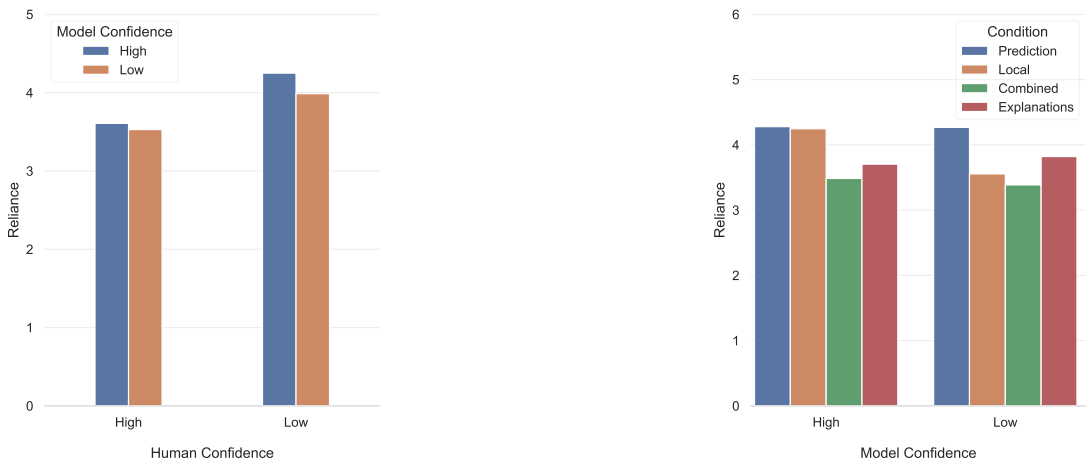
potential perils of not taking into account human confidence, since as soon as we adjusted for this factor, the paradox resolved itself.

Figure 7.3 breaks down the difference between assisted and unassisted accuracy, as a function of condition, human confidence, and model confidence. Participants' accuracy showed a significant improvement when they were themselves uncertain, but the model showed high confidence in its predictions, in all but the **Combined** condition (see Appendix B), meaning that the significant effect in the **Local** condition was driven by this interaction. Furthermore, for the **Combined** condition, we found a significant improvement when both model and human confidence were low. These findings demonstrated that although we found no significant overall improvement for participants in the **Combined** and **Explanations** conditions, interpreting our results through the interaction of human and model confidence allowed us to detect fine grained effects that would have been otherwise missed.

On the other hand, when participants were confident about their predictions, but the model was not, there was virtually no difference in accuracy, indicating that participants' predictions were primarily driven by their own intuitions or knowledge of the world. Finally, when both parties were confident in their predictions, participants' performance slightly declined, but this effect only reached significance in the **Explanations** condition. A possible interpretation of this pattern is that explanations and high model confidence prompted participants to exhibit a slightly over-reliance on the model, which is consistent with the findings in (Kaur et al., 2020). The fact that the reverse trend was observed in the **Prediction** condition strengthens this interpretation, suggesting that in the absence of uncertainty estimates, participants' own confidence dominated, thus no over-reliance was observed. These findings provide strong evidence in favour of **H1**, suggesting that the interaction between human and model confidence is an important factor influencing when and how much a model's predictions will be followed, above and beyond model confidence.

### 7.3.2 Reliance, Understanding, and Trust

This set of analyses examines the effect of human confidence, model confidence and condition on participants reliance, understanding, and trust. Following the discussion in (Wobbrock and Kay, 2016), we opted for analyzing our data using a semi-parametric ANOVA approach, which is robust against violations of the underlying parametric ANOVA assumptions, such as normality, in line with numerous recent studies (Roo and Hachet, 2017; Gugenheimer et al., 2017; Hartmann et al., 2019; Thoravi Kumaravel et al., 2020; Kudo et al., 2021) that utilize non- or semi-parametric methods. In particular, we based our analysis on the Wald-type statistic proposed in (Konietschke et al., 2015).



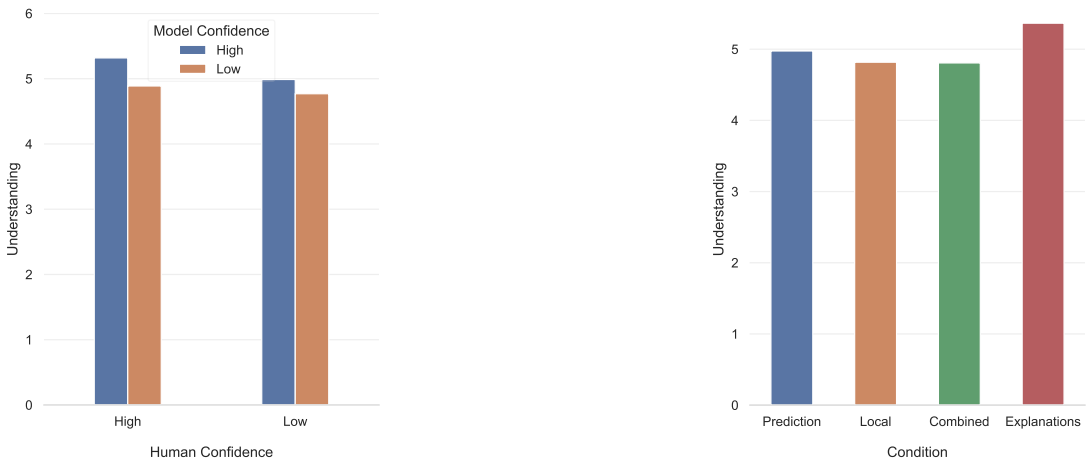

(a) (b)

Figure 7.4: (a) Differences in reliance with respect to the interaction of human and model confidence. (b) Differences in reliance with respect to the interaction of condition and model confidence.

#### 7.3.2.1 Reliance

Starting with reliance, a three-way repeated measures ANOVA with Human Confidence×Model Confidence×Condition identified a main effect of Human Confidence ($W(1) = 40.17$, $p < 0.001$), a main effect of Model Confidence ($W(1) = 5.138$, $p = 0.023$), as well as an interaction between Condition and Model Confidence ($W(3) = 17.574$, $p = 0.001$). Participants' reliance dropped by 7.8% when they themselves were confident, compared to when they were uncertain. Moreover, participants' reliance increased by 2.4% when the model made high confidence predictions. Contrasting these two effect sizes, we see that the former is more than 3 times bigger than the latter, providing evidence that it is primarily human confidence that influences model reliance, in line with **H3.1**. However, overall this hypothesis was only partially confirmed, since we did not detect a significant interaction between human and model confidence ($W(1) = 1.344$, $p = 0.246$). That being said, we suspect this was due to sample size limitations, since the combination of confidences (Human - Low & Model - High) showed the greatest reliance, suggesting that a bigger sample size would lead to statistically significant results (see Figure 7.4a).

With respect to the interaction between Condition and Model Confidence, pairwise comparisons revealed that this effect was due to the **Local** condition ($Z = 32$, $p < 0.001$). Moreover, as Figure 7.4b shows the remaining conditions exhibited virtually no variation in reliance for the different levels of model confidence. In the **Local** condition, participants' reliance was 9.8% higher when the model was confident, compared to when it was not. A possible interpretation of this finding is that while local confidence communicates model uncertainty, it does not provide any meta-information quantifying the robustness of this information, thus it did not allow participants to adjust their reliance. This is because they were only aware of the model's uncertainty, but they did not have any information about either the




(a) (b)

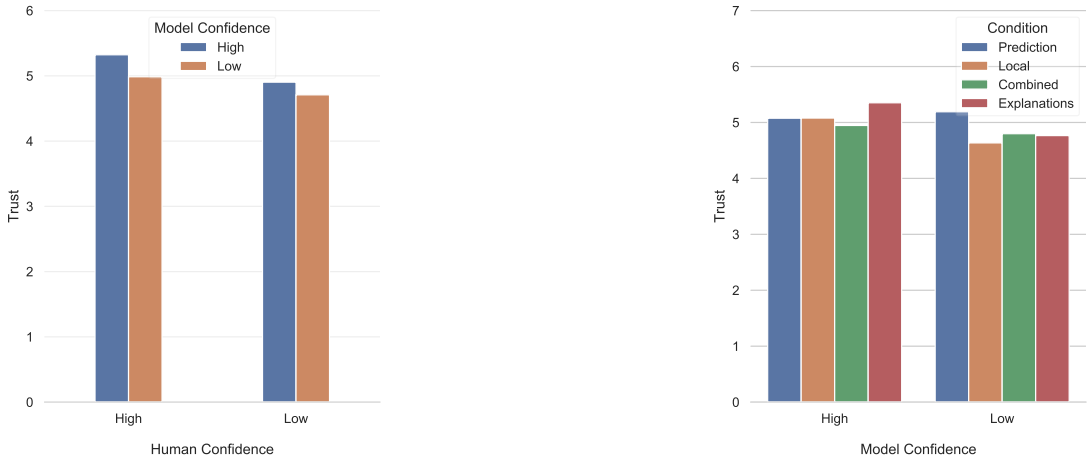
Figure 7.5: (a) Differences in understanding with respect to the interaction of human and model confidence. (b) Differences in understanding with respect to each condition.

model's global error rates (as in the **Combined** condition) or about the reasons behind the prediction (as in the **Explanations** condition). This is a very interesting finding that demonstrates that although extra information might not necessarily lead to better predictive accuracy, it can play a major part in adjusting human behaviour.

#### 7.3.2.2 Understanding

Moving on we turn our attention to participants understanding, and how it was impacted by the various factors in our study. A three-way repeated measures ANOVA with Human Confidence$\times$Model Confidence$\times$Condition identified a main effect of Human Confidence ($W(1) = 18.114, p < 0.001$), a main effect of Model Confidence ($W(1) = 23.015, p < 0.001$), a main effect of Condition ($W(3) = 10.944, p = 0.012$), as well as an interaction between Human Confidence and Model Confidence ($W(1) = 3.963, p = 0.047$). Participants' subjective understanding improved by 3.2%, when they had high confidence, suggesting that they took into account their own knowledge when interpreting the model's predictions. Moreover, participants' understanding improved by 4.6% when the model was confident, compared to when it was not, providing evidence that high confidence model predictions made participants feel more certain that their understanding was correct. With respect to the interaction of human and model confidence, pairwise comparisons revealed that when both human and model confidence were high, understanding was significantly higher than all the remaining combinations. In more detail, compared to the combinations (Human - High & Model - Low), (Human - Low & Model - High), (Human - Low & Model - Low), understanding was 4.72% ($Z = 1755, p < 0.001$), 6.41% ($Z = 1555, p < 0.001$), and 7.84% ($Z = 1148.5, p < 0.001$), higher, respectively. This provided evidence that the interaction of human and model confidence influences model

understanding, which fully supported **H3.2**. No other comparison yielded significant differences (see Figure 7.5a and Appendix B)




(a) (b)

Figure 7.6: (a) Differences in trust with respect to the interaction of human and model confidence. (b) Differences in trust with respect to the interaction of condition and model confidence.

Finally, looking at the main effect of Condition, pairwise comparisons showed that subjective understanding ratings in the **Explanations** condition differed significantly from the ones in the **Local** ($U = -2.5, p = 0.0365$) and **Combined** ($U = -3.01, p = 0.007$) conditions, but not from the ones in the **Prediction** condition ($U = -1.13, p = 0.774$). Figure 7.5b shows the average subjective understanding per condition. The fact that there was no difference between the **Explanations** and **Prediction** conditions, is consistent with the finding that humans tend to project their reasoning on the model, without actually having a well-versed understanding of the model's decision making process. In contrast, in the **Local** and **Combined** conditions, participants were aware of the model's uncertainty, so they were more conservative with their understanding scores. The actual discrepancy of model understanding between the **Explanations** and **Prediction** conditions will become more apparent in Section 7.3.3, where we discuss participants' objective model understanding.

#### 7.3.2.3 Trust

We concluded this part of the analysis studying participants' trust towards the model's predictions. A three-way repeated measures ANOVA with Human Confidence$\times$Model Confidence$\times$Condition identified a main effect of Human Confidence ($W(1) = 46.269, p < 0.001$), a main effect of Model Confidence ($W(1) = 12.942, p < 0.001$), as well as an interaction between Condition and Model Confidence ($W(3) = 14.817, p = 0.002$). Participants' trust increased by 5% when they were confident in their predictions. Moreover, participants' trust increased by 3.7% when model confidence was high. The difference in size between these two effects suggests that while both

influenced participants' trust, the uncertainty stemming due to their own confidence had a slightly more pronounced effect. Despite the fact that we did not find significant evidence in favour of the effect arising from the interaction between human and model confidence ($W(1) = 1.358, p = 0.244$), we suspect that this was mainly due to sample size limitations, since the pattern shown in Figure 7.6a, suggested that when both parties were confident, participants' trust was likely higher. As it was the case when studying reliance, **H3.3** was partially supported, calling for further investigations on the effect of the interaction of human and model confidence on trust.

Finally, following up on the interaction between Condition and Model Confidence, pairwise comparisons revealed that in the **Local** ($Z = 88, p = 0.035$) and **Explanations** ($Z = 77, p = 0.016$) conditions participants tended to trust high confidence model predictions more than low ones (see Figure 7.6b). In the **Local** condition, high confidence model predictions improved trust ratings by 6.3%. In the **Explanations** condition, this difference was even more pronounced, and equal to 8.4%. There is a rather intuitive interpretation of this result, in the sense that when participants were presented with local confidence information, it was reasonable that high confidence predictions imparted higher levels of trust. However, when these scores were complemented with global error rates, participants became aware of the fact that high confidence predictions might not necessarily translate into high accuracy, which is why they did not induce the same level of trust ($Z = 160, p = 1$). Having said that, when all this information was paired with explanations, participants were able to inspect the model's reasoning for each individual instance, so high confident predictions paired with reasonable explanations bypassed the uncertainty induced due to poor global error rates (as when the model predicts More than 100K dollars).

### 7.3.3 Objective Understanding

In this section we studied objective model understanding, as captured via the 9 multiple choice questions that participants completed before exiting the experiment. We looked for differences between **Prediction** and every other condition, to assess whether including uncertainty estimates or explanations led to improved understanding, compared to providing model predictions alone. Recall that these questions addressed 5 different aspects of objective model understanding. Each aspect is analyzed separately in order to gain a more refined picture of participants' understanding. Figure 7.7, shows the difference in scores between conditions, broken down by each aspect of understanding. Starting with global feature importance, participants' scores in the **Explanations** condition significantly outperformed those in the **Prediction** one, while there was no difference between the remaining contrasts. This result was not surprising since global feature importance information was available to participants in the **Explanations** condition. However, the fact that there was no difference among the remaining conditions highlighted that uncertainty estimates were as effective as plain predictions in helping participants infer such information.

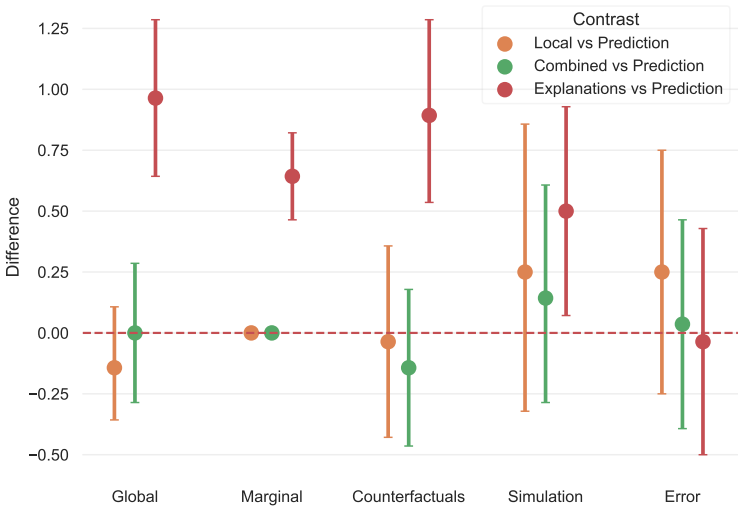


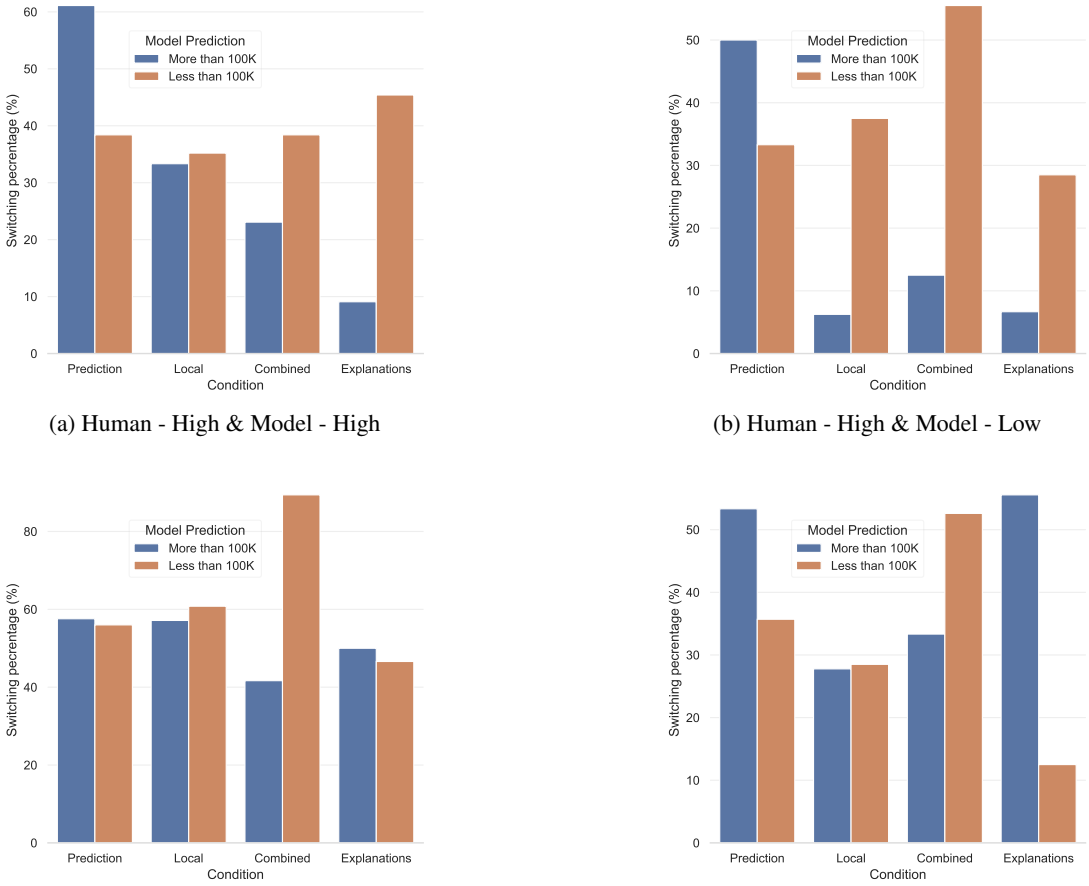

Figure 7.7: Difference between **Prediction** and every other condition, for each aspect of model understanding.

With respect to local feature importance the discrepancy was even more severe, since no participant in the **Prediction**, **Local**, **Combined** conditions was able to provide a correct answer. On the other hand, 64.3% of the participants in the **Explanations** condition answered this question correctly. Again, we expected participants in the latter to have an edge on this task, however, in contrast to global feature importance which remains constant across instances, local feature importance information depends on the instance at hand, meaning that this effect was not due to mere memorization. Instead, participants needed to critically reflect on the information presented throughout the experiment to reach their decision. This sharp difference clearly demonstrated that when it came to inferring local feature importance the information in the remaining conditions was insufficient.

Participants' scores in the counterfactual component of the test showed again that only those in the **Explanations** condition significantly outperformed those in the **Prediction** condition. This is a very interesting finding, indicating that although explanations contained factual information, participants were able to extract counterfactual knowledge out of them, while uncertainty information did not provide any such benefits. The exact same pattern was observed when considering the aspect of model simulation, despite the fact that explanations themselves did not explicitly contain any information regarding simulating the model's behaviour. Regardless, the enhanced understanding of the model's decision making process helped participants in the **Explanations** condition achieve superior performance in the simulation component of the test.

Finally, participants' ability to detect erroneous model predictions was assessed, where no significant differences between conditions were found. Error detection closely resembled the main prediction task, since it required inspecting an instance and the corresponding model prediction to assess its correctness. This means participants in all conditions had substantial exposure/familiarity with this procedure, which explains why there was no difference in their

performance. Overall, the preceding analysis provided strong evidence suggesting that explanations led to better model understanding, compared to uncertainty estimates, thus fully supporting **H2**.




(a) Human - High & Model - High

(b) Human - High & Model - Low

(c) Human - Low & Model - High

(d) Human - Low & Model - Low

Figure 7.8: The switching percentages for the different model predictions. Each subplot corresponds to a combination of human and model confidence.

### 7.3.4 Switching and Agreement

We concluded our analysis by addressing two issues, starting with the effect of pairing uncertainty estimates of different scopes, and then moving to the potential pitfalls of utilizing switching and agreement percentages to measure trust. To this end, we began with a brief qualitative analysis of users' switching behaviour. Unfortunately, when isolating trials where participants' initial answer differed from the model's prediction, the statistical power of our analysis is greatly reduced, so our tests fail to detect significant differences. Despite that, there are some clear patterns present in the data, from which we can gain valuable insights, so we opted for providing a qualitative analysis, instead of dismissing them. Overall, participants' switching percentage in the **Prediction**, **Local**, **Combined**, **Explanations** conditions was 50%, 37%, 45%, and 34%, respectively. Furthermore, in all conditions switching helped participants

improve their performance, since by altering their initial prediction to follow the model's suggestion their accuracy increased by 41%, 25%, 15%, and 17%, following the same order as before.

Focusing on the **Local** and **Combined** conditions, we looked for differences in switching behaviour that can be explained by the fact that global error rates were available in the latter, but not in the former. Figure 7.8 depicts the percentage of trials participants switched their prediction, depending on Condition, Human Confidence, and Model Confidence, where we differentiate between cases where the model predicts Less than 100K and those where it predicts More than 100K. In the (Human - High & Model - Low) combination participants exhibited a similar behaviour in both conditions, presumably because their behaviour was driven by their own intuitions. However, in every other confidence combination participants' behaviour in the **Local** and **Combined** conditions were strikingly different. One the one hand, in the **Local** condition, switching percentages between the two classes were almost identical, but on the other hand, in the **Combined** condition, the switching percentage when the model's prediction was Less than 100K was much higher than when the prediction was More than 100K, consistent with the view that the poor global error rates of the More than 100K class lessened the chances of participants switching to match the model's prediction. Inversely, the great global error rates in the Less than 100K class prompted participants to follow these suggestions.

This is more clearly demonstrated when (Human - Low & Model - High), where knowing that the model had 91% success rate when predicting Less than 100K, encouraged participants in the **Combined** condition to switch in 89% of the trials, compared to 60% in the **Local** one. In line with this reasoning, when the prediction was More than 100K, participants in the former condition were aware that model performance was relatively poor, so their switching percentage plummeted to 41%, which is substantially lower than the 57% in the **Local** condition. This observation perfectly captures the added benefits of pairing these estimates together, as global error rates convey information about the robustness of local confidence scores themselves, which is in line with **H4**, however, additional studies are necessary in order to provide more robust evidence confirming this effect.

In the same vein, while the **Combined** and **Explanations** conditions followed a similar trend for instances with high human confidence, the pattern was drastically different for low confidence instances. Especially when (Human - Low & Model - Low), the trends got reversed, which could be interpreted as additional evidence that explanations promoted case by case reasoning. According to this account, participants in the **Explanations** condition looked past the poor error rate of the More than 100K predictions, using explanations to verify whether the model's reasoning was sound for the instance at hand. Notably they were very successful in doing so, since their accuracy in cases where they switched to follow a More than 100K model prediction was 80%. Future research should investigate this topic in more detail, however this pattern along with the one in Section 7.3.2.3, provided some very promising indications in favour of this interpretation of the results.

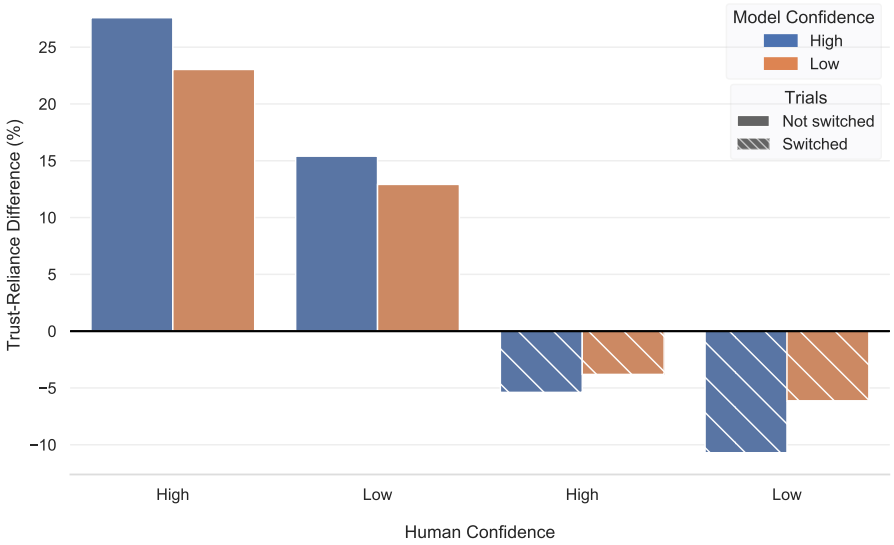



Figure 7.9: The difference between trust and reliance, in terms of the interaction of human and model confidence. Solid bars correspond to trials where participants did not switch their prediction, while dashed ones are computed based on switching trials.

Finally, we discuss a pattern that illustrates the non-equivalence of reliance and trust. Figure 7.9 shows the average difference between participants' trust and reliance scores, once considering trials where participants did not switch their predictions (regardless of whether they initially agreed with the model), and once considering only trials where they switched. In the former, there was a positive trend for all human/model combinations of confidence, meaning that participants' trust scores were higher than their reliance ones. However, when considering only switching trials, a stark contrast was observed, with the trend getting completely reversed, and reliance scores dominating the corresponding trust ones. We should note that this discrepancy was induced by differences in reliance, since although participants' trust increased by 5.51% in switching trials, the corresponding increase in reliance was equal to an impressive 33.11%. Even though we only offer a qualitative account of this phenomenon, the observed pattern is consistent with previous works that argue that both agreement and switching percentages are indicators of reliance, not trust. Adding to this, we found that in 29% of all trials where participants and model agreed, their reported reliance scores were lower that 3 out of 7, meaning that their predictions were predominantly driven by their own intuitions. This indicates that switching percentage is a stronger indicator of reliance, since human-model agreement on its own does not necessarily imply high reliance. Regardless, interpreting either as a manifestation of trust may result to misleading conclusions.

## 7.4 Discussion

In this section we discuss and contextualize our results, as well as we propose several future research directions.

### 7.4.1 The role of human confidence

Our findings provided strong evidence that human confidence has a major effect on multiple aspects of the joint human-AI synergy. Extending the results in (Bansal et al., 2021b), we showed that humans were predominantly benefited by the model's assistance in cases where they are uncertain, but the model made high confidence predictions. This finding is in line with highly influential existing theories on human-computer interaction (Lee and See, 2004; Hoff and Bashir, 2015), where it is argued that users' self-confidence impacts their attitude towards automation. Furthermore, the results presented in Section 7.3.1, demonstrated that not accounting for human confidence may severely distort an analysis. In light of these findings, future experimental studies should be designed in a way that records or controls for human confidence, instead of solely focusing on model confidence. Interestingly, an emerging line of research calls for training ML models using procedures that incorporate human confidence (Bansal et al., 2021a; Mozannar and Sontag, 2020; Wilder et al., 2020), indicating that there is a general interest into utilizing and accounting for this factor.

Beyond predictive performance, our findings suggested that the influence of human confidence extends to users' reliance, understanding, and trust towards a model. Moreover the discussion in Section 7.3.4, emphasized that human confidence also influenced switching and agreement percentages, while also raising concerns about the suitability of these two indicators to assess trust. Previous research has been consistent that both of these measure reliance (Dixon and Wickens, 2006; Madhavan and Phillips, 2010; Miller et al., 2016), and has discussed the caveats of studying trust through reliance (Chancey et al., 2015; Hussein et al., 2020). In our opinion, this calls for rethinking experimental designs or for adjusting the way final results are interpreted. A potential resolution would be to compliment reliance indicators with items from specialized trust measuring scales, and assess trust based on both, which has been the standard practice within the human factors and human-computer interaction communities (Wang et al., 2009; Chancey et al., 2013; Moray et al., 2000; Merritt and Ilgen, 2008), or to use more elaborated behavioural indicators that capture multiple aspects of trust, such as those in (De Vries et al., 2003; Miller et al., 2016). An alternative to modifying the experimental designs, would be to motivate surveys and form hypotheses in terms of reliance (Lee and See, 2004).

### 7.4.2 The complementary effect of uncertainty and explanations

Another central question we explored in this work concerns the role of combining uncertainty estimates and explanations. Prior work suggested that in terms of accuracy, pairing model predictions with the corresponding confidence is as effective as pairing them with explanations (Bansal et al., 2021b; Lai and Tan, 2019; Lai et al., 2020), implying that, performance-wise, uncertainty estimates are as powerful as explanations, while arguably being simpler to understand and implement. Consistent with this idea, our results provided evidence that when

both predictions and confidence information were available, providing participants with additional information did not lead to better performance. Despite that, we identified a strong complementary effect, since participants in the **Explanations** condition had significantly higher self-reported understanding, while also exhibiting a far superior objective model understanding. Interestingly, although only feature importance explanation were provided, their effect permeated multiple aspects of model understanding. Increased understanding has been linked to higher rates of model acceptance (Shin, 2021), while the findings in (Ashoori and Weisz, 2019) indicate that when the stakes are high, ethical considerations may lead to people entirely dismissing a model, regardless of its accuracy, unless they are able to understand its decision-making process. A promising future direction is to adopt a longitudinal experimental design and quantify the effect of explanations on model acceptance or retention. In general, user behaviour is shaped over multiple interactions with the model through an extended period of time, where unexpected or otherwise surprising behaviour may manifest, so longitudinal designs have the potential to provide important insights that are missed by cross sectional designs, which do not record how user behaviour changes over extended periods of time.

Moreover, our results indicated that complementary effects can be found within uncertainty measures too, as discussed in Section 7.3.4. This is consistent with the recent discussions in (Bhatt et al., 2021), demonstrating how communicating different kinds of uncertainty information can induce different user behaviour. In this work we considered predicted probabilities and recall, however there is a lot of room for exploring different measures or combinations thereof, such as precision, false discovery rate, etc. In particular, we find the approach of combining information with diverse scopes (e.g. local and global) to be very promising and worthy of further exploration. An immediate follow up study stemming from our work could explore the effect of more refined global uncertainty information. For example, instead of providing the overall recall of each class, we could first cluster the datapoints based on similarity, and then compute cluster-wise recalls. This localized version of a global summary allows for capturing potential variability in model performance within the same class, depending on sub-population characteristics. However, it should be noted that such approaches require users to have a certain level of numerical competency, which differs substantially from person to person (Zikmund-Fisher et al., 2007), so alternatives exploring visualizations and/or natural language expressions of uncertainty should be considered as well.

### 7.4.3 Explanations in AI

Our findings suggested that explanations provided unique insights that impact model understanding, however explanatory needs are highly dependent on the application (Zhou et al., 2021; Ribera and Lapedriza, 2019). Our work only considered feature importance explanations, however alternative scenarios may call for different types of explanations, such as generating counterfactual instances (Wachter et al., 2018) or propositional rules (Ribeiro et al., 2018). Although there is a number of recent surveys that compare the effect of various explanation

types (Wang and Yin, 2021; Bansal et al., 2021b; Lai and Tan, 2019), to our knowledge there has not been a systematic effort to study the relationship between application characteristics and explanation style preference or efficacy. Furthermore, even within the same application, we expect stakeholders of different expertise to have different explanatory preferences.

Finally, in Section 7.3.1, we provided evidence that when participants had low confidence, model assistance significantly improved their performance, especially when the model generated high confidence predictions. Having said that, when both parties had high confidence, we mostly observed a downwards trend, which resulted in a significant decline in performance in the **Explanations** condition. It is possible that this finding was due to participants' having an information overload (Poursabzi-Sangdeh et al., 2021), where they had a hard time keeping track of all the information that was presented to them. However, other surveys have raised concerns about human over-reliance on a model when explanations are provided (Bansal et al., 2021b; Kaur et al., 2020), so the observed decline in accuracy might be related to this phenomenon. In our view, a promising step towards resolving this situation could be to explore the effect of communicating information about the robustness of an explanation. Most XAI techniques heavily rely on approximations, which means that the final explanation might not be faithful to the model, thus distorting its decision-making process. Moreover, even if no approximations are performed, explanations might face stability issues, where small feature perturbations may lead to drastically different explanations (Yeh et al., 2019). If presented with such information, it is reasonable to assume that users would be more skeptical of explanations, thus reducing their over-reliance. All things considered, we believe that the interplay between uncertainty and explanations calls for further exploration, as it can be integral in guiding the safe and responsible adaptation of automated systems.

## 7.5 Limitations

We acknowledge that one limitation of our study is that we only recruited participants residing in USA, thus we make no claims about the cross-cultural validity of our results. Moreover, we did not record information about participants' familiarity and attitude towards AI, so our results may be influenced by participants predispositions towards automation. Furthermore, participants were not experts on salary prediction tasks. We alleviated this limitation by including a familiarization phase in our experiment. The fact that participants' performance was comparable to the model's indicates that our approach was effective.

Another limitation is that participants were not held liable for their performance, which bared no consequence to them. We addressed this limitation by providing additional performance-based rewards to motivate participants strive for optimal performance.

## 7.6 Conclusions

Previous empirical studies have demonstrated that pairing model predictions and confidence is more effective than explanations in assisting humans improve their accuracy in decision-making tasks. In this work we ask whether bringing them together can provide complementary, non-accuracy related benefits, while also exploring how the interaction of human and model confidence influences human-AI joint accuracy, reliance, understanding, and trust towards the model. To this end, we conducted a study with 112 human participants. We found strong evidence suggesting that human performance is improved in cases where they have low confidence themselves, but the model makes high confidence predictions. Moreover, we found that pairing uncertainty estimates with explanations induces a complementary effect, resulting in high performance and significantly better model understanding. We concluded our findings by providing a qualitative analysis outlining the benefits of combining uncertainty estimates with different scopes, as well as the potential pitfalls of utilizing reliance indicators to measure trust.

We hope that this work will motivate future research that further investigates the role of self-confidence and how different combinations of information influence the human-AI collaboration, in situations where time constraints or other inherent risks are present. Furthermore, another promising direction would be to explore whether interactive methods where humans can actively enquiry a model to satisfy their explanatory needs yield additional benefits, compared to static strategies (like the ones considered in this experiment). Achieving a synergistic relationship between humans and AI is set to be one of the main end goals of the responsible incorporation of AI in our society, and advances along these lines should hopefully bring us a step closer to achieving these endeavours.

# Chapter 8

# Education in XAI

## 8.1 Introduction

As technological advancements have increased computational hardware, modern research has resulted in high performing ML models that have found numerous applications. However, in many of these applications models are treated as black boxes, where the output in no way indicates the decision making process behind it. Consequently, model understanding poses a notable challenge that is imperative to overcome if it is to meet the objective of responsible and beneficial use of ML systems.

To this end, the field of explanations in AI (XAI) has emerged, aiming at designing tools and methodologies that allow the extraction of meaningful information out of black-box models (Arrieta et al., 2020; Linardatos et al., 2020). Although XAI is a relatively young field, it has already generated an impressive amount of scientific literature (Larsson et al., 2019). Furthermore, there is a number of high-performance, open-source implementations of some of the most popular XAI techniques, which has facilitated their rapid adoption in commercial settings. This can be also seen by the spike of scientific publications discussing the deployment of XAI in healthcare, banking, e-commerce, cybersecurity, etc.

However, when it comes at actually employing XAI in practical applications, there is alarming evidence that professionals/data scientists use these tools in a wrong way (Kaur et al., 2020), where misuse most often arises due to misunderstandings around the scope and kind of insights that can be gained when using certain XAI techniques. Consequently, this leads to sub-optimal use of XAI, and by a misinterpretation of the resulting explanations. This finding is related to a broader issue regarding the proper use of AI/ML, which is commonly referred to as *user competence*.

Despite the need for developing the technical skills required to competently use the tools provided by an AI-driven society, XAI related academic resources are extremely limited.

Although there is a number of tutorial and introductory articles that can be found online, there is only a single formal academic course on XAI (Lakkaraju and Lage, 2019).

In this chapter we fill attempt to alleviate this situation, providing a pedagogical perspective on how to structure a course on XAI, in a way the introduces students and professionals to various explainability techniques, while also keeping an eye on the big picture of the field. This proposal takes a distinct stance from the course in (Lakkaraju and Lage, 2019), since the latter focuses on introducing several technical approaches, while the presented course emphasizes the conceptual aspects of explanations, and the discussed techniques are introduced as specific realizations of broader conceptual categories. This decouples individual XAI techniques from the overall objectives, advantages, and challenges of XAI, allowing for updating the material in accordance with the new developments in the field. In particular:

- We structure a course around a putative data scientist, Jane, and discuss how she might go about explaining her models by asking the right questions.
- We pair each lecture with a series of open-ended questions to promote an exchange of ideas between lecturers and/or participants.
- We develop a series of technical tutorials that discuss practical implementations of XAI techniques.
- We evaluate the course based the feedback and assignments provided by the MSc students that attended the course, as it was offered by the University of Edinburgh.

## 8.2 Learning Objectives

Probably the most common way to introduce XAI to individuals interested in applying related techniques is through tutorials, for example (Bennetot et al., 2021; Rothman, 2020). However, most of them are targeted towards a technical research audience, where the purpose is research discovery among peers and not the teaching of fundamentals. Keeping in mind that tutorials usually serve as short, technical introductions to a subject, providing detailed insights regarding the nuances of different techniques or explanations styles is beyond their scope. In contrast, a tutorial following the thrust of the contributions of this chapter was presented at AAMAS 2022[1].

One of the main goals of the course is to fill this need, by imparting a number of key concepts. The first one is that the various explainabilty approaches can be taxonomized such that a technique can be selected by considering explanation types, explanation properties, advantages and disadvantages, as well as the specific model under consideration. Another important concept is that XAI can be used to structure a narrative in order to successfully answer

[1] Link to AAMAS 2022 tutorials: https://aamas2022-conference.auckland.ac.nz/program/tutorials/

potential questions raised by stakeholders, in line with recent works providing evidence that explainability techniques are best linked to stakeholder questions (Arya et al., 2019). Along with imparting theory, an equally important goal is to provide students with experience in applying these explainability techniques using commonly available APIs, so they can gain an understanding of the implementation pipeline (e.g. data cleaning, parameter tuning, etc).

In broader terms, this course enables inclusivity, empowerment, and responsibility with respect to XAI. In regards to *inclusivity*, the course suggests strategies that can help make XAI more easily understood/accessible, especially since current developments can be very technical and difficult to grasp for those not keeping up with the state of the art. Focusing on a representative subset of techniques and showing that they can be tightly coupled with certain types of questions, provides an accessible strategy for introducing students/practitioners into the field. With respect to *empowerment*, the course is designed for students with some experience with data, however it also includes preparatory lectures on machine learning. It gives students the opportunity to engage with machine learning models, debug them, and inspect whether the resulting models fit their purpose. In terms of enabling *responsibility*, it is widely acknowledged that responsible design in artificial intelligence includes many facets, from bias detection to value alignment. In this broad picture of ensuring that machine learning models perform as they should, explainability is an essential ingredient, so it is important that practitioners can make competent use of such tools.

To the best of our knowledge, this is the first work on this topic, so it is not possible to make empirical comparisons. However, the course was evaluated based on students' final assignment, as well as their feedback, achieving very positive results. Hopefully this will be the start of a discussion on how to effectively teach this very important topic.

Upon completion of this course, students are expected to have learnt to apply XAI techniques to enquiry a given model, as well as to have gained an overview understanding of the conceptual distinctions of explanations. More specifically, the expected learning outcomes are as follows:

- **Analyze:** Describe the context of the machine learning application and why explainability would help, but also scrutinise which kind of explainability technique is necessary.
- **Design:** Define the implementation pipeline for an application; provide a means to clean the data, install and set up one or more post hoc explainability techniques.
- **Apply:** Competently apply a wide range of techniques and tools, also knowing their particular features and drawbacks. Have the foundations to understand new and upcoming methods and techniques.
- **Evaluate:** Critically reflect on the results of XAI techniques and investigate their utility in the given context.

In particular, since both the theoretical and practical aspects are covered, students should be able to understand the context in which these techniques are deployed, but also understand the theory in justifying these techniques. The final project in particular is an opportunity for students to create a narrative and an application and motivate and argue for or against a model by using a sequence of techniques.

## 8.3 Course structure and content

The course consists of 9 lectures, which cover the following topics: i) ML preface, ii) XAI preface, iii) SHAP, iv) PDP/ICE, v) Counterfactuals, vi) Anchors, vii) Deletion diagnostics, viii) InTrees, ix) Future research directions. Furthermore, the course includes 4 tutorial sessions, where students have the chance to engage in coding exercises and raise practical issues about the corresponding XAI techniques. Finally, students are expected to submit 2 assignments and a final project.

In order to focus on the evaluation of the course, all the information about the specifics of how each technique was introduced to students, the Jane narrative, as well as the open ended questions, have been moved to Appendix C, while a discussion about the tutorials and the intermediate assignments can be found in Appendix D.

### 8.3.1 Final Project

The final project requires students to consider a ML application, and then carry out all the necessary steps to train a model. After the training is completed, students need to utilize a series of XAI techniques to evaluate the resulting model and argue about whether it should be retained or dismissed. The minimum time commitment expected from students is 14 hours.

Essentially all aspects of the problem specification are decided by the students, i.e. dataset selection, model selection, XAI techniques. Furthermore, students need to come up with a narrative describing the problem, for example, *"Taking a credit scoring dataset, and the XGBoost model, convince a banking institution to reject the model using (at least) technique 1 and technique 2"*. The goal of the project is to prompt students to use various XAI techniques in order to convince a (hypothetical) stakeholder to approve/dismiss the underlying model. This situation resembles what students might come across when applying XAI in a professional setting, so it is important that they can form sound arguments based on the explanations at hand.

| Questions | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 6. Please rate how confident do you feel in applying the XAI techniques you learned in your own models | **Not at all** – I do not feel I can **any** apply the techniques I learned to my models | | **Somewhat** – I feel I can apply **most** of the techniques I learned to my models | | **Very** – I feel I can apply **all** the techniques I learned to my models |
| 7. Please rate how satisfied are you from the diversity of XAI techniques covered in the course | **Not at all** – The techniques were **overly similar** | | **Somewhat** – The techniques were somewhat diverse, but there was **significant overlap** | | **Very** – The techniques were very diverse, and had **minimal overlap** |
| 8. Please rate how much do you feel your understanding of XAI was benefited by the course | **Not at all** – I do not feel the course helped me understand XAI **at all** | | **Somewhat** – I feel I now understand **some** aspects of XAI better | | **Very** – I feel I now understand **many** aspects of XAI better |
| 9. Please rate how much do you feel you have comprehended the conceptual distinctions, advantages, and disadvantages of the XAI techniques covered in the course | **Not at all** – I do not feel I have comprehended **any** of these, for **any** technique | | **Somewhat** – I feel I understand **some** of these, for **some** of the techniques | | **Very** – I feel I understand **all** of these, for **all** the techniques |
| 16. The course was overly theoretical and abstract | **Strongly Disagree** – The course was not theoretical **at all** | | **Neutral** – It had some theoretical aspects, but it was **not excessive** | | **Strongly Agree** – The course was **overly theoretical** |
| 17. The course was overly practical | **Strongly Disagree** – The course was not practical **at all** | | **Neutral** – It had some practical aspects, but it was **not excessive** | | **Strongly Agree** – The course was **overly practical** |
| 19. Overall, I am satisfied with this course | **Strongly Disagree** – I am **not** satisfied at all with this course | | **Neutral** – I am **somewhat** satisfied with this course | | **Strongly Agree** – I am **satisfied** with this course |
| 20. **Analyze:** Describe the context of the machine learning application and why explainability would help, but also scrutinize which kind of explainability technique is necessary. | **Unsuccessful** – The course did **not improve** my abilities to analyze a problem | | **Somewhat successful** – The course **moderately improved** my abilities to analyze a problem | | **Successful** – The course **significantly improved** my abilities to analyze a problem |
| 21. **Design:** Define the implementation pipeline for the project: provide a means to clean the data, install and set up one or more post hoc explain ability techniques through a self-chosen set of programming platforms. | **Unsuccessful** – The course did **not improve** my abilities to design a pipeline for a problem | | **Somewhat successful** – The course **moderately improved** my abilities to design a pipeline for a problem | | **Successful** – The course **significantly improved** my abilities to design a pipeline for a problem |
| 22. **Evaluate:** Critically reflect on the results from such techniques and suggest how it helps the problem context. | **Unsuccessful** - The course did **not improve** my abilities to evaluate the obtained results | | **Somewhat successful** – The course **moderately improved** my abilities to evaluate the obtained results | | **Successful** – The course **significantly improved** my abilities to evaluate the obtained results |
| 23. **Apply:** Competently apply a wide range of techniques and tools, also knowing their particular features and drawbacks. Have the foundations to understand new and upcoming methods and techniques. | **Unsuccessful** – The course did **not improve** my abilities to apply or understand new techniques | | **Somewhat successful** – The course **moderately improved** my abilities to apply or understand new techniques | | **Successful** – The course **significantly improved** my abilities to apply or understand new techniques |

Table 8.1: A description of the questions included in the analysis.

## 8.4 Evaluation Methodology

To assess the effectiveness of the course, a modified version of the Course Experience Questionnaire (CEQ) and a general performance analysis based on the coding assignment were used.

### 8.4.1 Context and Participants

Sixteen survey responses were collected from the students who attended the presented course as part of their MSc studies at the University of Edinburgh. Eight of the respondents were postgraduate research students, five were doing non-academic work, while three responded "Other". Prior to taking this course, 87.5% of respondents were not familiar with the topic of XAI. Most of the respondents had at least some practical experience of working with ML (81.3%) and theoretical knowledge of ML (81.3%). One respondent had no theoretical or practical knowledge of ML, and two respondents had some theoretical, but no practical ML experience. Four respondents rated their practical experiences in ML as close to expert level, and six rated their theoretical ML knowledge close to expert level.

### 8.4.2 Procedures and Data Collection

Conventionally the CEQ is employed as a measure of teaching quality of university courses (Ramsden, 1991). The questionnaire assumes a strong connection between the quality of student learning and student perceptions of teaching (McInnis, 1997). When combined with the additional course assessment items and adapted to the course context it can reliably be applied as a domain-neutral indicator of university course quality (Griffin et al., 2003). For this study 12 items were selected from the original version of the CEQ based on their relevance to the XAI course context (see appendix E). These items were scored on a 5-point Likert-type rating scale from "strongly disagree" to "strongly agree".

In addition to the CEQ items, context-specific items were used in order to gather information about (i) pre-existing skills and theoretical knowledge of XAI; (ii) pre-existing skills and theoretical knowledge of Machine Learning (ML); (iii) satisfaction of the diversity of the taught techniques; (iv) course ability to build understanding of XAI techniques; (v) level of comprehension of the conceptual distinctions, advantages, and disadvantages of the XAI techniques covered in the course; (vi) success in meeting the four pre-set learning objectives. These responses were also based on a five-point Likert scale. Finally, in an open-ended manner, students were asked to list their favourite aspect of the course and to suggest anything that could help to improve the course in future. The resulted questionnaire was given to the students after concluding the final lecture, but before the results of the final assignment came out, so they were not aware of whether they had passed the course.

### 8.4.3 Data Analysis

The results were divided into two parts. Overall CEQ scoring of the responses, as well as quantitative and qualitative analysis of the individual statement ratings. The CEQ raw scores were recorded as follows: a raw score of 1 ('strongly disagree') was recoded to $-100$, 2 to $-50$, 3 to 0, 4 to 50, and 5 ('strongly agree') to 100, eliminating the need for decimal points.

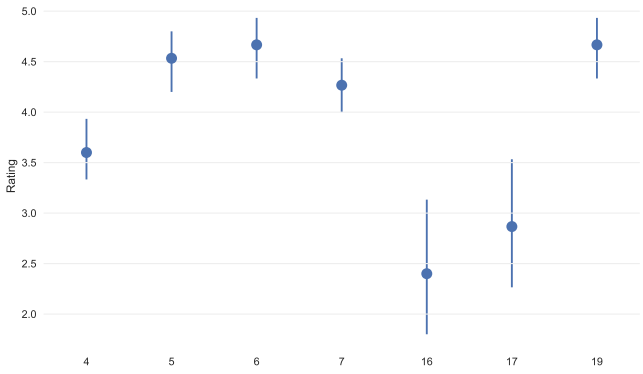



Figure 8.1: Questions

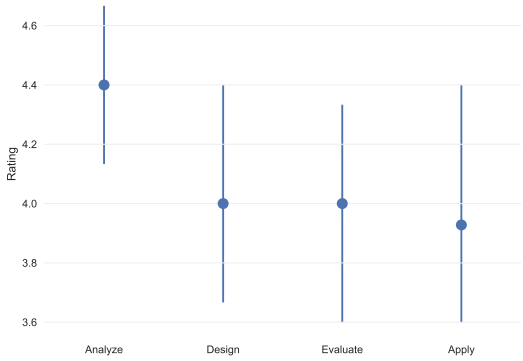



Figure 8.2: Objectives

The scoring of negatively worded items was reversed. In interpreting CEQ results, a negative value corresponds to disagreement with the questionnaire item and a positive value to agreement with the item. Positive high scores indicate high course quality as perceived by graduates. The responses revealed a high positive overall CEQ score of $12,050$, which indicated high course quality as perceived by respondents. The CEQ is widely accepted as a reliable, verifiable, and useful measure of the perceived course quality (Griffin et al., 2003).

On top of that, in order to gain a more fine grained picture of students' responses, individual questions $4, 5, 6, 7, 16, 17, 19, 20, 21, 22, 23$ were analyzed. Questions $20 - 23$ correspond to the learning objectives (Analyze, Design, Evaluate, Apply), while the remaining ones correspond to items indicating the level of students' satisfaction/confidence (see Table 8.1). The analysis was performed by estimating a 95% confidence interval for the average score of each of these questions. Each interval was constructed using the non-parametric bootstrap method (Efron and Tibshirani, 1986). Figures 8.1, 8.2 show the obtained results,

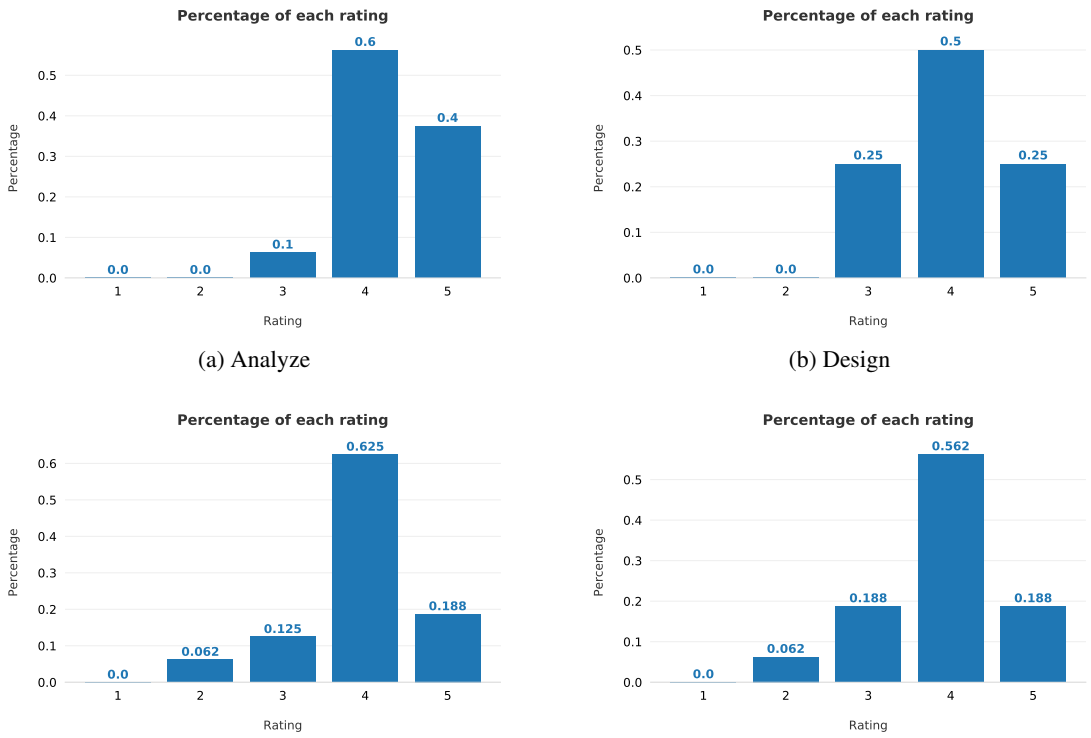
Starting from Figure 8.1, questions $16, 17$ examined whether the course was overly theoretical or practical (respectively), so the fact that they have both received a score of about 3, indicated that the course exhibited a nice balance between the two, not favouring one over the other. The observed difference of about 0.5 could be attributed to the fact that while all the lectures were comprised of both theoretical and technical parts, none of the assignments had theoretical

exercises, which could be perceived as giving a greater emphasis on the technical side, by the students. Among the remaining items, question 4 received the lowest score (which was still significantly better than average), so future implementations of the course could include more practical aspects that would support students' ability to apply taught XAI techniques in their own models. Apart from that, all other questions received a score significantly higher than 4 (as indicated by the limits of the confidence intervals). This suggested high perceived course effectiveness in building an overall understanding of XAI, while it also indicated that students were satisfied by the diversity of the XAI techniques covered in the course. It is worth noting that question 19, which concerned the overall course satisfaction, received an average score of more than 4.5, with the upper bound of the corresponding confidence interval being very close to 5.

Moving on to Figure 8.2, all of the objectives received a high score (significantly higher than 3), with *Analyze* having the highest point estimate, 4.4, providing evidence of the students' confidence in deciding the appropriate XAI techniques for the problem at hand. Both *Design* and *Evaluate* achieved a score of 4, suggesting that students felt comfortable designing pipelines for explaining a model, and interpreting the final results. Finally, *Apply* had a slightly lower average score, as well as a wider confidence interval, implying there was greater variation in the corresponding scores given by the students. In fact, to get an even more detailed picture of the underlying distributions of scores, Figure 8.3 shows a collection of barplots representing the relative frequency of each. For all objectives, 4 was the most common score given by students, which indicated their agreement with the statement that the corresponding objective was met by the course. For the *Evaluate* and *Apply* there was a single student that gave a rating of 2, but otherwise ratings were mostly on the high end of the spectrum. Having said that, this could be seen as evidence that future implementations of the course would benefit by including additional intermediate assignments, putting more emphasis on practical aspects of XAI.

To further assess the effectiveness of the course, an analysis of the students' performance on the final project was performed. Each submission was evaluated based on the the students' ability to carry out the pipeline shown in the class (data preprocessing, model training, model explanation), as well as the quality of their arguments. Since the project was open-ended, the correction guidelines were that the code should run correctly, the pipeline should be executed reasonably well, and the arguments should be substantial, following the insights gathered from the explanations. Based on that, all students were able to pass the course, demonstrating a sufficient level of competence in performing the aforementioned tasks.

Moreover, about 26% of the students considered both a transparent and a black-box model to address the selected application, although there were no related instructions. This was an indication that (at least a portion of the) students took away the message that black-box models should be used only when achieving significantly better performance than transparent ones. However, since the course was focused on XAI, it was reasonable that most of the students opted for considering just black-box models. Furthermore, about 89% of the students used at least 3 XAI techniques, although 2 were enough to met the project's requirements. This was an




(a) Analyze

(b) Design

(c) Evaluate

(d) Apply

Figure 8.3: The students' answers on how much the course met each of its objectives

indication that students felt confident to inspect a model from multiple angles, using techniques that bring different insights. Among them, about 87% carried out a comparison between importance scores coming from SHAP and those coming from LIME (Ribeiro et al., 2016), which is another popular XAI technique. This was evidence that students took proactive measures to make sure that a feature's importance was robust. This was in tune with the course material, where it was emphasized that due to the underlying approximations, many XAI techniques suffer from stability issues. Finally, 15% of the students received a borderline pass, due to the fact that although they performed the pipeline adequately, when forming their arguments in favour or against retaining a model, they underutilized the obtained insights. Despite that, their arguments were substantial, so they were sufficient to pass the course, however they could have been strengthened by taking into account all the available information.

With respect to the qualitative part of the analysis, the open-ended questions revealed specific aspects of the course that were recognised as respondents' favourite. Eleven respondents answered the question "what was your favourite aspect of the course?". Tutorials were mentioned by five respondents. Students appreciated being able to try out the theoretical course aspects in a practical way using the provided workbooks. For example, S-5 said: *"trying out the techniques in the workbooks"*. S-9 said: *"practical application in lab books"*. S-7 said: *"the workbooks and assignment questions. They had the right mix of theory and practical aspects"*. Five respondents mentioned recorded lectures and tutorials. Students appreciated being able to discuss the course material in the online tutorials and lectures. S-1 put it: *"the*

*discussion and Knowledge exchange during Lectures and Tutorial classes"*. S-2 said: *"the meaningful discussions and open-ended questions"*. S-8 said: *"the ambience of the teams' sessions is done with a "brainstorming" approach which gives us the opportunity to discuss ideas, bring questions from the real world and hear different opinions"*. Respondents also mentioned open-ended questions, sufficient examples, and assignment questions. Overall responses were very positive, for example S-2 said: *"I was very impressed with the structure and delivery of the material...It [the course] made me not only appreciate the XAI fundamentals but the whole approach towards applying ML algorithms...I consider myself very lucky for selecting this course and I believe it has helped me tremendously in my understanding of ML projects."* S-6 said: *"I have understood why the area of XAI techniques has gotten attention and is important to make AI / ML available for general use in the Data Analytics Project."* S-8 reflected: *"The topic of XAI it's very interesting! Thank you for including it in the program and giving us exposure to these approaches."*

## 8.5 Discussion and Implications

Overall, the survey results supported the claim that the course material and its delivery can be highly effective in teaching XAI techniques. Analysis of individual ratings showed that this course was especially useful in promoting understanding of a diverse array of XAI techniques, and their conceptual distinctions, advantages, and disadvantages. The respondents' answers to open-ended questions suggested that interactive and practical aspects of the course were important in the successful process of translating XAI theory into practical skills. The open questions and codebooks were also important parts of the course, especially in combination with the ensuing discussions. The survey results also suggested that the course can be effective in teaching XAI techniques to individuals having no or minimal experience and knowledge about XAI.

Students' performance and forum questions suggested that, besides the technical aspects, such as consideration of Python libraries updates, the course could be improved by providing more support for the output analysis and evaluation aspect of the XAI. Most of the students found this part most challenging. This was also reflected in a slightly lower overall score of the the evaluation learning objective, i.e., the ability to critically reflect on the results from the XAI techniques and suggest how it helps the given context. This could be because the selected datasets were not relevant to students' professional or research interests, however it could also mean that more practical exercises focused on the analysis part should be included in the course. Potentially more practice analysing XAI outputs could lead to a better understanding.

### 8.5.1 Limitations

The scores of the questionnaire were self-reported and reflected the subjective evaluation of respondents' own understanding of the course material. The high evaluations of the course

effectiveness were reflected in the objective assignment performances. Although this course has been delivered to data science experts working within the banking sector, in this chapter only the students' responses were analysed. This limitation prevented evaluation of the generalisability of the course effectiveness across different settings and expertise levels. In the future, further surveys will be conducted to assess the effectiveness of the course for the more experienced data science and ML experts strictly working in the professional setting.

## 8.6 Conclusions

In this chapter, an approach for structuring a course on explainability in machine learning was presented. The aim of designing the course was to provide a formal introduction to the field of explainability. Although advances in XAI come at a rapid pace, the fundamental ideas and objectives are likely to remain the same. The course was primarily designed for industry professionals, data scientists, and students with a programming and data science background. One of the main drivers governing the development of the material was to address the reported misuse XAI techniques in practical applications.

To this end, a combination of diverse XAI techniques was included in the material, focusing on conceptual details and distinctions between different explanation types. Furthermore, each lecture was structured around a putative data scientist and the challenges she might come across when trying to provide explanations for a model's behaviour. Finally, most of the lectures were accompanied by a workbook demonstrating the function and utility of the corresponding XAI technique, in order to allow students to gain some hands-on experience. Students' feedback and performance provided strong evidence that the course was effective in meeting its learning objectives.

# Chapter 9

# Conclusions

Achieving a transparent and responsible integration of AI/ML systems in critical applications, is set to be a decisive factor in ensuring that they have a positive impact on society. This thesis is motivated by such concerns and provides for an attempt at exploring several dimensions of transparency, addressing both technical and social aspects of it. With respect to the former, we consider the problem of establishing connections between TPMs and BNs. TPMs extend traditional BNs, allowing for a simple and unified framework for performing inference, while also potentially leading to reductions in the required time and space. However, this comes at the expense of losing the clear representational semantics of BNs, the very property that has rendered them indispensable in a wide array of high-stakes applications. Following this endeavour, we were able to develop novel algorithms and provide formal proofs that TPMs support several interpretable features, similarly to BNs. Moreover, with respect to the latter, we study the influence of self-confidence, model explanations, and uncertainty estimates on the relationship between humans and automated systems. Finally, we take proactive measures to promote the competent use and understanding of explanations.

## 9.1 Summary of contributions

In Chapter 3, we seek to answer the question of whether it is possible to transform TPMs into alternative graphical models, in a way that uncovers their internal representations. We begin by highlighting the limitations of existing SPN to BN decompilations, and how the resulting graphs may fail to recover important information that is embedded in a SPN. To mitigate this issue, we identify sufficient conditions, as well as utilize alternative interpretations of SPN sum nodes, in order to design the first exact decompilation algorithm. By taking advantage of the SPN's structure and parameters, we prove that the resulting algorithm can always accurately decompile SPNs that compiled from BNs. Furthermore, in deriving this algorithm, we uncover novel insights regarding the connection between the local Markov property in BNs and the distributions represented by SPN sum nodes. Moving on, we consider PSDDs, where we

propose an algorithm that clearly depicts their inner hierarchical structure. This transformation is based on the specific properties of SDDs, which naturally allow for establishing parent child relationships between the internal PSDD rules. We conclud the chapter by showing how the resulting graph can be used to expand the PSDD semantics, allowing to define a valid counterfactual distribution over its internal rules.

Moving to our next question, in Chapter 4 we study the problem of constraint incorporation in SPNs. It is well known that BNs satisfy this property, since adjusting the graph's topology during model specification, allows for easily injecting them with domain knowledge. However, when it comes to SPNs, existing results are limited, while their complex underlying formalism makes it challenging to apply the same arguments as in the BN case. In pursuing this question, we provide formal proofs that it is indeed possible to perform this task in SPNs, too, showing that they support both probabilistic and interventional constraints. This result makes use the SPN's multilinear representation, which allows for identifying a correspondence between incorporating a constraint and satisfying a multivariate system of equations involving the SPN parameters. Furthermore, we retrieve sufficient conditions that guarantee that the resulting systems are solvable, i.e. the constraints can be enforced. We demonstrate the effectiveness of our approach by showing how it can easily recover related existing results, while we also propose two optimization approaches for training SPNs under constraints.

Our last technical contribution is presented in Chapter 5, where we explore whether discriminant SPNs can be used as a skeleton for developing a framework for generating diverse counterfactuals for BN-based classifiers. To answer this question, we take advantage of multilinearity, formulating an ILP program, which we formally prove is guaranteed to output valid counterfactual instances. On top of that, under slight modifications the same framework can be used in order to generate prime implicant explanations, which provide additional insights regarding a model's decision making process. Moreover, since our approach only assumes a multilinear model structure, it is applicable to the whole class of multilinear models. This includes decision trees and random forest, for which we show how they can be incorporated into our proposed framework, as well as that this may potentially lead to an infinite set of counterfactuals, instead of just a single instance. We also draw connections with other related recent approaches, showing that some of them correspond to a special case of our framework. Finally, we empirically demonstrate how diverse counterfactuals can be used to guide uncovering biased model behaviour.

In Chapter 7, we move past the technical side of transparency, studying how various factors influence the relationship between human users and AI models. To this end, we design and implement a behavioural experiment to explore the effect of users' self-confidence, as well as to assess whether there are potential complementary benefits of combining uncertainty estimates and explanations. In particular, we explore how the interaction of user confidence, model confidence, and model assistance affects the accuracy, reliance, understanding, and trust towards a model. The subsequent data analysis provides evidence supporting the view that uncertainty and explanations have indeed complementary functions, with the former being

sufficient for improving accuracy in decision making tasks, and the latter providing unique insights about the model's decision making process. Moreover, we provide evidence that self-confidence has a significant effect on multiple aspects of user behaviour, while we also demonstrate how failing to account for it may distort the interpretation of the final results. In a similar spirit, we discuss potential pitfalls of utilizing reliance indicators to measure trust, raising concerns about their suitability to perform this task.

Finally, following our previous findings, in Chapter 8 we design and evaluate an introductory XAI course. Despite the fact that XAI has been one of the most prominent approaches to satisfy the pressing need of ensuring that AI/ML systems are sensible and align with human expectations, academic resources on the topic have been limited so far. To alleviate this issue, we develop a course discussing the various explanation types and the kind of insights they provide, while also suggesting ways to combine multiple explanations together in order to gain a more well-versed understanding of the underlying model. The resulting course was offered by the University of Edinburgh, at MSc students, during the 2021-2022 academic season. Students' performance on the final assignment, as well as their own feedback suggest that the course is effective in providing a good overview of the field, thus validating our approach.

## 9.2 Future work

There are several future research directions stemming from our work, addressing both technical and social dimensions transparency. When it comes to TPMs, one of the most promising research topics is directly related to the findings in Chapter 3. In particular, SPNs have already found many applications due to their ability to easily define mixtures of distributions (Wang and Wang, 2016; Rathke et al., 2017; Amer and Todorovic, 2015; Stelzner et al., 2019; Zheng and Pronobis, 2019). However, as we discuss in Chapter 3, in general this comes at the cost of interpreting sum nodes as uninformative latent variables. A way around this limitation is to impose a connection between sum nodes and quantities that have meaningful interpretations. For example, in our results this is achieved by interpreting sum nodes as representing a variable. There have already been some early attempts that implicitly make use of this observation, such as (Desana and Schnörr, 2017; Zheng et al., 2018), in order to propose more transparent SPN-like variants, however there is still a lot of room for developing systematic approaches for taking advantage of this property. Research along this line could result into highly flexible, interpretable, tractable models.

Another promising direction is motivated by the results presented in Chapter 4, and concerns the potential of using SPNs as auxiliary models to facilitate constraining neural networks. As seminal works, such as (Xu et al., 2018; Xie et al., 2019), demonstrate, tractable architectures can be used in order to embed deep learning models with symbolic knowledge, for example forcing them to take into account physical constraints when generating routes on a map. Our results provide for a complementary research direction, where instead of deterministic

constraints, SPNs are used to impose probabilistic ones, such as enforcing independence between variables. Based on the findings in Chapter 4, SPNs have the capacity to perform this task, opening the door for further developing hybrid techniques that combine TPMs with neural networks. Additional research topics include investigating ways to generate counterfactual instances for TPMs. In Chapter 5, we make a step in this direction, combining TPMs with flexible distance functions that consider the relative cost of modifying each variable. However, other distance functions, such as those in (Cui et al., 2015), allow for incorporating semantic constraints into the resulting counterfactuals. In principle, such functions should be compatible with our framework, since they are linear, so research along this line would enable probing TPMs for counterfactuals that are constrained across multiple dimensions.

In terms of the transparency aspects considered in Chapters 7,8, there is a number of interesting directions that can be motivated by our contributions. For example, the relationship between a user's self-confidence and application-specific characteristics, such as high-stakes outcomes or time constraints calls for further investigation. Although similar questions have been studied in the context of humans acting as machine operators (Chancey et al., 2017; Miller et al., 2016), they have not received the same attention in decision-making tasks. The same holds true with respect to the way a user's behaviour is shaped through repeated interactions with a model. Exploring the effect of time on the human-AI synergy can lead to more robust findings, analogous to the ones in (Yang et al., 2017; Kraus et al., 2020; Chen et al., 2018), since this is a more naturalistic setting, resembling real-life conditions, so the resulting findings are more likely to translate to practical applications. Moreover, studying the effectiveness of different combinations of uncertainty and explanations makes up for another interesting direction. Our findings indicate that different sources of information elicit different user behaviours, so a natural next step would be to consider the effect of alternative estimates/explanations and combinations thereof. Research along these lines could greatly facilitate identifying the advantages of various ensembles of information, which could in turn have significant implications when designing interfaces to enable users to actively interrogate a model in order to satisfy their explanatory needs (Cheng et al., 2020; Gomez et al., 2020; Hohman et al., 2019; Wexler et al., 2019).

Finally, given the diverse backgrounds and goals of the stakeholders that interact with automated systems, developing specialized courses that are focused on the most prominent models/explanations within a certain application domain, e.g. healthcare, would allow stakeholders to get a more in-depth understanding regarding the nuances of the tools that are relevant to them, compared to more general courses that aim at covering a wide spectrum of techniques.

We believe that all aspects of transparency considered in this thesis may lead to interesting and exciting future developments, which should facilitate the safe and responsible integration of automated systems. In our opinion this is one of the greatest challenges presented to the scientific and sociotechnical communities, and its outcome will decide to a large extent the

imprint of automation. We hope that the contributions contained in this thesis will prove to be useful in this endeavour.

# Appendix A

# Objective Model Understanding Questions

The following are the 9 multiple choice questions that were used to assess participants' objective model understanding. All questions and answers were taken from the test developed in (Wang and Yin, 2021). The correct answers are in red.

## A.1 Global Feature Importance

### A.1.1 Question 1

In general, the value of which feature has the greatest influence on the model's predictions?

| | |
|---|---|
| a) Age | b) Employer |
| c) Education | d) Marital Status |
| e) Occupation | f) Ethnic Background |
| g) Gender | h) Hours-per-week |

### A.1.2 Question 2

In general, the value of which feature has the least influence on the model's predictions?

| | |
|---|---|
| a) Age | b) Employer |
| c) Education | d) Marital Status |
| e) Occupation | f) Ethnic Background |
| g) Gender | h) Hours-per-week |

## A.2 Local Feature Importance

### A.2.1 Question 3

| | Feature values |
|---|---|
| **Age** | 54 |
| **Employer** | Private |
| **Education** | Community College |
| **Marital status** | Married |
| **Occupation** | Sales |
| **Ethnic Background** | White |
| **Gender** | Male |
| **Hours-per-week** | 50 |

Figure A.1: Question 3

For this particular person, the value of which feature had the greatest influence on the model's prediction?

a) Age b) Employer
c) Education d) Marital Status
e) Occupation f) Ethnic Background
g) Gender h) Hours-per-week

## A.3 Counterfactuals

### A.3.1 Question 4

| | Feature values |
|---|---|
| **Age** | 43 |
| **Employer** | Private |
| **Education** | Masters |
| **Marital status** | Separated |
| **Occupation** | Tech Support |
| **Ethnic Background** | White |
| **Gender** | Male |
| **Hours-per-week** | 50 |

Figure A.2: Question 4

Our model currently predicts this person earns more than 100K dollars. If we change only one feature of this profile but leave all other features unchanged, which of the following changes is going to change our model's prediction (i.e., make the model predict that the person earns less than 100K dollars)?

a) Change Age from 43 to 25

b) Change Marital Status from Separated to Married

c) Change Ethnic Background from White to Black

d) Change Gender from Male to Female

### A.3.2 Question 5

| | Feature values |
|---|---|
| **Age** | 40 |
| **Employer** | State Government |
| **Education** | Doctorate |
| **Marital status** | Not Married |
| **Occupation** | Professional Specialty |
| **Ethnic Background** | Black |
| **Gender** | Female |
| **Hours-per-week** | 40 |

Figure A.3: Question 5

Our model currently predicts this person earns less than 100K dollars. If we change only one feature of this profile but leave all other features unchanged, which of the following changes is going to change our model's prediction (i.e., make the model predict that the person earns more than 100K dollars)?

a) Change Age from 40 to 50

b) Change Employer from State Government to Federal Government

c) Change Marital Status from Not Married to Married

d) Change Hours-per-week from 40 to 45

## A.4 Model Simulation

### A.4.1 Question 6

| | Feature values |
|---|---|
| **Age** | 33 |
| **Employer** | State Government |
| **Education** | Masters |
| **Marital status** | Not Married |
| **Occupation** | Professional Specialty |
| **Ethnic Background** | Black |
| **Gender** | Female |
| **Hours-per-week** | 35 |

Figure A.4: Question 6

What do you think our model will predict for this person?

a) The model will predict this person earns Less than 100K dollars

b) The model will predict this person earns More than 100K dollars

### A.4.2 Question 7

| | Feature values |
|---|---|
| **Age** | 41 |
| **Employer** | Private |
| **Education** | Community College |
| **Marital status** | Married |
| **Occupation** | Sales |
| **Ethnic Background** | White |
| **Gender** | Male |
| **Hours-per-week** | 60 |

Figure A.5: Question 7

What do you think our model will predict for this person?

a) The model will predict this person earns Less than 100K dollars

b) The model will predict this person earns More than 100K dollars

## A.5 Error Detection

### A.5.1 Question 8

| | Feature values |
|---|---|
| **Age** | 41 |
| **Employer** | Private |
| **Education** | Doctorate |
| **Marital status** | Separated |
| **Occupation** | Sales |
| **Ethnic Background** | White |
| **Gender** | Female |
| **Hours-per-week** | 50 |

Figure A.6: Question 8

Our model predicts that this person earns Less than 100K dollars. Do you believe this prediction is correct?

a) Yes, I think this prediction is correct

b) No, I think this prediction is wrong

### A.5.2 Question 9

| | Feature values |
|---|---|
| **Age** | 51 |
| **Employer** | State Government |
| **Education** | Highschool Graduate |
| **Marital status** | Married |
| **Occupation** | Executive/Managerial |
| **Ethnic Background** | White |
| **Gender** | Male |
| **Hours-per-week** | 40 |

Figure A.7: Question 9

Our model predicts that this person earns Less than 100K dollars. Do you believe this prediction is correct?

a) Yes, I think this prediction is correct

b) No, I think this prediction is wrong

# Appendix B

# CIs and Comparisons

Here are all the details of the CIs and comparisons that were presented in Chapter 7. All CIs and all p-values have been adjusted using the Bonferroni correction method in order to control the family-wise error rate.

## B.1 CIs for Section 7.3.1

| Condition | Average Accuracy | 95% CI |
|---|---|---|
| **Prediction** | 70.5% | $(64.7, 76.7)$ |
| **Local** | 74.1% | $(69.8, 79.4)$ |
| **Combined** | 77.6% | $(73.4, 81.6)$ |
| **Explanations** | 76.7% | $(73.6, 79.9)$ |

Table B.1: Participants' unassisted accuracy

| Condition | Average Accuracy | 95% CI |
|---|---|---|
| **Prediction** | 77.9% | $(74.7, 80.8)$ |
| **Local** | 78.5% | $(75, 82.3)$ |
| **Combined** | 79.9% | $(76.3, 83.4)$ |
| **Explanations** | 78.3% | $(75.4, 81)$ |

Table B.2: Participants' assisted accuracy

| Condition | Average Difference | 95% CI |
|---|---|---|
| **Prediction** | 7.36% | $(2.45, 11.8)$ |
| **Local** | 4.46% | $(0.44, 7.81)$ |
| **Combined** | 2.23% | $(-1.11, 5.35)$ |
| **Explanations** | 1.56% | $(-1.33, 4.68)$ |

Table B.3: Difference in participants' assisted and unassisted accuracy

| Condition | Model Confidence | Average Difference | 95% CI |
|---|---|---|---|
| **Prediction** | Low | 1.33% | $(-4.46, 6.25)$ |
| **Prediction** | High | 13.39% | $(2.67, 22.32)$ |
| **Local** | Low | 4.91% | $(-2.67, 10.71)$ |
| **Local** | High | 4.01% | $(-1.78, 9.82)$ |
| **Combined** | Low | 3.57% | $(-3.12, 8.92)$ |
| **Combined** | High | 0.89% | $(-4.46, 6.25)$ |
| **Explanations** | Low | 1.78% | $(-2.23, 5.8)$ |
| **Explanations** | Low | 1.33% | $(-2.67, 5.8)$ |

Table B.4: Difference in participants' assisted and unassisted accuracy, with respect to the levels of model confidence

| Condition | Human Confidence | Model Confidence | Average Difference | 95% CI |
|---|---|---|---|---|
| **Prediction** | Low | Low | 5.35% | $(-2.67, 1.25)$ |
| **Prediction** | High | Low | $-2.67\%$ | $(-8.92, 3.57)$ |
| **Prediction** | Low | High | 22.32% | $(10.71, 33.03)$ |
| **Prediction** | High | High | 4.46% | $(-7.14, 14.28)$ |
| **Local** | Low | Low | 8.92% | $(-1.78, 17.85)$ |
| **Local** | High | Low | 0.89% | $(-5.35, 6.25)$ |
| **Local** | Low | High | 10.71% | $(2.67, 17.85)$ |
| **Local** | High | High | $-2.67\%$ | $(-8.03, 2.67)$ |
| **Combined** | Low | Low | 8.92% | $(0.89, 16.07)$ |
| **Combined** | High | Low | $-1.78\%$ | $(-8.92, 4.46)$ |
| **Combined** | Low | High | 7.14% | $(-2.67, 16.96)$ |
| **Combined** | High | High | $-5.35\%$ | $(-10.71, 0.89)$ |
| **Explanations** | Low | Low | 6.25% | $(-2.67, 13.39)$ |
| **Explanations** | High | Low | $-2.67\%$ | $(-5.35, 1.78)$ |
| **Explanations** | Low | High | 8.03% | $(0.0, 16.07)$ |
| **Explanations** | High | High | $-5.35\%$ | $(-9.82, 0.0)$ |

Table B.5: Difference in participants' assisted and unassisted accuracy, with respect to the levels of human and model confidence

## B.2 Effects and comparisons for Section 7.3.2.1

| Factor | df | Statistic | p-value |
|---|---|---|---|
| Condition | 3 | 5.529 | 0.137 |
| Model Confidence | 1 | 5.138 | 0.023 |
| Condition $\times$ Model Confidence | 3 | 17.574 | 0.001 |
| Human Confidence | 1 | 40.17 | $< 0.001$ |
| Condition $\times$ Human Confidence | 3 | 5.255 | 0.154 |
| Model Confidence $\times$ Human Confidence | 1 | 1.344 | 0.246 |
| Condition $\times$ Model Confidence $\times$ Human Confidence | 3 | 2.703 | 0.44 |

Table B.6: ANOVA table for Section 7.3.2.1

| Condition | Average Difference | Statistic | p-value |
|---|---|---|---|
| **Prediction** | 0.15% | 198 | 1 |
| **Local** | 9.87% | 32 | $< 0.001$ |
| **Combined** | 1.41% | 166 | 1 |
| **Explanations** | $-1.65\%$ | 175 | 1 |

Table B.7: Difference in participants' reliance between high and low confidence model predictions

## B.3 Effects and comparisons for Section 7.3.2.2

| Factor | df | Statistic | p-value |
|---|---|---|---|
| Condition | 3 | 10.944 | 0.012 |
| Model Confidence | 1 | 23.015 | $< 0.001$ |
| Condition $\times$ Model Confidence | 3 | 6.765 | 0.08 |
| Human Confidence | 1 | 18.114 | $< 0.001$ |
| Condition $\times$ Human Confidence | 3 | 2.206 | 0.531 |
| Model Confidence $\times$ Human Confidence | 1 | 3.963 | 0.047 |
| Condition $\times$ Model Confidence $\times$ Human Confidence | 3 | 4.788 | 0.188 |

Table B.8: ANOVA table for Section 7.3.2.2

| Human/Model Confidence Contrast | Average Difference | Statistic | p-value |
|---|---|---|---|
| High/High vs High/Low | 6.41% | 1555 | $< 0.001$ |
| High/High vs Low/Low | 7.84% | 1148.5 | $< 0.001$ |
| High/High vs Low/High | 4.72% | 1755 | $< 0.001$ |
| High/Low vs Low/Low | 1.69% | 2683 | 0.975 |
| High/Low vs Low/High | $-1.42\%$ | 2848 | 1 |
| Low/High vs Low/Low | 3.12% | 2443 | 0.218 |

Table B.9: Difference in participants' understanding between the various configurations of human/model confidence

| Condition Contrast | Average Difference | Statistic | p-value |
|---|---|---|---|
| **Explanations** vs **Prediction** | 5.58% | $-1.13$ | 0.5 |
| **Explanations** vs **Local** | 7.83% | $-2.5$ | 0.036 |
| **Explanations** vs **Combined** | 7.97% | $-3.01$ | 0.007 |

Table B.10: Difference in participants' subjective understanding between **Explanations** and the remaining conditions

## B.4 Effects and comparisons for Section 7.3.2.3

| Factor | df | Statistic | p-value |
|---|---|---|---|
| Condition | 3 | 1.862 | 0.601 |
| Model Confidence | 1 | 12.942 | $< 0.001$ |
| Condition $\times$ Model Confidence | 3 | 14.817 | 0.002 |
| Human Confidence | 1 | 46.269 | $< 0.001$ |
| Condition $\times$ Human Confidence | 3 | 0.661 | 0.882 |
| Model Confidence $\times$ Human Confidence | 1 | 1.358 | 0.244 |
| Condition $\times$ Model Confidence $\times$ Human Confidence | 3 | 2.078 | 0.556 |

Table B.11: ANOVA table for Section 7.3.2.3

| Condition | Average Difference | Statistic | p-value |
|---|---|---|---|
| **Prediction** | $-1.64\%$ | 153 | 1 |
| **Local** | 6.34% | 88 | 0.035 |
| **Combined** | 2.1% | 160 | 1 |
| **Explanations** | 8.37% | 77 | 0.016 |

Table B.12: Difference in participants' trust between high and low confidence model predictions

## B.5 CIs for Section 7.3.3

| Contrast | Aspect | Average Difference | 95% CI |
|---|---|---|---|
| **Local** vs **Prediction** | Global feature importance | $-0.143$ | $(-0.35, 0.1)$ |
| **Combined** vs **Prediction** | Global feature importance | 0 | $(-0.28, 0.28)$ |
| **Explanations** vs **Prediction** | Global feature importance | 0.964 | $(0.64, 1.28)$ |
| **Local** vs **Prediction** | Local feature importance | 0 | $(0, 0)$ |
| **Combined** vs **Prediction** | Local feature importance | 0 | $(0, 0)$ |
| **Explanations** vs **Prediction** | Local feature importance | 0.643% | $(0.46, 0.82)$ |
| **Local** vs **Prediction** | Counterfactuals | $-0.036$ | $(-0.42, 0.35)$ |
| **Combined** vs **Prediction** | Counterfactuals | $-0.143$ | $(-0.46, 0.17)$ |
| **Explanations** vs **Prediction** | Counterfactuals | 0.893 | $(0.53, 1.25)$ |
| **Local** vs **Prediction** | Model simulation | 0.25 | $(-0.35, 0.85)$ |
| **Combined** vs **Prediction** | Model simulation | 0.143 | $(-0.32, 0.60)$ |
| **Explanations** vs **Prediction** | Model simulation | 0.5 | $(0.07, 0.92)$ |
| **Local** vs **Prediction** | Error detection | 0.25 | $(-0.25, 0.75)$ |
| **Combined** vs **Prediction** | Error detection | 0.036 | $(-0.39, 0.46)$ |
| **Explanations** vs **Prediction** | Error detection | $-0.036$ | $(-0.5, 0.42)$ |

Table B.13: Difference in participants' objective model understanding

# Appendix C

# Course structure and content

In what follows, we outline the structure of the course in Chapter 8, which assumes a basic understanding of machine learning and/or data analytics, as well as basic programming skills in python. When delivering the course, technical details were discussed right after motivating the need of the corresponding technique. The remaining of this section follows Jane, a putative data scientist, in her quest to gain a deeper understanding of the decision-making process underlying a model's outcomes.

**ML preface**

The lecture starts with introducing Jane, a data scientist working in bank, that has been tasked with developing a ML model to automate the process of approving a loan. However, due to the sensitive nature of this application, she knows that performance alone is not enough, she needs to be able to gain insights regarding the model's internal reasoning. After thinking about it, she realizes she has two main options:

- She can go for a transparent model, resulting in a clear interpretation of the decision boundary, allowing for immediately inspecting how a decision is made. For example, if using logistic regression, the notion of defaulting can seen as a weighted sum of features, so a feature's coefficient will tell you this feature's impact on predicting a loan default.
- Otherwise, she can go for an opaque model, which usually achieves better performance and generalizability than its transparent counterparts. Of course, the downside is that in this case is it will not be easy to interpret the model's decisions.

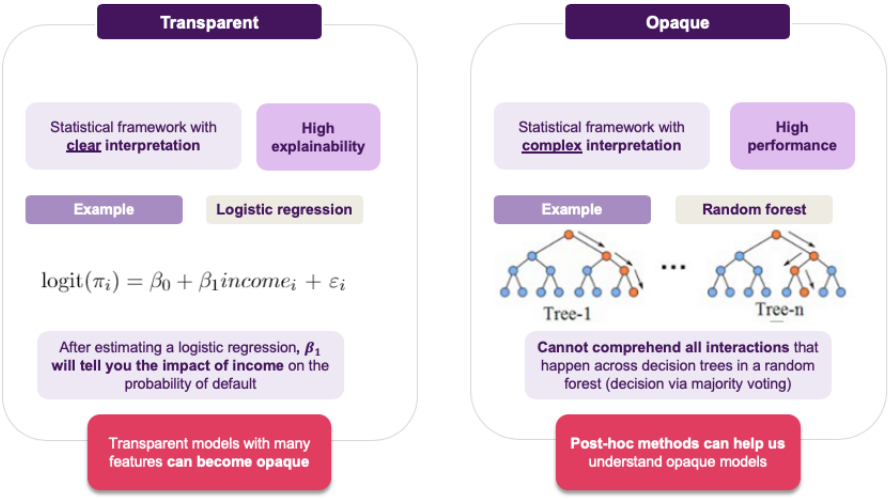



Figure C.1: Jane's choices: should she go for a transparent model or an opaque one?

After giving various transparent models (such as linear models, k-NN classifiers, and decision trees) a try, the resulting accuracy is not satisfactory, compared to the accuracy black-box models (like random forests, neural networks, and support vector machines) achieve. Among them, she finds out that random forests have the best performance, so this is what she will use. The downside is that the resulting model is not immediate to explain anymore (cf. Figure C.2).

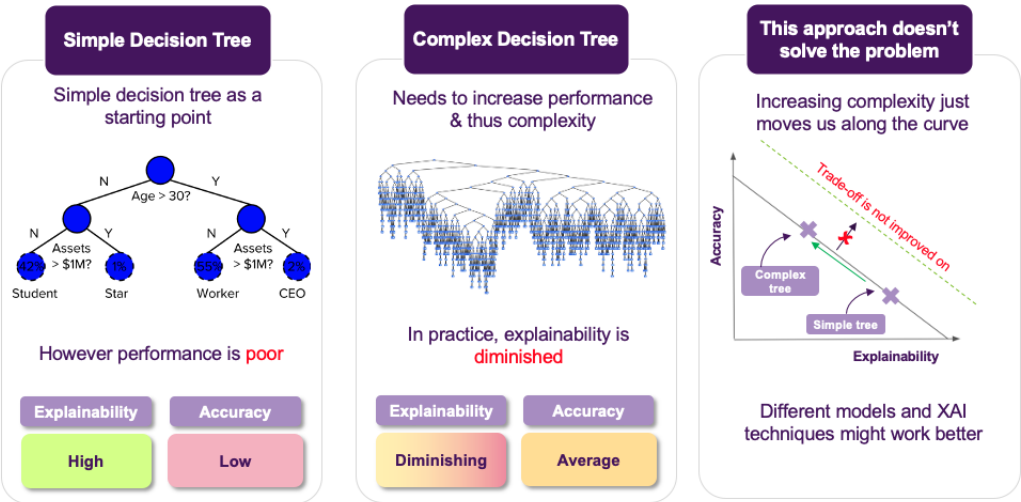



Figure C.2: As transparent models become increasingly complex they may lose their explainability features. The primary goal is to maintain a balance between explainability and accuracy. In cases where this is not possible, opaque models paired with post-hoc XAI approaches provide an alternative solution.

Throughout this lecture, there should be a running discussion about the technical details that render a model to be thought of as transparent or black-box, since the goal here is to highlight which properties make a model more understandable to a human, emphasising intuition. This serves as a first step towards understanding the necessity of developing XAI techniques in

order to enhance black-box models with interpretable features, so they resemble their transparent counterparts.

Open ended questions, such as those below, are presented to students for debate and discussion; such questions are discussed in each lecture:

- Can you think of a reason for why opaque models, often, have better performance than transparent models?
- Can you think of cases when opaque models are outperformed by transparent models? (Hint: consider relational data.)
- Can you argue about why we may want to solve a ML challenge using both transparent and opaque modelling?

By encouraging students to think about the pros and cons of the two approaches and provide use cases, they encounter the challenges that arise when deciding whether to employ a transparent or an opaque model.

**XAI preface**

In this lecture, Jane takes her first look into the field of XAI, since she needs generate explanation for her model's decisions. To this end, her first step is to find out more details about the role of XAI as a means to identify issues, such as:

- **User Acceptance:** By providing explanations, users are more likely be satisfied and accept a ML decision.
- **Improving Human Insight:** Beyond just using ML to perform automation tasks, scientists can use ML for research purposes with respect to big data. An intelligible model can provide information to scientists based on the data being modelled.
- **Legal Imperatives:** Using ML to assess legal liability is a growing issue, as auditing situations to determine liability requires clear explanations from a model's decision. The European Union's GDPR legislation decrees citizens' right to an explanation further strengthens the need for intelligible models.

Open ended questions to present at the end of the lecture include:

- Can you argue how XAI differs from standard criteria that assess the "goodness" of a ML model?
- If a medical system offers 98% accuracy over a transparent model that only offers 88% accuracy, what might you prefer, and why?

- Would an ensemble of different transparent models be considered transparent?

Again, these questions can motivate discussions about timely topics in the deployment of ML, as well as help students appreciate the difference between standard performance measures and explainability.

**SHAP**

In this lecture, Jane decides to apply her first XAI technique. Looking at the available open-source implementations, she decides to utilize SHAP, which is arguably one of the most popular and widely used approaches. She goes on applying SHAP to explain a specific decision made by the model. At this point, there should be a discussion about the technical details and theoretical motivation of SHAP. After this is done, returning to the Jane example, she computes the importance of each feature and shares it with the stakeholders that are responsible for confirming that a model is ready for deployment. Figure C.3 shows an example of the outcome that results when using SHAP. The lecture is concluded with a discussion about the information displayed on this plot, followed by the corresponding open-ended questions.

From a pedagogical perspective, introducing SHAP comes with a number of benefits, such as:

- It exemplifies how well established mathematical ideas can be adjusted to take on new problems, demonstrating the multidisciplinary nature of ML related research.
- Another benefit is that Shapley values are known to satisfy some important properties, allowing for a discussion focused on why these properties are important when generating explanations, or why it is important to have such theoretical guarantees.
- A recent line of research has explored ways to design models that are "immune" to Shapley values explanations, in the sense that although they might be heavily biased, SHAP fails to uncover this behaviour (Slack et al., 2020). Raising this issue can motivate a discussion about possible pitfalls of XAI techniques, as well as the need to examine a model from various perspectives, instead of utilizing only a single technique.
- The current implementation of the SHAP python module comes with an array of different visualisations, which the students can inspect in order to strengthen their understanding of SHAP.
- Different choices in how to define $f_S(x_S)$ lead to different variants of SHAP (Lundberg and Lee, 2017), which, again, can motivate a discussion about the impact as well as the consequences of each choice, emphasizing the need not to treat XAI techniques as black box solutions, but rather making informed choices based on the application at hand.

Open ended questions include:

- What are the trade offs and differences between TreeSHAP and KernelSHAP?
- How could a biased trained model "trick" SHAP by hiding its bias, i.e. assign Shapley values to protected features that do not match their actual importance in the model's decision?
- With KernelSHAP, the sampling for missing values assumes feature independence, is there a way to remedy this issue? Can you think of possible solutions?

These questions aim at motivating a discussion around the assumptions and computational aspects of evaluating SHAP values. They require students to think about the differences between sampling from the marginal and the conditional distributions, as well as the computational complexity they induce. This question has also been considered in recent academic works (Van den Broeck et al., 2020), so interested students can also look into them, as well as alternative strategies to compute Shapley values (Castro et al., 2009).

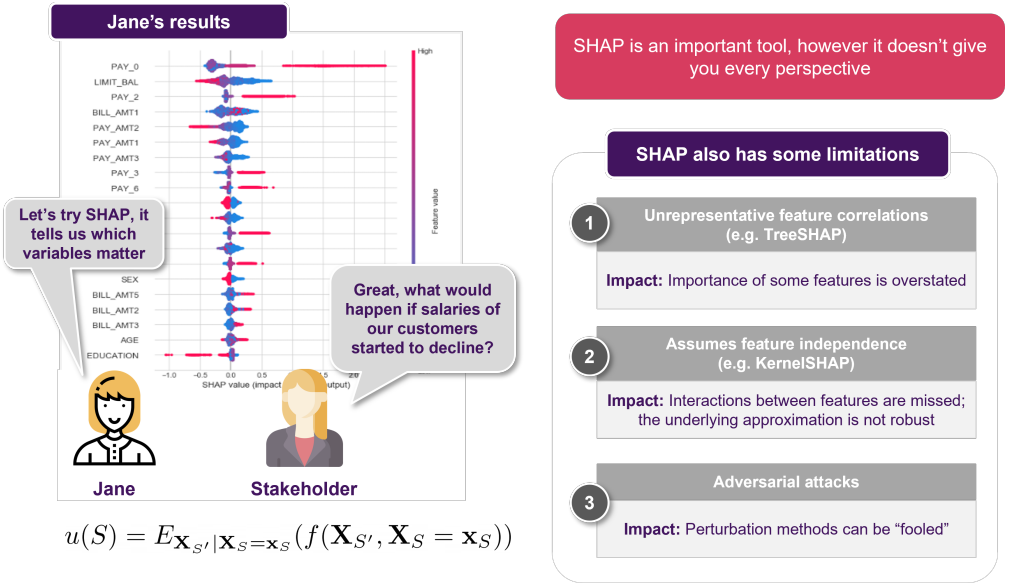



Figure C.3: Jane decides to use SHAP, but cannot resolve all of the stakeholder's questions. Its also worth noting that although SHAP is an important method for explaining opaque models, users should be aware of its limitations, often arising from either the optimization objective or the underlying approximation.

### PDP/ICE

In the previous lecture, Jane generated some importance scores using SHAP, but upon further inspection the stakeholders came up with a reasonable question: could it be that the model relies heavily on an applicant's salary, for example, missing other important factors? How would the model perform on instances where applicants have a relatively low salary? For example, assuming that everything else in the current application was held intact, what is the salary's threshold that differentiates an approved from a rejected application?

These questions cannot been addressed using SHAP, since they refer to how the model's predictive behaviour would change, where SHAP can only explain the instance at hand, so Jane realises that she will have to use additional techniques to answer these questions. To this end, she decides to employ Individual Conditional Expectation (ICE) plots to inspect the model's behaviour for a specific instance, where everything except salary is held constant, fixed to their observed values, while salary is free to attain different values. She could also complement this technique using Partial Dependence Plots (PDPs) to plot the model's decision boundary as a function of the salary, when the rest of the features are averaged out. This plot allows her to gain some insights about the model's average behavior, as the salary changes (Figure C.4). At this point, the technical details of PDP/ICE should be introduced, leading to the open-ended questions.

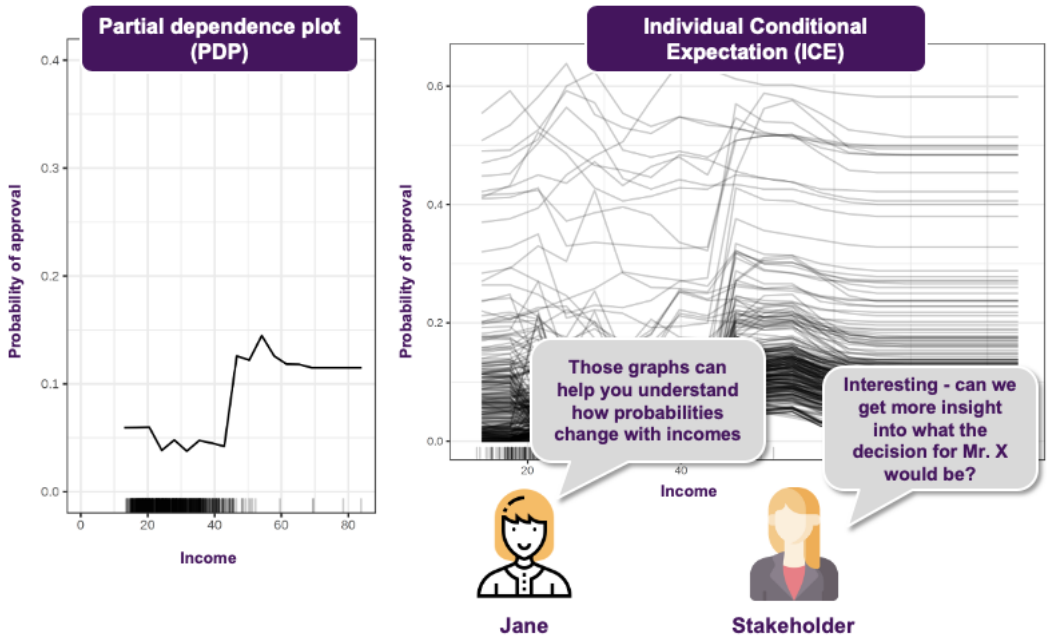



Figure C.4: Visualizations can facilitate understanding the model's reasoning, both on an instance and a global level.

Open ended questions include:

- What are the key limitations of PDP/ICE?
- Roughly sketch a 3-dimensional PDP and ICE plot. Based on that, argue about whether this makes explaining the model easier or not.
- Instead of averaged out values, how can you show the minimum and maximum for the features in PDP? Is that useful?

This time the questions are more focused on the limitations of PDP/ICE, as well as possible ways to manipulate them in order to produce new visualizations. Having a discussion around

these topics can help students identify alternative visualizations that remedy some of these issues, for example ALE plots. Furthermore, coming up with ways to modify the existing plots to convey different kind of information can instil confidence into the students that they have grasped the important details of PDP/ICE.

### Counterfactuals

As before, Jane discusses her new results with the stakeholders, explaining how these plots provide answers to the questions that were raised, but this time there is a new issue to address. In the test set there is an application that the model rejects, which comes contrary to what various experts in the bank think should have happened. This leaves the stakeholders in question of why the model decides like that and whether a slightly different application would have been approved by the model. Jane decides to tackle this using counterfactuals. She applies this approach and she finds out that it was the fact that the applicant had missed one payment that led to this outcome, and that had he/she missed none the application would had been accepted (Figure C.5).

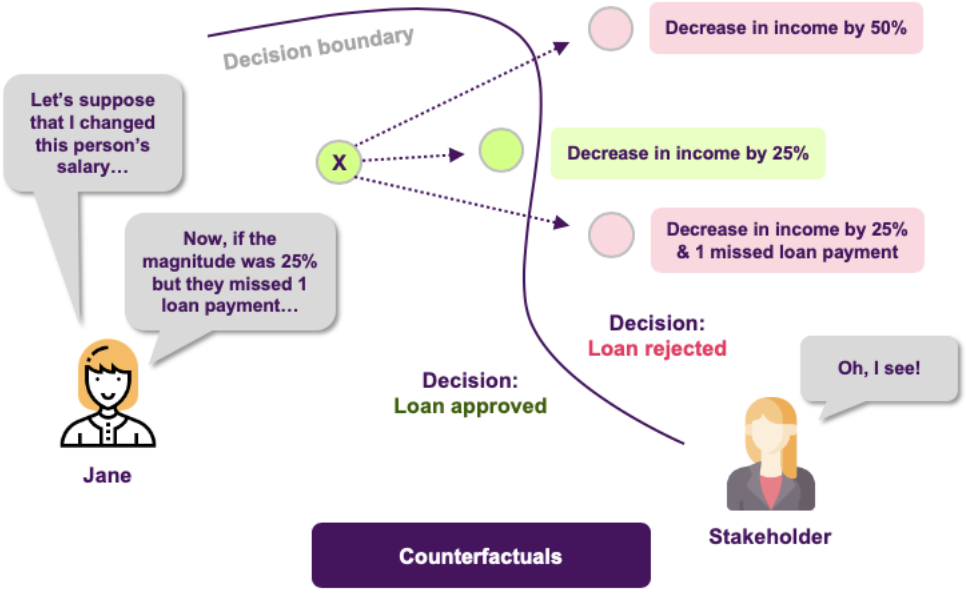



Figure C.5: Counterfactuals produce a hypothetical instance, representing a minimal set of changes of the original one, so the model classifies it in a different category.

At this point, the theoretical details of how counterfactuals are computed should be discussed, so the students get a sense of potential pitfalls and limitations that may arise when empoying them in practical applications. Introducing counterfactuals comes with certain benefits, such as:

- They provide an entry point for drawing connections with concepts from causal inference (CI). CI is expected to be one of the most promising future research directions, but it is often challenging for students to grasp the underlying concepts. However, the

notion of counterfactuals used in XAI is a simplified version of the ones in CI, so it is possible to build on them in order to facilitate the understanding of more advanced ideas.

- In addition, discussing the progression from the initial work in (Wachter et al., 2018) to more recent advances demonstrates how XAI is a dynamic field, where a technique can be refined by taking into account new requirements or desiderata. This could help develop students' critical thinking, enabling them to identify the reasons behind such progressions happen.
- Finally, counterfactuals showcase the interplay between XAI and other domains, such as fairness in AI, or applications, like model debugging, all of which exemplifies the interdisciplinarity of XAI related research. For example, by probing a model through generating multiple counterfactuals, we can examine whether changes on sensitive attributes (such as gender) may lead to the model producing a different outcome. If this is the case, then this is a clear indication that the model exhibits biased behaviour.

Open ended questions this time include:

- Can you get multiple counterfactuals for a given instance? If yes, how should we interpret them?
- How can we handle discrete features?
- How would counterfactuals work with image data?

These questions aim at motivating a discussion about the interpretation of counterfactuals, as well as more conceptual issues. For example, the last question prompts the students to think whether it makes sense to consider single pixel perturbations within an image as a meaningful way to uncover information. Informing a person that the model's decision can be changed if a small number of pixels take on a different value is probably not helpful, so, instead, it would be more meaningful to look for counterfactuals with repsect to objects and shapes within an image, as opposed to pixels.

### Anchors

In this lecture, the stakeholders think that the counterfactuals provide some insights that seem reasonable, but now that they see how influential the number of missed payments is, they think that it would be nice to be able to extract some kind of information explaining how the model operates for instances that are similar to the one under consideration, for future reference.

Jane thinks about it and she decides to use anchors in order to achieve just that, generate easy-to-understand "if-then" rules that approximate the opaque model's behaviour in a local area (Figure C.6). The resulting rules would now look something like "if salary is greater than $20k$ £ and there are no missed payment, then the loan is approved."

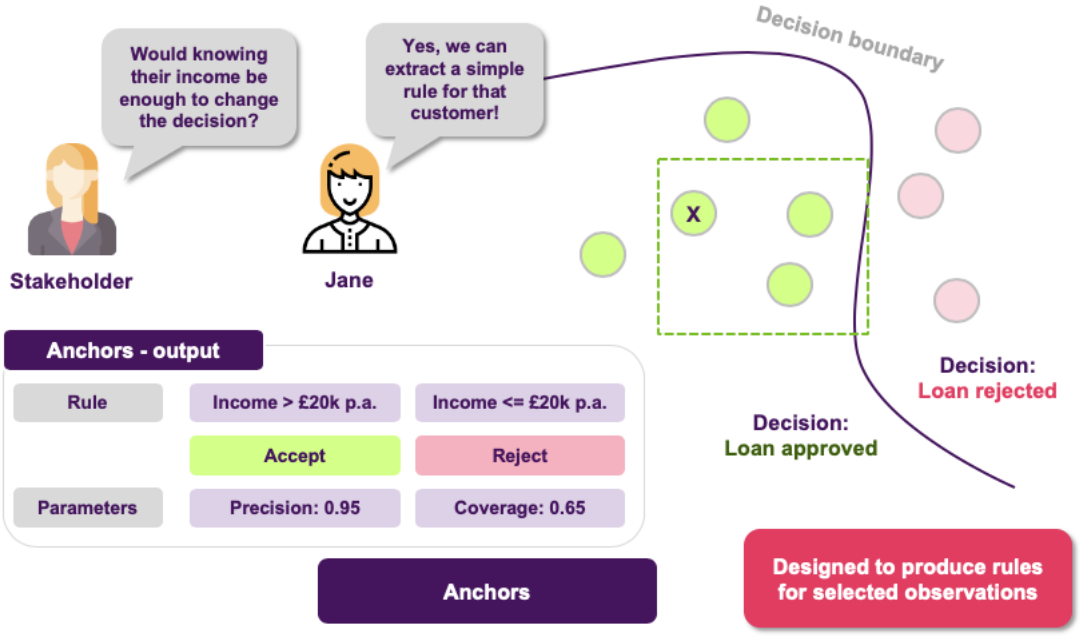



Figure C.6: Local explanations as rules. High precision means that the rule is robust and that similar instances will get the same outcome. High coverage means that large number of the points satisfy the rule's premises, so the rule "generalizes" better.

The lecture should proceed with a discussion focused on the technical details of anchors, as well as the interpretation of precision and coverage, leading up to the open-ended questions.

Open ended questions include:

- What would you choose between anchors with high precision and low coverage vs anchors with low precision and high coverage?
- Can you give examples of rules that might apply to recent data you have encountered? Can you argue about what precision/coverage you expect them to have?
- Compare anchors to other local explainability techniques, such as SHAP. What are the advantages and disadvantages compared to it?

The questions above can motivate a discussion about the utility of anchors under different conditions, as well as a comparison between rules and other forms of explanations, such as importance scores. This gives students the chance to compare different techniques and then potentially reach the conclusion that the answer to this depends largely on the audience the explanation is intended for. This is an important topic that has gained a lot of traction within the academic community, so students can familiarize themselves with timely topics, as well as deepen their understanding on communicating explanations (Bhatt et al., 2020b).

### Deletion diagnostics

Following the findings gathered using the techniques in the previous lectures, the stakeholders are happy with both the model's performance and the degree of explainability. However, upon further inspection, they found out that there are some data points in the training dataset that are too noisy, probably not corresponding to actual data, but rather to instances that were included in the dateset by accident. They turn to Jane, in order to get some insights about how deleting these data points from the training dataset would affect the models behaviour. This is exactly why deletion diagnostics have been developed for, so Jane decides to utilize this approach to answer this question. As it turns out, deletion diagnostics show that omitting these instances would not affect the models performance (Figure C.7).

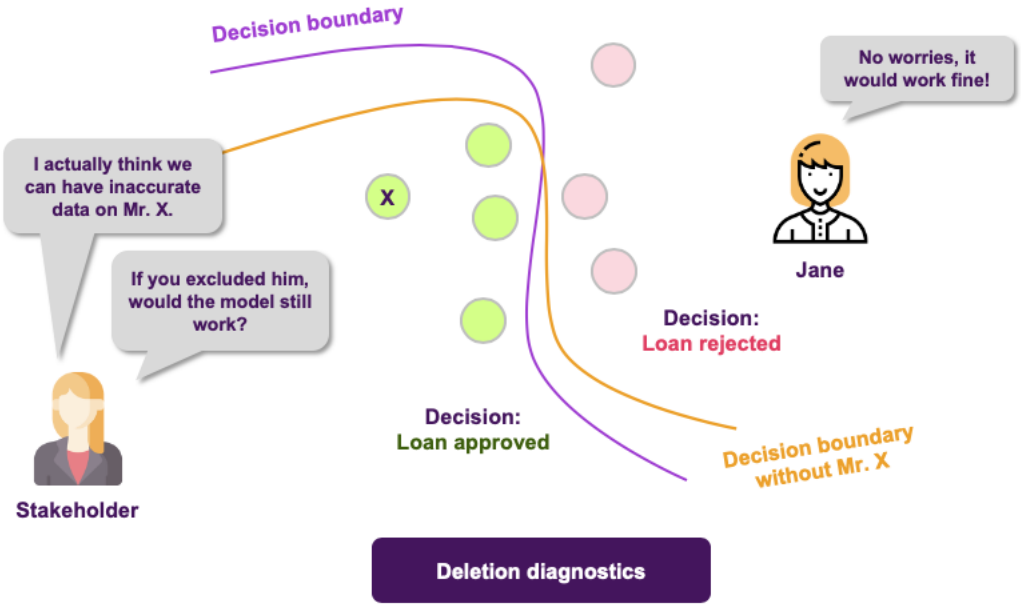



Figure C.7: The quality of a ML model is vastly affected by the quality of the data it is trained on. Finding influential points that can, for example, alter the decision boundary or encourage the model to take a certain decision, contributes in having a more complete picture of the model's reasoning.

At this point there should be a discussion about traditional approaches to compute quantities related to deletion diagnostics, as well as contemporary approaches, especially influence functions, that greatly reduce the resources needed for computing the same quantities for modern ML models.

Open ended questions include:

- Can you explain how deletion diagnostics can be done efficiently without retraining?
- Do you think deletion diagnostics can be applied to random forests?

- What do you think is the relative usefulness of deletion diagnostics compared to influence functions?

This time the questions are mostly on the practical side, aiming at motivating a discussion around the differences of deletion diagnostics and influence functions. This encourages students to think about scenarios where, for example, only one of them is applicable, as well as the advantages and disadvantages of employing each technique.

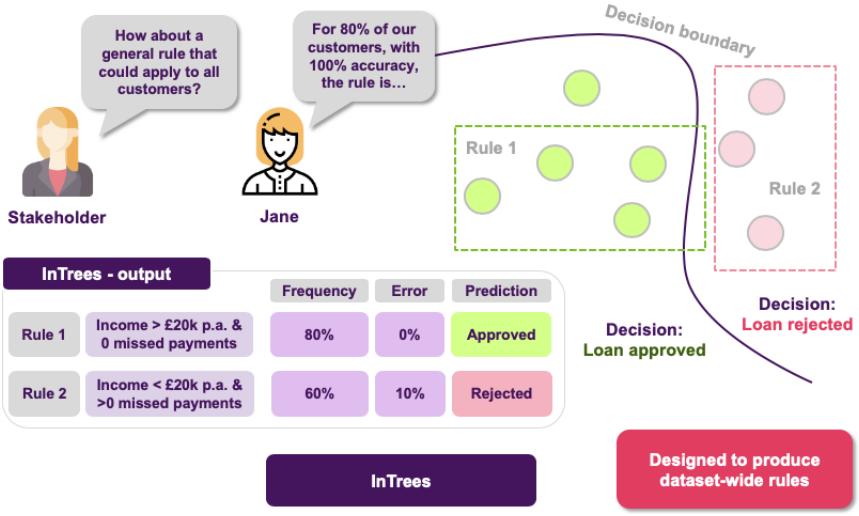



Figure C.8: Extracting rules from a random forest. Frequency of a rule is defined as the proportion of data instances satisfying the rule condition. The frequency measures the popularity of the rule. Error is defined as the number of instances that are incorrectly classified by the rule. So she is able to say that 80% of the customers satisfy the rule "if income >20k and there are 0 missed payments, the application is approved" with 100% accuracy (ie. 0% error).

### InTrees

The last XAI technique discussed in the lectures is InTrees. This is motivated by the stakeholders asking Jane if it is possible to have a set of rules describing the model's behaviour on a global scale, so they can inspect it to find out whether the model has picked up any undesired functioning. At this point, Jane thinks that this is generally a hard task, but since she uses a Random Forest, she could potentially go for a technique that takes this into account, utilizing the underlying ensemble of Decision Trees to generate accurate global rules. This is because since a Decision Tree already consists of a number of rules, it should be possible to extract and combine the most robust rules in the ensemble. Indeed, this is exactly the motivation behind InTrees, extracting, ranking, pruning, and summarizing the collection of rules generated by the ensemble of Decision Trees. (Figure C.8).

After introducing the technical details of this technique, there should be a discussion focused on the previous observation, and how it perfectly captures the utility of model-specific techniques; instead of relying on universal approximations, develop alternatives that take

advantage of the specific characteristics of the model at hand. The majority of model-agnostic approaches make significant assumptions about the underlying model, which are often violated, compromising the quality of the resulting explanations. Consequently, one of the main drives of model-specific explanations is to reduce the number of assumptions, leading to more accurate results. Introducing InTrees has the benefit of clearly demonstrating the concept in a simple way, as opposed to more mathematically challenging alternatives, for example neural network LRP explanations (Bach et al., 2015). This should improve the students' understanding of why model-specific explanations are important, without obscuring the message with overly complex technical details.

Open ended questions include:

- What is the tradeoff between frequency and error in practical scenarios? Which should we aim to optimize?
- Can you argue with examples about what happens if pruning is not applied?
- Can a similar idea be applied to non-tree ensembles (e.g. SVM, neural networks)? If so, how do you think this would be possible?

This time the questions touch upon both practical issues of deploying InTrees and more conceptual aspects. In particular, the last question can motivate a discussion around the features of tree ensembles that make them suitable for developing techniques such as InTrees. Contrasting them to different models can highlight their dissimilarities, as well as how properties that can be found in these models may lead to new approaches. For example, while neural networks do not have internal splitting rules, their layered structure allows for back-propagating messages that quantify the importance of each feature, such as in LRP (Bach et al., 2015).

**Future research directions**

The final lecture of the course is about the future of XAI related research. Its goal is to discuss limitation of XAI and prepare students for the next generation of techniques, as well as to provide an overview of which concepts are likely to play a central role in the future. This way the interested students have the chance to study these concepts in advance, so they have the prior knowledge required to grasp future techniques. Of course, it is not possible to be exhaustive and cover all directions, instead we provide an indicative list of current XAI limitations.

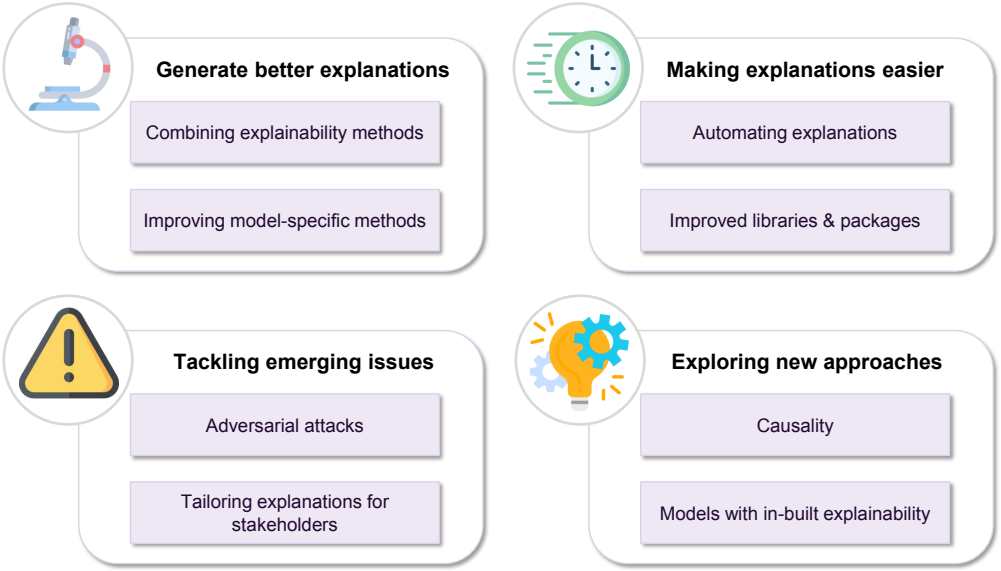



Figure C.9: Possible avenues for XAI research.

An important point of emphasis for students is to realize that limitations arise both on a technical and a practical level. Most of the existing techniques, especially model-agnostic ones, require resorting to approximations. This means that there is always the danger that the resulting explanations might not be accurate, or even be misleading. Furthermore, existing approaches are not really able to identify spurious correlations and report them back. Due to this, it is possible for features to look like they have a strong influence on each other, when, in reality, they only correlate due to a confounder. A possible resolution to these issues could be introducing more concepts from causal analysis, which is already a major drive in related areas, such as fairness in ML. For example, if an explanation was accompanied by a causal model it would not be difficult to check for any spurious correlations.

On a more practical level, developing XAI pipelines to explain a model is still an open research question. Currently, there is no consensus regarding neither the characteristics of a good explanation, nor the way of combining existing techniques in order to adequately explain a model. While there is some overlap between the various explanation types, for the most part they appear to be segmented, each one addressing a different question. This hinders the development of pipelines that aim at automating explanations, or even reaching an agreement on how a complete explanation should look like. On top of that, it is not clear whether explanations should be *selective* (focus on primary causes of the decision making process), or *contrastive* (indicate why a model made decision X, and provide justification for deciding X rather than Y), or both, and how to extract such information from current techniques. Audiences interacting with XAI may include experts in the field, policy-makers, or end users with little ML background, so intelligibility should be varied in its explanations depending on the knowledge level and objectives of the audience. Interdisciplinary research combining

psychology, sociology, and cognitive sciences can help XAI in delivering appropriate explanations (Miller, 2019).

Open ended questions include:

- What do you think are the most important limitations of XAI?
- Can you suggest ways to automate XAI?
- Can you suggest ways to address the potential dangers of transparency?

These questions give students the chance to express their views on how to take on these challenges. [1] This can motivate a discussion around what they think are the most important topics moving forward, as well as how the future of XAI might look like.

[1] Articles that address related questions can also be found in the literature, for example (Weller, 2019), which interested students are pointed to, so they can study a more in-depth analysis.

# Appendix D

# Tutorials and Assignments

The course in Chapter 8 includes 4 jupyter notebook (Kluyver et al., 2016) tutorials, covering SHAP, Counterfactuals, Anchors, and InTrees (PDP/ICE and deletion diagnostics omitted for simplicity). Each tutorial demonstrates a singular XAI technique, while 2 of them (SHAP and Counterfactuals) come with additional questions that students must analyse and answer for assessment. These questions include technical code based implementations of a specific XAI technique and short answer questions asking for student interpretation of the outputs.

### Tutorials

At the beginning of each tutorial, students need to import the relevant python libraries (`pandas`, `numpy`, etc.) as well as the library associated with the corresponding XAI technique. Following the initial setup, basic ML preprocessing practices such as handling *missing values, visualizations,* and *feature engineering*, are applied.

The next step is *Modelling*, which includes training the model and assessing its performance. Each tutorial uses an arbitrary model, the majority of which are black-box models, and performance is measured by looking at the accuracy, precision, recall, and f1 scores.

The third section of the tutorial is where the corresponding XAI technique is applied. The SHAP and Counterfactual tutorials end with a series of assignment questions which comprise a portion of the submission work. Assignment questions include 2-3 short answer questions regarding the technique and/or potential concerns in applying a given model, followed by 3-4 technical questions where the student will need to apply the XAI technique either on another model or on an augmented dataset. In the remaining of this section, aspects of the SHAP tutorial are presented to give a representative example of the structure/content of each tutorial.

**SHAP:** The first step is to import the relevant library (`!pip install shap`) and after preprocessing the data and training a LGBM model, SHAP is invoked. As LGBM is a tree ensemble method, the model-specific SHAP implementation is utilized (see figure D.1).

Included throughout the tutorial are summary questions, which are designed as general discussions points following the introduction of a ML or XAI technique. Questions $(1-2)$ are specific to assessing the black-box model, while questions $(3-10)$ are specific to SHAP and its implementation. In general, these questions are discussed with students during tutorial sessions.

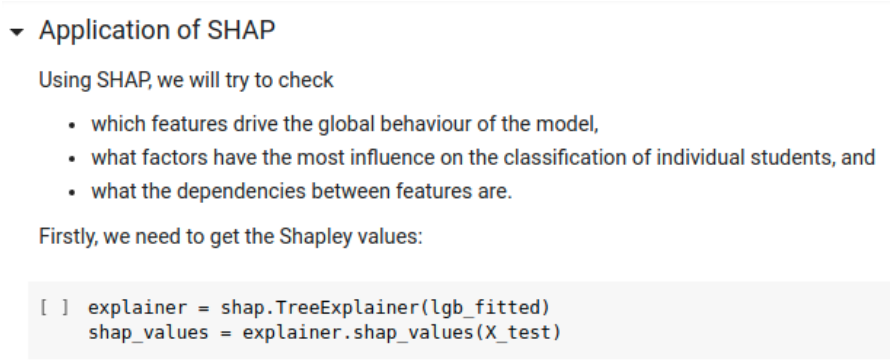



Figure D.1: Demonstration of applying SHAP in the corresponding tutorial.

1. How would you interpret 50% precision?
2. Which metric do you think would be of the most interest to a stakeholder developing a model that aims to predict students' performance?
   - *This summary question brings attention to one of the learning aims, that is these techniques can be used as part of a greater explanatory narrative.*
3. Generate SHAP explanation plots for instances in class 0. What is the link between these graphs and what has been generated for class 1?
4. What are the 5 most important features which are driving this model's decisions?
   - *Here students are asked to read the output of one of the SHAP plots.*
5. How would you interpret the horizontal axis of the a summary plot? What does a SHAP value of $-1.5$ mean?
   - *Students are asked to discuss the characteristics of a SHAP plot and the corresponding values. As there is a bit of a learning curve to understanding such plots, these discussions are important to help students understand the outputs.*
6. Can we express SHAP values in terms of probability? Justify your answer.
   - *A more open-ended question asking students to reflect how SHAP values can be interpreted.*

7. According to the SHAP plots, which student is most likely to fail?
    - *Students are asked to analyse the SHAP outputs for a number of local explanations.*
8. Are there any features which are important for the model, but could indicate some fairness issues?
    - *This question brings the notion of fairness to the students' attention as well as a larger discussion of local explanations.*
9. Produce a SHAP plot for a student (feel free to select your own observation). How different is it from the global plot?
    - *Here students can apply a SHAP plot on an instance to get more familiar with the technique and its implementation.*
10. What does the horizontal spectrum on the top of the SHAP plot show? What do those values mean?
    - *Students are asked to discuss the features of a SHAP plot.*

At the end of the tutorial, the assignment questions are presented to the students. Overall, questions can be separated into technical (coding focused) and short answer. For the SHAP workbook questions $1-5$ are primarily technical, while $6-7$ are short answer ones, requiring students interpreting some outcomes.

1. Use the public dataset introduced in this tutorial and apply an XGBoost model. Your outcome variable will be Portuguese language scores pooled into class 0 and 1. Make predictions on your test set and produce a set of measures that describe the model's performance.
    - *Students are asked to apply a new model, different from the tutorial section of the workbook, and simply assess its performance.*
2. Using SHAP summary plots, what are the 5 most important features in the model?
    - *Here students are asked to apply SHAP and assess which features are the most significant to the model.*
3. Create a decision plot for all observations and all features in your test set, highlight misclassified observations and create decision plots for the set of misclassified observations and for 4 single misclassified observations. Then include force plots for all observations as well as for the set of misclassified observations.
    - *This questions requires students to identify incorrect predictions of the model and then produce a series of SHAP plots for these points. The purpose is to see if any feature-value pairs have a significant correlation with inaccurate predictions.*

4. Make SHAP dependence plots of the 4 most important features. Use sex as the feature possibly influencing SHAP outputs. This done by setting the `interaction_index` as "sex".

    - *Students are now asked to use SHAP plots to see if there is any bias in the model.*

5. In light of the plots from 3 and 4, discuss whether the interaction effect between sex and other features can meaningfully impact the predictions of your model.

    - *The first short answer questions asks students to interpret the outputs from the previous questions. The aim is for the students to list a series of observations based on the generated SHAP plots.*

6. Discuss how various SHAP-based plots can be used in the process of model validation.

    - *A more open-ended short answer question, asking students to discuss the various SHAP graphs and their uses.*

7. Write a paragraph for a non-technical audience explaining how your model makes predictions, based on the SHAP outputs. Ensure the text is clear of jargon!

    - *Here students are expected to write a longer short-answer response, summarizing the previous plots and analysing the results. Importantly, students should tailor their response to a non-expert.*

## Appendix E

# Questionnaire

# XAI Course Feedback Form

1. Which of the following best describes your current status:

   ◯ Research Student (Masters or PhD)

   ◯ Non-Academic Worker

   ◯ Other

2. How did you hear about the XAI course?

   ◯ UoE Email list

   ◯ Word of mouth (friend, family, colleague)

   ◯ Other

3. Please rate your familiarity with the topic of XAI from a score of 1 (not at all familiar) to 5 (very familiar) prior to taking this course

   1 2 3 4 5

   ◯ ◯ ◯ ◯ ◯

4. Please rate your hands-on experience with ML from a score of 1 (no prior experience) to 5 (an expert) prior to taking this course

   1 2 3 4 5

   ◯ ◯ ◯ ◯ ◯

5. Please rate the level of theoretical knowledge of ML from a score of 1 (no prior knowledge) to 5 (an expert) prior to taking this course

   1 2 3 4 5

   ◯ ◯ ◯ ◯ ◯

6. Please rate how confident do you feel in applying the XAI techniques you learned in your own models from a score 1 (not at all confident) to 5 (very confident)

   1 2 3 4 5

   ◯ ◯ ◯ ◯ ◯

7. Please rate how satisfied are you from the diversity of XAI techniques covered in the course from a score 1 (not at all satisfied) to 5 (very satisfied)

   1 2 3 4 5

   ◯ ◯ ◯ ◯ ◯

8. Please rate how much do you feel your understanding of XAI was benefited by the course from a score 1 (not at all) to 5 (very)

   1 2 3 4 5

   ◯ ◯ ◯ ◯ ◯

9. Please rate how much do you feel you have comprehended the conceptual distinctions, advantages, and disadvantages of the XAI techniques covered in the course from a score of 1 (not at all) to 5 (very)

   1 2 3 4 5

   ◯ ◯ ◯ ◯ ◯

10. Please rate each statement from a score of one (strongly disagree) to five (strongly agree)

| | Strongly Disagree (1) | Disagree (2) | Neutral (3) | Agree (4) | Strongly Agree (5) |
|---|---|---|---|---|---|
| The teaching staff made the subject interesting | ○ | ○ | ○ | ○ | ○ |
| The teaching staff made an effort to understand difficulties students were having with their work | ○ | ○ | ○ | ○ | ○ |
| The teaching staff gave helpful feedback | ○ | ○ | ○ | ○ | ○ |
| The materials needed to do the course were readily available | ○ | ○ | ○ | ○ | ○ |
| The materials were relevant to learning in this course | ○ | ○ | ○ | ○ | ○ |
| There was a sufficient variety of learning materials | ○ | ○ | ○ | ○ | ○ |
| It was clear what was expected of me in this course | ○ | ○ | ○ | ○ | ○ |
| I was generally given enough time to understand the thingswe had to learn | ○ | ○ | ○ | ○ | ○ |
| The course was overly theoretical and abstract | ○ | ○ | ○ | ○ | ○ |
| The course was overly practical | ○ | ○ | ○ | ○ | ○ |
| Given the nature of the assignments, the materials, videos and explanations provided were sufficient and accessible | ○ | ○ | ○ | ○ | ○ |
| Overall, I am satisfied with this course | ○ | ○ | ○ | ○ | ○ |

11. Please rate how successful you were in meeting the learning objectives from a score of 1 (not all successful) to 5 (very successful)

| | Unsuccessful | Neither successful nor unsuccessful | Somewhat successful | Successful | Very Successful |
|---|---|---|---|---|---|
| **Analyze:** Describe the context of the machine learning application and why explainability would help, but also scrutinise which kind of explainability technique is necessary. | ○ | ○ | ○ | ○ | ○ |
| **Design**: Define the implementation pipeline for the project: provide a means to clean the data, install and set up one or more post hoc explain ability techniques through aself-chosen set of programming platforms. | ○ | ○ | ○ | ○ | ○ |
| **Evaluation:**Critically reflect on the resultsfrom such techniques and suggest how it helps the problem context. | ○ | ○ | ○ | ○ | ○ |
| **Apply:** Competently apply a wide range of techniques and tools, also knowing their particular features and drawbacks. Have the foundations to understand new and upcoming methods and techniques. | ○ | ○ | ○ | ○ | ○ |

12. What was your favourite aspect of the course?

13. Do you have any suggestions for improving the course in future?

# Bibliography


Barbara D Adams, Lora E Bruyn, Sébastien Houde, Paul Angelopoulos, Kim Iwasa-Madge, and Carol McCann. Trust in automated systems. *Ministry of National Defence*, 2003.

Tameem Adel, David Balduzzi, and Ali Ghodsi. Learning the structure of sum-product networks via an svd-based algorithm. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, page 32–41, Arlington, Virginia, USA, 2015. AUAI Press. ISBN 9780996643108.

Michael Anis Mihdi Afnan, Yanhe Liu, Vincent Conitzer, Cynthia Rudin, Abhishek Mishra, Julian Savulescu, and Masoud Afnan. Interpretable, not black-box, artificial intelligence should be used for embryo selection. *Human reproduction open*, 2021(4):hoab040, 2021.

Icek Ajzen. Understanding attitudes and predicting social behavior. *Englewood cliffs*, 1980.

Igor Aleksander. Partners of humans: a realistic assessment of the role of robots in the foreseeable future. *Journal of Information Technology*, 32(1):1–9, 2017.

Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

Oscar Alvarado and Annika Waern. Towards algorithmic experience: Initial efforts for social media contexts. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–12, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356206. . URL https://doi.org/10.1145/3173574.3173860.

Mohamed R Amer and Sinisa Todorovic. Sum product networks for activity recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(4):800–813, 2015.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. In *Ethics of Data and Analytics*, pages 254–264. Auerbach Publications, 2016.

Anna Markella Antoniadi, Yuhan Du, Yasmine Guendouz, Lan Wei, Claudia Mazo, Brett A Becker, and Catherine Mooney. Current challenges and future opportunities for xai in machine learning-based clinical decision support systems: a systematic review. *Applied Sciences*, 11(11):5088, 2021.

Miller Janny Ariza-Garzón, Javier Arroyo, Antonio Caparrini, and Maria-Jesus Segovia-Vargas. Explainability of a machine learning granting scoring model in peer-to-peer lending. *Ieee Access*, 8:64873–64890, 2020.

Dennis S Arnon, George E Collins, and Scott McCallum. Cylindrical algebraic decomposition i: The basic algorithm. *SIAM Journal on Computing*, 13(4):865–877, 1984.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra Mojsilović, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques, 2019. URL https://arxiv.org/abs/1909.03012.

Maryam Ashoori and Justin D Weisz. In ai we trust? factors that influence trustworthiness of ai-infused decision-making processes. *arXiv preprint arXiv:1912.02675*, 2019.

Francis R Bach and Michael I Jordan. Thin junction trees. In *Advances in Neural Information Processing Systems*, pages 569–576, 2002.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015. . URL https://doi.org/10.1371/journal.pone.0130140.

Nora Balfe, Sarah Sharples, and John R Wilson. Understanding is key: An analysis of factors pertaining to trust in a real-world automation system. *Human factors*, 60(4):477–495, 2018.

Gagan Bansal, Besmira Nushi, Ece Kamar, Eric Horvitz, and Daniel S Weld. Is the most accurate ai the best teammate? optimizing ai for teamwork. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11405–11414, 2021a.

Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2021b.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. GAN dissection: Visualizing and understanding generative adversarial networks. *CoRR*, abs/1811.10597, 2018. URL http://arxiv.org/abs/1811.10597.

Adrien Bennetot, Ivan Donadello, Ayoub El Qadi, Mauro Dragoni, Thomas Frossard, Benedikt Wagner, Anna Saranti, Silvia Tulli, Maria Trocan, Raja Chatila, Andreas Holzinger, Artur d'Avila Garcez, and Natalia Díaz-Rodríguez. A practical tutorial on explainable ai techniques, 2021.

James Bennett, Stan Lanning, and Netflix Netflix. The netflix prize. In *In KDD Cup and Workshop in conjunction with KDD*, 2007.

Michel Besserve, Rémy Sun, and Bernhard Schölkopf. Counterfactuals uncover the modular structure of deep generative models. *CoRR*, abs/1812.03253, 2018. URL http://arxiv.org/abs/1812.03253.

Umang Bhatt, McKane Andrus, Adrian Weller, and Alice Xiang. Machine learning explainability for external stakeholders. *arXiv preprint arXiv:2007.05408*, 2020a.

Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José M. F. Moura, and Peter Eckersley. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 648–657, New York, NY, USA, 2020b. Association for Computing Machinery. ISBN 9781450369367. . URL https://doi.org/10.1145/3351095.3375624.

Umang Bhatt, Javier Antorán, Yunfeng Zhang, Q Vera Liao, Prasanna Sattigeri, Riccardo Fogliato, Gabrielle Melançon, Ranganath Krishnan, Jason Stanley, Omesh Tickoo, et al. Uncertainty as a form of transparency: Measuring, communicating, and using uncertainty. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 401–413, 2021.

Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. 'it's reducing a human being to a percentage': Perceptions of justice in algorithmic decisions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–14, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356206. . URL https://doi.org/10.1145/3173574.3173951.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

Engin Bozdag. Bias in algorithmic filtering and personalization. *Ethics and information technology*, 15(3):209–227, 2013.

Engin Bozdag and Jeroen Van Den Hoven. Breaking the filter bubble: democracy and design. *Ethics and information technology*, 17(4):249–265, 2015.

Philippe Bracke, Anupam Datta, Carsten Jung, and Shayak Sen. Machine learning explainability in finance: an application to default risk analysis. 2019.

Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, August 1996. ISSN 0885-6125. . URL https://doi.org/10.1023/A:1018054314350.

Randal E. Bryant. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Comput. Surv.*, 24(3):293–318, September 1992. ISSN 0360-0300. . URL https://doi.org/10.1145/136035.136043.

Wray L. Buntine. Chain graphs for learning. In *UAI'95*, 1995.

Niklas Bussmann, Paolo Giudici, Dimitri Marinelli, and Jochen Papenbrock. Explainable machine learning in credit risk management. *Computational Economics*, 57(1):203–216, 2021.

Cory J. Butz, Jhonatan de S. Oliveira, and Robert Peharz. Sum-product network decompilation. In Manfred Jaeger and Thomas Dyhre Nielsen, editors, *International Conference on Probabilistic Graphical Models, PGM 2020, 23-25 September 2020, Aalborg, Hotel Comwell Rebild Bakker, Skørping, Denmark*, volume 138 of *Proceedings of Machine Learning Research*, pages 53–64. PMLR, 2020. URL http://proceedings.mlr.press/v138/butz20a.html.

Béatrice Cahour and Jean-François Forzy. Does projection into use improve trust and exploration? an example with a cruise control system. *Safety science*, 47(9):1260–1270, 2009.

Oana-Maria Camburu and Zeynep Akata. Natural-xai: Explainable ai with natural language explanations. International Conference on Machine Learning, 2021. URL https://icml.cc/virtual/2021/tutorial/10835.

Samuel Carton, Qiaozhu Mei, and Paul Resnick. Feature-based explanations don't help people detect misclassifications of online toxicity. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 95–106, 2020.

Andrea Castelletti and Rodolfo Soncini-Sessa. Bayesian networks and participatory modelling in water resource management. *Environmental Modelling & Software*, 22(8):1075–1088, 2007.

Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *Comput. Oper. Res.*, 36(5):1726–1730, may 2009. ISSN 0305-0548. . URL https://doi.org/10.1016/j.cor.2008.04.004.

Eric T Chancey, Alexandra Proaps, and James P Bliss. The role of trust as a mediator between signaling system reliability and response behaviors. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 57, pages 285–289. SAGE Publications Sage CA: Los Angeles, CA, 2013.

Eric T Chancey, James P Bliss, Alexandra B Proaps, and Poornima Madhavan. The role of trust as a mediator between system characteristics and response behaviors. *Human factors*, 57(6):947–958, 2015.

Eric T Chancey, James P Bliss, Yusuke Yamani, and Holly AH Handley. Trust and the compliance–reliance paradigm: The effects of risk, error bias, and reliability on trust and dependence. *Human factors*, 59(3):333–345, 2017.

Eunice Yuh-Jie Chen, Yujia Shen, Arthur Choi, and Adnan Darwiche. Learning bayesian networks with ancestral constraints. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 2333–2341, 2016a.

Min Chen, Stefanos Nikolaidis, Harold Soh, David Hsu, and Siddhartha Srinivasa. Planning with trust for human-robot collaboration. In *Proceedings of the 2018 ACM/IEEE international conference on human-robot interaction*, pages 307–315, 2018.

Ning Chen, Bernardete Ribeiro, and An Chen. Financial credit risk assessment: A recent review. *Artif. Intell. Rev.*, 45(1):1–23, jan 2016b. ISSN 0269-2821. . URL https://doi.org/10.1007/s10462-015-9434-x.

Furui Cheng, Yao Ming, and Huamin Qu. Dece: Decision explorer with counterfactual explanations for machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1438–1447, 2020.

Arthur Choi and Adnan Darwiche. On relaxing determinism in arithmetic circuits. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 825–833, 2017.

Arthur Choi, Andy Shih, Anchal Goyanka, and Adnan Darwiche. On symbolically encoding the behavior of random forests. *CoRR*, abs/2007.01493, 2020. URL https://arxiv.org/abs/2007.01493.

Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, 2017. . URL https://doi.org/10.1089/big.2016.0047. PMID: 28632438.

Brian Christian. *The alignment problem: Machine learning and human values*. WW Norton & Company, 2020.

Angèle Christin. Algorithms in practice: Comparing web journalism and criminal justice. *Big Data & Society*, 4(2):2053951717718855, 2017. . URL https://doi.org/10.1177/2053951717718855.

Eric Chu, Deb Roy, and Jacob Andreas. Are visual explanations useful? a case study in model-in-the-loop prediction. *arXiv preprint arXiv:2007.12248*, 2020.

George E Collins. Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In *Automata theory and formal languages*, pages 134–183. Springer, 1975.

R. Dennis Cook. Detection of influential observation in linear regression. *Technometrics*, 19 (1):15–18, 1977. . URL https://doi.org/10.1080/00401706.1977.10489493.

Gregory F Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial intelligence*, 42(2-3):393–405, 1990.

Andrew Cotter, Maya Gupta, Heinrich Jiang, Nathan Srebro, Karthik Sridharan, Serena Wang, Blake Woodworth, and Seungil You. Training well-generalizing classifiers for fairness metrics and other data-dependent constraints. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1397–1405, 2019a.

Andrew Cotter, Heinrich Jiang, and Karthik Sridharan. Two-player games for efficient non-convex constrained optimization. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, pages 300–332, 2019b.

David Cox, John Little, and Donal OShea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.

Zhicheng Cui, Wenlin Chen, Yujie He, and Yixin Chen. Optimal action extraction for random forests and boosted trees. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 179–188, 2015.

Phil F Culverhouse, Robert Williams, Beatriz Reguera, Vincent Herry, and Sonsoles González-Gil. Do experts make mistakes? a comparison of human and machine indentification of dinoflagellates. *Marine ecology progress series*, 247:17–25, 2003.

Adnan Darwiche. A logical approach to factoring belief networks. In *Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning*, pages 409–420, 2002.

Adnan Darwiche. A differential approach to inference in bayesian networks. *Journal of the ACM (JACM)*, 50(3):280–305, 2003.

Adnan Darwiche. *Modeling and reasoning with Bayesian networks*. Cambridge university press, 2009.

Adnan Darwiche. Sdd: A new canonical representation of propositional knowledge bases. pages 819–826, 01 2011. .

Adnan Darwiche. Causal inference using tractable circuits. *arXiv preprint arXiv:2202.02891*, 2022.

Cassio P De Campos, Zhi Zeng, and Qiang Ji. Structure learning of bayesian networks using constraints. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 113–120, 2009.

Peter De Vries, Cees Midden, and Don Bouwhuis. The effects of errors on system trust, self-confidence, and the allocation of control in route planning. *International Journal of Human-Computer Studies*, 58(6):719–735, 2003.

Rina Dechter. Bucket elimination: A unifying framework for reasoning. *Artif. Intell.*, page 41–85, 1999.

Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial intelligence*, 49(1-3):61–95, 1991.

Houtao Deng. Interpreting tree ensembles with intrees. *arXiv:1408.5456*, 08 2014. .

Mattia Desana and Christoph Schnörr. Sum–product graphical models. *Machine Learning*, 109:135–173, 2017.

Vasant Dhar. Data mining in finance: using counterfactuals to generate knowledge from organizational information systems. *Information Systems*, 23(7):423–437, 1998.

Stephen R Dixon and Christopher D Wickens. Automation reliability in unmanned aerial vehicle control: A reliance-compliance model of automation dependence in high workload. *Human factors*, 48(3):474–486, 2006.

Jonathan Dodge, Q Vera Liao, Yunfeng Zhang, Rachel KE Bellamy, and Casey Dugan. Explaining models: an empirical study of how explanations impact fairness judgment. In *Proceedings of the 24th international conference on intelligent user interfaces*, pages 275–285, 2019.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science advances*, 4(1), 2018.

Bradley Efron and Robert Tibshirani. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical science*, pages 54–75, 1986.

Upol Ehsan, Philipp Wintersberger, Q. Vera Liao, Elizabeth Anne Watkins, Carina Manger, Hal Daumé III, Andreas Riener, and Mark O Riedl. Human-centered explainable ai (hcxai): Beyond opening the black-box of ai. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI EA '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391566. . URL https://doi.org/10.1145/3491101.3503727.

Rasmus Fatum and Michael M Hutchison. Evaluating foreign exchange market intervention: Self-selection, counterfactuals and average treatment effects. *Journal of International Money and Finance*, 29(3):570–584, 2010.

Ad Feelders and Linda C Van der Gaag. Learning bayesian network parameters under order constraints. *International Journal of Approximate Reasoning*, 42(1-2):37–53, 2006.

Rubén R. Fernández, Isaac Martín de Diego, Víctor Aceña, Alberto Fernández-Isabel, and Javier M. Moguerza. Random forest explainability using counterfactual sets. *Information Fusion*, 63:196–207, 2020. ISSN 1566-2535. . URL https://www.sciencedirect.com/science/article/pii/S1566253520303134.

Steven Finlay. Multiple classifier architectures and their application to credit risk assessment. *European Journal of Operational Research*, 210(2):368–378, 2011. URL https://EconPapers.repec.org/RePEc:eee:ejores:v:210:y:2011:i:2:p:368-378.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 29(5):1189–1232, 10 2001. . URL https://doi.org/10.1214/aos/1013203451.

Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine learning*, 29:131–163, 1997.

Tal Friedman and Guy Van den Broeck. On constrained open-world probabilistic databases. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 5722–5729, 08 2019.

Andreas Friis-Hansen. Bayesian networks as a decision support tool in marine applications. 2000.

Iason Gabriel. Artificial intelligence, values, and alignment. *Minds and machines*, 30(3): 411–437, 2020.

Vassilis Galanos. Exploring expanding expertise: artificial intelligence as an existential threat and the role of prestigious commentators, 2014–2018. *Technology Analysis & Strategic Management*, 31(4):421–432, 2019.

Dan Geiger, Thomas Verma, and Judea Pearl. d-separation: From theorems to algorithms. In *Machine Intelligence and Pattern Recognition*, volume 10, pages 139–148. Elsevier, 1990.

R. Gens and P. Domingos. Learning the structure of sum-product networks. In *International Conference on Machine Learning*, 2013.

Tarleton Gillespie. *Trendingistrending: When Algorithms Become Culture*. Routledge, June 2016. URL https://www.microsoft.com/en-us/research/publication/trendingistrending-when-algorithms-become-culture-3/.

Yeow Goh, Xin Cai, Walter Theseira, Giovanni Ko, and Khiam Khor. Evaluating human versus machine learning performance in classifying research abstracts. *Scientometrics*, 125, 07 2020. .

Oscar Gomez, Steffen Holter, Jun Yuan, and Enrico Bertini. Vice: Visual counterfactual explanations for machine learning models. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 531–535, 2020.

Ben Green and Yiling Chen. The principles and limits of algorithm-in-the-loop decision making. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–24, 2019.

Gert-Martin Greuel, Gerhard Pfister, Olaf Bachmann, Christoph Lossen, and Hans Schönemann. *A Singular introduction to commutative algebra*, volume 34. Springer, 2002.

Nina Grgić-Hlača, Muhammad Bilal Zafar, Krishna P. Gummadi, and Adrian Weller. The Case for Process Fairness in Learning: Selection for Fair Decision Making. In *Symposium on Machine Learning and the Law at the 29th Conference on Neural Information Processing Systems*, Barcelona, Spain, 2016.

Patrick Griffin, Hamish Coates, Craig Mcinnis, and Richard James. The development of an extended course experience questionnaire. *Quality in Higher Education*, 9(3):259–266, 2003.

William M Grove, David H Zald, Boyd S Lebow, Beth E Snitz, and Chad Nelson. Clinical versus mechanical prediction: a meta-analysis. *Psychological assessment*, 12(1):19, 2000.

Jan Gugenheimer, Evgeny Stemasov, Julian Frommel, and Enrico Rukzio. Sharevr: Enabling co-located experiences for virtual reality between hmd and non-hmd users. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4021–4033, 2017.

David Gunning and David Aha. Darpa's explainable artificial intelligence (xai) program. *AI magazine*, 40(2):44–58, 2019.

David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. Xai—explainable artificial intelligence. *Science Robotics*, 4(37):eaay7120, 2019.

Weisi Guo. Explainable artificial intelligence for 6g: Improving trust between human and machine. *IEEE Communications Magazine*, 58(6):39–45, 2020.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 3323–3331, USA, 2016.

Jeremy Hartmann, Christian Holz, Eyal Ofek, and Andrew D Wilson. Realitycheck: Blending virtual environments with situated physical reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

Miguel A Hernán, David Clayton, and Niels Keiding. The simpson's paradox unraveled. *International journal of epidemiology*, 40(3):780–785, 2011.

Sarah Hilder, Richard W Harvey, and Barry-John Theobald. Comparison of human and machine-based lip-reading. In *AVSP*, pages 86–89, 2009.

Kevin Anthony Hoff and Masooda Bashir. Trust in automation: Integrating empirical evidence on factors that influence trust. *Human factors*, 57(3):407–434, 2015.

Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. Metrics for explainable ai: Challenges and prospects. *arXiv preprint arXiv:1812.04608*, 2018.

Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, and Steven M Drucker. Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–13, 2019.

Aya Hussein, Sondoss Elsawah, and Hussein A Abbass. Trust mediating reliability–reliance relationship in supervisory control of human–swarm interactions. *Human Factors*, 62(8): 1237–1248, 2020.

Jiun-Yin Jian, Ann M Bisantz, and Colin G Drury. Foundations for an empirically determined scale of trust in automated systems. *International journal of cognitive ergonomics*, 4(1): 53–71, 2000.

Steven A Julious and Mark A Mullee. Confounding and simpson's paradox. *Bmj*, 309(6967): 1480–1481, 1994.

Alan M Kalet, John H Gennari, Eric C Ford, and Mark H Phillips. Bayesian network models for error detection in radiotherapy plans. *Physics in Medicine & Biology*, 60(7):2735, 2015.

Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. Dace: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In *IJCAI*, pages 2855–2862, 2020.

Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, Yuichi Ike, Kento Uemura, and Hiroki Arimura. Ordered counterfactual explanation by mixed-integer linear optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11564–11574, 2021.

Konstantinos V Katsikopoulos, Özgür Şimşek, Marcus Buckmann, and Gerd Gigerenzer. Transparent modeling of influenza incidence: Big data or a single data point from psychological theory? *International Journal of Forecasting*, 38(2):613–619, 2022.

Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. Interpreting interpretability: understanding data scientists' use of interpretability tools for machine learning. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–14, 2020.

C Kelly. Guidelines for trust in future atm systems-principles. 2003.

Hassan Khosravi, Simon Buckingham Shum, Guanliang Chen, Cristina Conati, Yi-Shan Tsai, Judy Kay, Simon Knight, Roberto Martinez-Maldonado, Shazia Sadiq, and Dragan Gašević. Explainable artificial intelligence in education. *Computers and Education: Artificial Intelligence*, 3:100074, 2022. ISSN 2666-920X. . URL https://www.sciencedirect.com/science/article/pii/S2666920X22000297.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Doga Kisa, Guy Van den Broeck, Arthur Choi, and Adnan Darwiche. Probabilistic sentential decision diagrams. In *Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2014.

Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. Human Decisions and Machine Predictions*. *The Quarterly Journal of Economics*, 133(1): 237–293, 08 2017. ISSN 0033-5533. . URL https://doi.org/10.1093/qje/qjx032.

Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press, 2016.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1885–1894. JMLR.org, 2017.

Frank Konietschke, Arne C Bathke, Solomon W Harrar, and Markus Pauly. Parametric and nonparametric bootstrap methods for general manova. *Journal of Multivariate Analysis*, 140:291–301, 2015.

Igor Kononenko. Machine learning for medical diagnosis: history, state of the art and perspective. *Artif. Intell. Medicine*, 23(1):89–109, 2001. . URL https://doi.org/10.1016/S0933-3657(01)00077-X.

Johannes Kraus, David Scholz, Dina Stiegemeier, and Martin Baumann. The more you know: trust dynamics and calibration in highly automated driving and the effects of take-overs, system malfunction, and system transparency. *Human factors*, 62(5):718–736, 2020.

Yoshiki Kudo, Anthony Tang, Kazuyuki Fujita, Isamu Endo, Kazuki Takashima, and Yoshifumi Kitamura. Towards balancing vr immersion and bystander awareness. *Proc. ACM Hum. Comput. Interact.*, 5(ISS):1–22, 2021.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

Vivian Lai and Chenhao Tan. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 29–38, 2019.

Vivian Lai, Han Liu, and Chenhao Tan. " why is' chicago'deceptive?" towards building model-driven tutorials for humans. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

Hima Lakkaraju and Ike Lage. Interpretability and explainability in machine learning, 2019.

Stefan Larsson, Mikael Anneroth, Anna Felländer, Li Felländer-Tsai, Fredrik Heintz, and Rebecka Cedering Ångström. Sustainable ai: An inventory of the state of knowledge of ethical, social, and legal challenges related to artificial intelligence. 2019.

Steffen L. Lauritzen and Thomas S. Richardson. Chain graph models and their causal interpretations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):321–348, 2002.

John Lee and Neville Moray. Trust, control strategies and allocation of function in human-machine systems. *Ergonomics*, 35(10):1243–1270, 1992.

John D Lee and Neville Moray. Trust, self-confidence, and operators' adaptation to automation. *International journal of human-computer studies*, 40(1):153–184, 1994.

John D Lee and Katrina A See. Trust in automation: Designing for appropriate reliance. *Human factors*, 46(1):50–80, 2004.

Stephan Lewandowsky, Michael Mundy, and Gerard Tan. The dynamics of trust: comparing humans to automation. *Journal of Experimental Psychology: Applied*, 6(2):104, 2000.

David Lewis. Causation. *Journal of Philosophy*, 70(17):556–567, 1973. .

David Lewis. Counterfactuals. *Tijdschrift Voor Filosofie*, 36(3):602–605, 1974.

Q Vera Liao and Kush R Varshney. Human-centered explainable ai (xai): From algorithms to user experiences. *arXiv preprint arXiv:2110.10790*, 2021.

Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020.

Michael P Linegang, Heather A Stoner, Michael J Patterson, Bobbie D Seppelt, Joshua D Hoffman, Zachariah B Crittendon, and John D Lee. Human-automation collaboration in dynamic mission planning: A challenge requiring an ecological approach. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 2482–2486. SAGE Publications Sage CA: Los Angeles, CA, 2006.

Tyler Loftus, Patrick Tighe, Amanda Filiberto, Philip Efron, Scott Brakenridge, Alicia Mohr, Parisa Rashidi, Gilbert Upchurch, and Azra Bihorac. Artificial intelligence and surgical decision-making. *JAMA Surgery*, 155, 12 2019. .

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature biomedical engineering*, 2(10):749–760, 2018.

Poornima Madhavan and Rachel R Phillips. Effects of computer self-efficacy and system reliability on user interaction with decision support systems. *Computers in Human Behavior*, 26(2):199–204, 2010.

Maria Madsen and Shirley Gregor. Measuring human-computer trust. In *11th australasian conference on information systems*, volume 53, pages 6–8. Citeseer, 2000.

Basim Mahbooba, Mohan Timilsina, Radhya Sahal, and Martin Serrano. Explainable artificial intelligence (xai) to enhance trust management in intrusion detection systems using decision tree model. *Complexity*, 2021, 2021.

John F Mahoney and James M. Mohen. Method and system for loan origination and underwriting. *US Patent 7,287,008. 1*, 2007.

Pablo Marquez Neila, Mathieu Salzmann, and Pascal Fua. Imposing hard constraints on deep networks: Promises and limitations. *CVPR Workshop on Negative Results in Computer Vision*, 2017.

Jérémie Mary, Clément Calauzenes, and Noureddine El Karoui. Fairness-aware learning for continuous attributes and treatments. In *International Conference on Machine Learning*, pages 4382–4391. PMLR, 2019.

M. McGough. *How Bad is Sacramento's Air, Exactly? Google Results Appear at Odds with Reality, some say*, chapter [online] Available: https://www.sacbee.com/news/california/fires/article216227775.html. 2018.

Craig McInnis. Defining and assessing the student experience in the quality management process. *Tertiary Education & Management*, 3(1):63–71, 1997.

Patrick E McKnight and Julius Najab. Mann-whitney u test. *The Corsini encyclopedia of psychology*, pages 1–1, 2010.

Christopher Meek. Strong completeness and faithfulness in bayesian networks. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, page 411–418, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), July 2021. ISSN 0360-0300. . URL https://doi.org/10.1145/3457607.

Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness  satisfaction in recommendation systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 2243–2251, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. . URL https://doi.org/10.1145/3269206.3272027.

Peter Menzies and Helen Beebee. Counterfactual theories of causation. 2001.

Stephanie M Merritt and Daniel R Ilgen. Not all trust is created equal: Dispositional and history-based trust in human-automation interactions. *Human factors*, 50(2):194–210, 2008.

D. Meyer. *Amazon Reportedly Killed an AI Recruitment System Because It Couldn't Stop the Tool from Discriminating Against Women*, chapter [online] Available: https://fortune.com/2018/10/10/amazon-ai-recruitment-bias-women-sexist/. October 2018.

Giacomo Zambelli Michele Conforti, Gérard Cornuéjols. *Integer programming*. Graduate texts in mathematics, . 271. Springer, Cham, 2014. ISBN 9783319110073.

David Miller, Mishel Johns, Brian Mok, Nikhil Gowda, David Sirkin, Key Lee, and Wendy Ju. Behavioral measurement of trust in automation: the trust fall. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 60, pages 1849–1853. SAGE Publications Sage CA: Los Angeles, CA, 2016.

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019. .

Alan Mishler, Edward H Kennedy, and Alexandra Chouldechova. Fairness in risk assessment instruments: Post-processing to achieve counterfactual equalized odds. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 386–400, 2021.

Dan H. Moore II. Classification and regression trees, by leo breiman, jerome h. friedman, richard a. olshen, and charles j. stone. brooks/cole publishing, monterey, 1984,358 pages, $27.95. *Cytometry*, 8(5):534–535, 1987. . URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cyto.990080516.

Neville Moray, Toshiyuki Inagaki, and Makoto Itoh. Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of experimental psychology: Applied*, 6(1):44, 2000.

Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020*

*Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 607–617, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369367. . URL https://doi.org/10.1145/3351095.3372850.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*, pages 7076–7087. PMLR, 2020.

Nithesh Naik, BM Hameed, Dasharathraj K Shetty, Dishant Swain, Milap Shah, Rahul Paul, Kaivalya Aggarwal, Sufyan Ibrahim, Vathsala Patil, Komal Smriti, et al. Legal and ethical consideration in artificial intelligence in healthcare: who takes responsibility? *Frontiers in surgery*, page 266, 2022.

Iago Paris, Raquel Sanchez-Cauce, and Francisco Javier Diez. Sum-product networks: A survey, 2020.

Urja Pawar, Donna O'Shea, Susan Rea, and Ruairi O'Reilly. Explainable ai in healthcare. In *2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, pages 1–2. IEEE, 2020.

Seyedeh Neelufar Payrovnaziri, Zhaoyi Chen, Pablo Rengifo-Moreno, Tim Miller, Jiang Bian, Jonathan H Chen, Xiuwen Liu, and Zhe He. Explainable artificial intelligence models using real-world electronic health record data: a systematic scoping review. *Journal of the American Medical Informatics Association*, 27(7):1173–1185, 2020.

Judea Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146, 01 2009a.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009b. ISBN 052189560X, 9780521895606.

Judea Pearl. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 62(3):54–60, 2019.

Robert Peharz, Robert Gens, and Pedro Domingos. Learning selective sum-product networks. In *LTPM workshop*, volume 32, 2014.

Robert Peharz, Sebastian Tschiatschek, Franz Pernkopf, and Pedro Domingos. On Theoretical Properties of Sum-Product Networks. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 744–752, San Diego, California, USA, 09–12 May 2015. PMLR.

Robert Peharz, Rüdiger Gens, Franz Pernkopf, and Pedro M. Domingos. On the latent variable interpretation in sum-product networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

Jose M. Pena. Approximate counting of graphical models via mcmc. In *PMLR*, 2007. URL http://proceedings.mlr.press/v2/pena07a.html.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2017. ISBN 978-0-262-03731-0. URL https://mitpress.mit.edu/books/elements-causal-inference.

Christo Wilson Piotr Sapiezynski, Valentin Kassarnig. Academic performance prediction in a gender-imbalanced environment. *FATREC*, 2017.

Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. *Proc. 12th Conf. on Uncertainty in Artificial Intelligence*, pages 337–346, 2011.

Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Wortman Vaughan, and Hanna Wallach. Manipulating and measuring model interpretability. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pages 1–52, 2021.

Asilomar AI Principles. *Principles developed in conjunction with the 2017 Asilomar conference*, chapter [online] Available: https://futureofife.org/ai-principles. 2017.

Mattia Prosperi, Yi Guo, Matt Sperrin, James S Koopman, Jae S Min, Xing He, Shannan Rich, Mo Wang, Iain E Buchan, and Jiang Bian. Causal inference and counterfactual prediction in machine learning for actionable healthcare. *Nature Machine Intelligence*, 2(7):369–375, 2020.

Murray H. Protter and Charles B. Jr. Morrey. *Intermediate Calculus*. Springer, 1985.

J. R. Quinlan. Simplifying decision trees. *Int. J. Man-Mach. Stud.*, 27(3):221–234, September 1987. ISSN 0020-7373. . URL https://doi.org/10.1016/S0020-7373(87)80053-6.

Thomas P Quinn, Stephan Jacobs, Manisha Senadeera, Vuong Le, and Simon Coghlan. The three ghosts of medical ai: Can the black-box present deliver? *Artificial intelligence in medicine*, 124:102158, 2022.

Luc De Raedt, Tias Guns, and Siegfried Nijssen. Constraint programming for data mining and machine learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, 2010.

Paul Ramsden. A performance indicator of teaching quality in higher education: The course experience questionnaire. *Studies in higher education*, 16(2):129–150, 1991.

Fabian Rathke, Mattia Desana, and Christoph Schnörr. Locally adaptive probabilistic models for global segmentation of pathological oct scans. In *Medical Image Computing and Computer Assisted Intervention- MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part I 20*, pages 177–184. Springer, 2017.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ”why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16982.

Mireia Ribera and Agata Lapedriza. Can we do better explanations? a proposal of user-centered explainable ai. In *IUI workshops*, volume 2327, page 38, 2019.

Neal J Roese. Counterfactual thinking. *Psychological bulletin*, 121(1):133, 1997.

Joan Sol Roo and Martin Hachet. One reality: Augmenting how the physical world is experienced by combining multiple mixed reality modalities. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pages 787–795, 2017.

Amirmohammad Rooshenas and Daniel Lowd. Learning sum-product networks with direct and indirect variable interactions. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page I–710–I–718. JMLR.org, 2014.

Casey Ross and Ike Swetlitz. Ibm's watson supercomputer recommended 'unsafe and incorrect'cancer treatments, internal documents show. *Stat*, 25, 2018.

D. Rothman. *Hands-On Explainable AI (XAI) with Python: Interpret, visualize, explain, and integrate reliable AI for fair, secure, and trustworthy AI apps*. Packt Publishing, 2020. ISBN 9781800202764. URL https://books.google.co.uk/books?id=2f30DwAAQBAJ.

David-Hillel Ruben. *Explaining Explanation*. Routledge, 1990.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1–85, 2022.

Chris Russell. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 20–28, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361255. . URL https://doi.org/10.1145/3287560.3287569.

Stuart Russell, Daniel Dewey, and Max Tegmark. Research priorities for robust and beneficial artificial intelligence. *Ai Magazine*, 36(4):105–114, 2015.

Wojciech Samek and Grégoire Montavon. Explainable ai for deep networks. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 2020. URL http://www.interpretable-ml.org/ecml2020tutorial/.

Robert J Sampson, John H Laub, and Christopher Wimer. Does marriage reduce crime? a counterfactual approach to within-individual causal effects. *Criminology*, 44(3):465–508, 2006.

Kyarash Shahriari and Mana Shahriari. Ieee standard review—ethically aligned design: A vision for prioritizing human wellbeing with artificial intelligence and autonomous systems. In *2017 IEEE Canada International Humanitarian Technology Conference (IHTC)*, pages 197–201. IEEE, 2017.

L. S Shapley. *A VALUE FOR N-PERSON GAMES*. Defense Technical Information Center, 1952.

Yujia Shen, Arthur Choi, and Adnan Darwiche. Tractable operations for arithmetic circuits of probabilistic models. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Yujia Shen, Arthur Choi, and Adnan Darwiche. Conditional psdds: Modeling and learning with modular knowledge. In *AAAI*, 2018.

Will Shenton, Barry T Hart, and Terence U Chan. A bayesian network approach to support environmental flow restoration decisions in the yarra river, australia. *Stochastic Environmental Research and Risk Assessment*, 28(1):57–65, 2014.

TB Sheridan. Trustworthiness of command and control systems. In *Analysis, Design and Evaluation of Man–Machine Systems 1988*, pages 427–431. Elsevier, 1989.

Thomas B Sheridan and Automation Telerobotics. *Human supervisory control*. Cambridge, MA: MIT Press, 1992.

Weijia Shi, Andy Shih, Adnan Darwiche, and Arthur Choi. On tractable representations of binary neural networks, 2020.

Andy Shih, Arthur Choi, and Adnan Darwiche. A symbolic approach to explaining bayesian network classifiers. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5103–5111. International Joint Conferences on Artificial Intelligence Organization, 7 2018. . URL https://doi.org/10.24963/ijcai.2018/708.

Donghee Shin. The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable ai. *International Journal of Human-Computer Studies*, 146:102551, 2021.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society (AIES)*, 2020. URL https://arxiv.org/pdf/1911.02508.pdf.

Kacper Sokol and Peter A Flach. Counterfactual explanations of machine learning predictions: opportunities and challenges for ai safety. *SafeAI@ AAAI*, 2019.

Frank Sottile. *Real solutions to equations from geometry*, volume 57. American Mathematical Soc., 2011.

Karl Stelzner, Robert Peharz, and Kristian Kersting. Faster attend-infer-repeat with tractable probabilistic models. In *International Conference on Machine Learning*, pages 5966–5975. PMLR, 2019.

B Stewart-Koster, SE Bunn, SJ Mackay, NL Poff, Robert J Naiman, and Philip Spencer Lake. The use of bayesian networks to guide investments in flow and catchment restoration for impaired river ecosystems. *Freshwater Biology*, 55(1):243–260, 2010.

Eliza Strickland. Ibm watson, heal thyself: How ibm overpromised and underdelivered on ai health care. *IEEE Spectrum*, 56(4):24–31, 2019.

Harini Suresh, Steven R Gomez, Kevin K Nam, and Arvind Satyanarayan. Beyond expertise and roles: A framework to characterize the stakeholders of interpretable machine learning and their needs. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2021.

Balasaravanan Thoravi Kumaravel, Cuong Nguyen, Stephen DiVerdi, and Bjoern Hartmann. Transceivr: Bridging asymmetrical communication between vr users and external collaborators. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 182–195, 2020.

Santtu Tikka, Antti Hyttinen, and Juha Karvanen. Identifying causal effects via context-specific independence relations. *Advances in neural information processing systems*, 32, 2019.

Gabriele Tolomei, Fabrizio Silvestri, Andrew Haines, and Mounia Lalmas. Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 465–474, 2017.

Laura Uusitalo. Advantages and challenges of bayesian networks in environmental modelling. *Ecological modelling*, 203(3-4):312–318, 2007.

Guy Van den Broeck, Anton Lykov, Maximilian Schleich, and Dan Suciu. On the tractability of SHAP explanations. *CoRR*, abs/2009.08634, 2020. URL https://arxiv.org/abs/2009.08634.

Tyler J VanderWeele. Invited commentary: counterfactuals in social epidemiology—thinking outside of "the box". *American Journal of Epidemiology*, 189(3):175–178, 2020.

Jennifer Wortman Vaughan and Hanna Wallach. A human-centered agenda for intelligible machine learning. *Machines We Trust: Getting Along with Artificial Intelligence*, 2020.

Warren J von Eschenbach. Transparency and the black box problem: Why we do not trust ai. *Philosophy & Technology*, 34(4):1607–1622, 2021.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard journal of law & technology*, 31:841–887, 04 2018.

Clifford H Wagner. Simpson's paradox in real life. *The American Statistician*, 36(1):46–48, 1982.

Jinghua Wang and Gang Wang. Hierarchical spatial sum–product networks for action recognition in still images. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(1):90–100, 2016.

Lu Wang, Greg A Jamieson, and Justin G Hollands. Trust and reliance on an automated combat identification system. *Human factors*, 51(3):281–291, 2009.

Xinru Wang and Ming Yin. Are explanations helpful? a comparative study of the effects of explanations in ai-assisted decision-making. In *26th International Conference on Intelligent User Interfaces*, pages 318–328, 2021.

Adrian Weller. Transparency: Motivations and challenges. In Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller, editors, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700 of *Lecture Notes in Computer Science*, pages 23–40. Springer, 2019. . URL https://doi.org/10.1007/978-3-030-28954-6_2.

James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. The what-if tool: Interactive probing of machine learning models. *IEEE transactions on visualization and computer graphics*, 26(1):56–65, 2019.

Rebecca Wexler. When a computer program keeps you in jail. *New York Times*, 2017.

Bryan Wilder, Eric Horvitz, and Ece Kamar. Learning to complement humans. *arXiv preprint arXiv:2005.00582*, 2020.

Jacob O Wobbrock and Matthew Kay. Nonparametric statistics in human–computer interaction. *Modern statistical methods for HCI*, pages 135–170, 2016.

Robert F Woolson. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, pages 1–3, 2007.

Julia L Wright, Jessie YC Chen, and Shan G Lakhmani. Agent transparency and reliability in human–robot interaction: the influence on user confidence and perceived reliability. *IEEE Transactions on Human-Machine Systems*, 50(3):254–263, 2019.

Mingli Wu, Yafei Huang, and Jianyong Duan. Investigations on classification methods for loan application based on machine learning. In *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*, pages 1–6, 2019. .

Yaqi Xie, Ziwei Xu, Mohan S Kankanhalli, Kuldeep S Meel, and Harold Soh. Embedding symbolic knowledge into deep networks. *Advances in neural information processing systems*, 32, 2019.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. A semantic loss function for deep learning with symbolic knowledge. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 5502–5511, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018.

X Jessie Yang, Vaibhav V Unhelkar, Kevin Li, and Julie A Shah. Evaluating effects of user experience and system transparency on trust in automation. In *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction*, pages 408–416, 2017.

Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems*, 32, 2019.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness constraints: Mechanisms for fair classification. In *AISTATS*, 2015.

Matej Zečević, Devendra Dhami, Athresh Karanam, Sriraam Natarajan, and Kristian Kersting. Interventional sum-product networks: Causal inference with tractable probabilistic models. *Advances in Neural Information Processing Systems*, 34:15019–15031, 2021a.

Matej Zečević, Devendra Singh Dhami, and Kristian Kersting. On the tractability of neural causal inference. *arXiv preprint arXiv:2110.12052*, 2021b.

Richard Zemel, Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork. Learning fair representations. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, 2013.

Yunfeng Zhang, Q Vera Liao, and Rachel KE Bellamy. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 295–305, 2020.

Han Zhao, Mazen Melibari, and Pascal Poupart. On the relationship between sum-product networks and bayesian networks. In *ICML'15*, 2015.

Han Zhao, Pascal Poupart, and Geoffrey J Gordon. A unified approach for learning the parameters of sum-product networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 433–441. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/6c9882bbac1c7093bd25041881277658-Paper.pdf.

Kaiyu Zheng and Andrzej Pronobis. From pixels to buildings: End-to-end probabilistic deep networks for large-scale semantic mapping. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3511–3518. IEEE, 2019.

Kaiyu Zheng, Andrzej Pronobis, and Rajesh P. N. Rao. Learning graph-structured sum-product networks for probabilistic semantic maps. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Jianlong Zhou, Amir H Gandomi, Fang Chen, and Andreas Holzinger. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 10(5):593, 2021.

Brian J Zikmund-Fisher, Dylan M Smith, Peter A Ubel, and Angela Fagerlin. Validation of the subjective numeracy scale: effects of low numeracy on comprehension of risk communications and utility elicitations. *Medical Decision Making*, 27(5):663–671, 2007.

Shoshana Zuboff. *In the age of the smart machine: The future of work and power*. Basic Books, Inc., 1988.