Bayesian Networks for Clinical Risk Prediction

Oisín Nolan

Master of Science

Artificial Intelligence

School of Informatics

University of Edinburgh

2022

Abstract

Risk prediction is essential in clinical practice to optimise patient outcomes. Automated risk prediction models using machine learning are becoming more popular, but the clinical domain imposes specific modelling constraints which must be satisfied in order to make safe, reliable predictions. Two challenges of particular importance are *interpretability* and *the ability to handle missing data*. Bayesian network models have been proposed as a good candidate for clinical risk prediction due to their inherent interpretability, and inference algorithms that account for missing data. This project evaluates these abilities during learning and inference in Bayesian networks. An initial set of experiments focuses on *structure learning*, in which the network structure is inferred from data. Further experiments evaluate inference in Bayesian networks, and compare their predictive performance to that of logistic regression, currently a popular choice for clinical risk prediction.

The structure learning experiments reveal that incorporating prior expert knowledge into the learning procedure results in more reliable, interpretable, network structures, and can increase predictive accuracy. Experiments in inference show that while Bayesian networks can handle missing data just as well as a logistic regression model using MICE imputation, the typical predictive performance is slightly lower than that of logistic regression (AUROC for LR: 0.77, versus for BN: 0.76). Another experiment uses a fitted Bayesian model to perform causal inference, estimating the causal effect of being vaccinated on in-hospital mortality, finding that vaccinated patients are $\sim 4.5\%$ less likely to die in hospital if they are vaccinated. Finally, the project proposes a workflow for producing causal Bayesian network models, guided by the experimental results discussed until that point, and uses this workflow to create a finalised clinical risk prediction model.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Oisín Nolan*)

Acknowledgements

Firstly, I would like to thank my supervisors Dr. Sohan Seth and Prof. Ewen Harrison for the guidance and support through the project. Our meetings were always lively, full of interesting discussion and new ideas. I would also like to thank the members of the SHaPE research group, whose exciting work was a source of motivation throughout my project, and who provided thoughtful feedback on my work when I presented it.

I would like to thank my parents Seán and Nuala, and my sister Molly for their constant support and cheerfulness throughout the year.

And finally, I would like to thank my friends, whose company and humour made for an unforgettable year in Edinburgh.

Table of Contents

Chapter 1

Introduction

Risk prediction is an important challenge in healthcare, and can be used to gain insight on patient risk factors and assist in deciding on treatments to optimise patient outcomes [52]. Applying machine learning in healthcare brings interesting challenges which constrain the kinds of model that can be used. One such challenge is that the model must be *interpretable* – that humans must be able to understand the reasoning process that led the model to make a given decision, or prediction [21]. Interpretability is considered a priority by clinicians [1], and is required of machine learning systems under the European Union's *right to an explanation* provision [62, 21]. Another important challenge associated with healthcare is that of *missing data* [4], where values for certain samples in the dataset haven't been observed, and are unknown. This is a common problem in healthcare data [15], and can be a source of bias in models of the data unless handled appropriately [11]. Bayesian networks (BNs) have been presented as a promising tool for clinical risk prediction due to their interpretable modelling and inference, and intrinsic handling of missing data [7]. The potential for BNs to solve these two key issues in clinical risk prediction is the primary motivator for this project, and will be explored via the following research questions:

(i) How well can BN *structure learning algorithms* learn models of the domain from observational data? What kinds of structures are produced, and how are they affected by the incorporation of prior expert knowledge about the domain?

(ii) How do BNs compare to logistic regression in terms of predictive performance and interpretability? And how well do these models handle missing values in the data during inference?

The project has been structured according to these research questions. An initial

Background chapter provides information on the theory surrounding BN modelling and logistic regression, which is essential in understanding the experiments that follow, and in evaluating the models' interpretability from a theoretical perspective. The *Methodology* and *Results & Discussion* chapters discuss a set of experiments that were carried out to evaluate the models empirically. Both of these chapters are split into the same two primary sections: *Structure Learning*, exploring research question (i), and *Inference*, exploring research question (ii). The final chapter, *Conclusion*, provides reflections on the research questions informed by experimental results, discusses limitations of the project, and proposes some directions for future work on this topic. The dataset used in this project consists of records for patients admitted to hospital with COVID-19, containing information on demographics, observed symptoms, outcomes, and more. Details about the dataset are provided in Section 3.1.

The main contributions of this project are the following: the proposal of a set of desiderata for structure learning algorithms and corresponding metrics to evaluate the degree to which they are satisfied; experimental results characterising three popular structure learning algorithms, which can serve to guide their application in future; experimental results comparing predictive performance of BNs and logistic regression, highlighting the strengths and weaknesses of each, and indicating which model may be a better choice for various use-cases; estimates for the causal effect of being vaccinated against COVID-19 on various health outcomes; the proposal of a workflow for producing accurate, interpretable BN models.

1.1 Related Work

Risk prediction models have been developed to help manage many health conditions, for example, heart failure [52] and chronic kidney disease [71]. More recently, the emergence of COVID-19 has spurred the development of many more risk prediction models, such as the 4C Mortality Score [37], which this project builds upon. It is common in these works to use some method for feature selection, for example, LASSO [23, 26], to identify salient explanatory variables for COVID-19 outcomes. Machine learning models have been employed extensively for this task due to their ability to model complex functions, achieving high predictive accuracy [2]. A key factor determining the suitability of machine learning models for this task is their interpretability, as is reflected by the choice of model in many of the approaches to this task. In particular, interpretable models such as Cox proportional hazards regression [31, 34], decision-treebased models (XGBoost) [26, 34, 3], and logistic regression [37, 80, 3] have commonly been used. BNs also fit the model requirements for clinical risk prediction very well, but as of yet have been used less commonly.

More generally, however, BNs have received some attention in recent years, with multiple surveys on structure learning [35, 58, 79] being published as new methods emerge. Some work empirical work evaluating and comparing structure learning algorithms has also been carried out [68, 60], typically using simulated data to determine the structural accuracy of a given method. Note, however, that evaluation using simulated data has been subject to some criticism, claiming that these benchmarks are "*easy to game*", leading to overly optimistic results [54]. This project thus focuses on evaluation metrics that make less strong assumptions about the data generation process. BNs have also found application in a variety of domains, such as environmental modelling [74], and risk assessment of various kinds [43, 70], typically chosen due to their interpretability, modelling of indirect associations, and ability to predict multiple variables with a single model. Many applications of BNs in healthcare also exist, although they typically focus on a narrow range of medical conditions, including cardiac conditions and cancers [46]. Despite the many papers published on this topic, BNs have rarely been deployed in the real world, which has been attributed to a lack of development processes [41]. This project addresses this problem in Section 4.3, where a simple workflow for developing accurate and interpretable BNs is proposed.

Some work modelling COVID-19 data with BNs has also been published recently, with focus on quantifying vaccination risk [42], contact tracing [22], and feature selection [76]. A couple of papers also use BNs to predict the probability of health outcomes in COVID-19 patients. One work manually specifies the network structure based on expert knowledge from the COVID-19 literature [66]. It finds that the BN outperforms a Support Vector Machine model in terms of classification accuracy, but uses very small training and test sets, containing 250 and 50 samples, respectively. Furthermore, the accuracy contribution per is weighted according to the label, using unexplained weights. These results should thus be met with some degree of skepticism, and motivate further evaluation with a larger dataset. Another paper [76] learns the network structure with the hill-climb algorithm [57], and finds that the model achieves good predictive performance, however it doesn't compare the BN to any baselines or other models.

Chapter 2

Background

2.1 Bayesian Networks

2.1.1 Model Definition

BNs model a joint probability distribution as a product of conditional probability distributions (CPDs), leveraging a set of conditional independence assumptions to make learning and inference computationally feasible [10]. This product of CPDs can be derived by applying the chain rule to factorise the joint distribution, and then applying simplifications of the form $p(x \mid y,z) = p(x \mid y)$ according to conditional independence assumptions $x \perp\!\!\!\perp z \mid y$. This results in a simplified factorisation of the joint, as shown in equation 2.1, where $\mathrm{pa}(x_i)$ is the set of variables in the conditioning set for x_i, known as its *parents*.

$$p(x_1, x_2, ..., x_d) = \prod_{i=1}^{d} p(x_i \mid \mathrm{pa}(x_i)) \tag{2.1}$$

This notion of parent variables may be used to visualise the factorisation as a *directed acyclic graph* (DAG) in which each variable is a node, and directed edges point to each variable from its parents. Figure 2.1 shows a simple example of this DAG visualisation.

A graphical criterion called **d-separation** [49] can be used to read conditional independencies directly from a given DAG. Two sets of nodes $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated by some set of observed nodes $\boldsymbol{S}$ if all trails between any nodes in $\boldsymbol{X}$ and $\boldsymbol{Y}$ are *blocked* by $\boldsymbol{S}$. Whether a trail is active or blocked can be deduced by examining the various kinds of edge configuration that compose the trail [38]:

- **Causal trail**, $X \to Z \to Y$: blocked iff $Z \in \boldsymbol{S}$

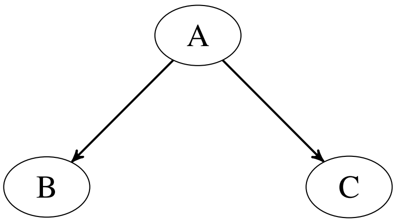


Figure 2.1: A DAG visualising the factorisation $p(A,B,C) = p(A)p(B \mid A)p(C \mid A)$.

- **Evidential trail**, $X \leftarrow Z \leftarrow Y$: blocked iff $Z \in \boldsymbol{S}$
- **Common cause**, $X \leftarrow Z \rightarrow Y$: blocked iff $Z \in \boldsymbol{S}$
- **Common effect**, $X \rightarrow Z \leftarrow Y$: blocked iff $Z \notin \boldsymbol{S}$, and none of Z's descendents $\in \boldsymbol{S}$.

If $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated by $\boldsymbol{S}$, then $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{S}$. This provides a means of defining the set of independencies asserted by any DAG $\mathcal{G}$, denoted $I(\mathcal{G})$ [38]:

$$I(\mathcal{G}) = \{(\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{S}) : \text{d-sep}_{\mathcal{G}}(\boldsymbol{X};\boldsymbol{Y} \mid \boldsymbol{S})\} \tag{2.2}$$

Because conditional independence is a symmetric relation between variables, i.e. $X \perp\!\!\!\perp Y \mid Z \iff Y \perp\!\!\!\perp X \mid Z$, it is possible that the same set of independence assumptions can correspond with multiple DAGs. The set of independencies specified by a DAG $\mathcal{G}$, may thus be equal to those of another DAG $\mathcal{G}'$, yielding $I(\mathcal{G}) = I(\mathcal{G}')$, in which case $\mathcal{G}$ and $\mathcal{G}'$ are said to be **I-equivalent** [38]. This equivalence relation partitions the set of all DAGs into **Markov equivalence classes** (MECs) – sets of DAGs which share the same independence assumptions [10]. A MEC can be conveniently represented by another graphical structure called a *completed partially directed acyclic graph* (CPDAG) [5]. The edge semantics for CPDAGs are as follows: an undirected edge $X - Y$ indicates that there is some DAG containing edge $X \rightarrow Y$ and another containing $X \leftarrow Y$ in the equivalence class, a directed edge $X \rightarrow Y$ means that every DAG in the equivalence class contains edge $X \rightarrow Y$, and the absence of an edge means that it is absent in every DAG in the equivalence class [30]. It is also possible that $I(\mathcal{G}) \subseteq I(\mathcal{G}')$, in which case $\mathcal{G}$ is said to be an *independence map*, or **I-map** of $\mathcal{G}'$.

In general, there are no restrictions on how the factors $p(x_i \mid \text{pa}(x_i))$ are defined, beyond them being valid conditional probability distributions. In the case of this project, in order to maintain model interpretability, discrete conditional probability tables were used, as described in Section 2.1.4.

2.1.2 Causal Interpretation

BNs, as described thus far, make claims about statistical dependencies between variables. However, a natural interpretation of such DAGs is *causal*: one in which each directed edge (X,Y) indicates that X has a causal effect on Y. The DAG can then be seen as a theory of the data generation process underlying the observed data. Reichenbach, in his *common cause principle* [53], made a connection between statistical and causal dependence, stating that if two variables X and Y are observed to be statistically dependent, then there must exist some variable Z that causally influences both X and Y, and that explains their dependence insofar as X and Y would become independent conditioned on Z [59]. Note that Z may be either of X or Y in this case, accounting for the simpler $X \rightarrow Y$ and $X \leftarrow Y$ cases.

In order to draw conclusions about causal relationships from observational data, a number of assumptions are typically necessary [30, 17]:

- **Causal sufficiency.** Reichenbach's common cause principle implies that if X and Y are statistically independent conditional on some $Z \neq X$ or Y, then neither causes the other, and we could be sure that no edge $X \rightarrow Y$ or $X \leftarrow Y$ should appear in the DAG. However, in order to be sure about such a conclusion, one would need to assume that they have accounted for all the possible confounders of X and Y, i.e. that there are no hidden confounding Z variables [17]. This assumption is known as causal sufficiency [69].

- **Causal representation.** This assumption states that there exists some DAG that is a causal representation of the underlying system that has generated the observed data [17]. Note the *acyclicity* constraint imposed by the fact the system is modelled as a DAG, preventing feedback loops [30].

- **Causal Markov condition.** The causal DAG, via d-separation, specifies the same set of probabilistic independencies as the underlying system [17].

- **Causal faithfulness.** This assumption states the inverse of the causal Markov condition, namely that any conditional independence property that holds of the underlying system is specified by the causal DAG [30].

Taken together, these assumptions state that some causal DAG must exist, that there must be a one-to-one correspondence between those independencies given by d-separation in the DAG and the independencies that hold of the underlying system, and that there are no hidden confounders.

2.1.3 Structure Learning

If the dependence structure of the variables is known *a priori* by a domain expert, the BN structure can be manually specified. However, this may be infeasible for a number of reasons; for example, the number of possible DAGs increases super-exponentially with the number of variables [55]. An alternative is to infer automatically the dependence structure of the BN DAG from data. This task is known as *structure learning* [58, 35], or *causal discovery* [25]. An idealistic goal for structure learning would be to recover the DAG corresponding to the true underlying process that generated the observed data, $\mathcal{G}^*$. However, the true distribution of the observed data, P^*, shares the same independencies as any DAG that in the same MEC as $\mathcal{G}^*$ [38]. Hence, even in the limit of data samples, the best structure learning algorithm will only be able to find the equivalence class of the true DAG, $\mathcal{G}^*$. Under the assumptions stated in Section 2.1.2, this equivalence class should contain the true causal DAG [17]. Once an equivalence class has been identified by a structure learning algorithm, domain expertise or further assumptions could be used to narrow the set down to a final candidate for $\mathcal{G}^*$. Additionally, some methods in inference which consider an equivalence class of DAGs, rather than a single DAG, have been developed, for example, *intervention-calculus when the DAG is absent* (IDA) [45]. While the primary usage of structure learning in this project is to produce models of a joint distribution which can be used to make predictions, a secondary use-case is that of *knowledge discovery* [38]. Structure learning algorithms can be used to surface relationships between variables in the data, and in particular, distinguish between *direct* dependencies, $X \rightarrow Y$, and *indirect* dependencies, $X \rightarrow M \rightarrow Y$, which could not be distinguished by simple pairwise correlation tests [38].

Many structure learning algorithms have been developed over the past couple of decades. They are typically classified as either *constraint-based*, *score-based*, or *hybrid*. Constraint-based algorithms employ conditional independence tests to determine which independencies hold in the data. The identified independencies are then used to construct an equivalence class of DAGs [35]. Score-based algorithms approach the problem from a more traditional machine learning perspective, by optimising an objective function that evaluates the suitability of a given DAG or equivalence class [35]. Hybrid algorithms aim to combine techniques from both constraint-based and score-based approaches. In this project, we have considered one constraint-based algorithm, *PC* [69], and two score-based algorithms, *greedy equivalence search* [14] and *NOTEARS* [81]. The following subsections will describe these algorithms in detail.

2.1.3.1 PC Algorithm

The PC algorithm [69] is a popular and well-established structure learning algorithm, named after its creators Peter Spirtes and Clark Glymour. It is a constraint-based algorithm, and employs conditional independence tests and logical rules to identify the causal DAG up to its MEC, represented as a CPDAG, under the causal assumptions discussed in Section 2.1.2. The PC algorithm consists of two steps: first, determine the undirected skeleton for the CPDAG is produced, and then orient as many edges as possible. The first step makes use of the following theorem [69] in determining whether edges should be present or absent in the output CPDAG:

Theorem 1 *for all vertices X, Y of a DAG $\mathcal{G}$, X and Y are adjacent if and only if X and Y are dependent conditioned on every set of vertices of $\mathcal{G}$ that does not include X or Y.*

Taking the contrapositive of Theorem 1, we see that if X and Y can be made independent by conditioning on some subset $\boldsymbol{S}$ of the vertices, then X and Y are not adjacent.

The algorithm begins with a fully-connected undirected graph, and for each pair of nodes X,Y, iterates through each subset $\boldsymbol{S} \subseteq \text{Adj}(X) \setminus Y$, where $\text{Adj}(X)$ gives the set of nodes adjacent to X. For each of these subsets $\boldsymbol{S}$ we run a conditional independence test to determine if $X \perp\!\!\!\perp Y \mid \boldsymbol{S}$ holds. If any of these tests are positive, at a specified confidence level α, then the edge between X and Y is removed [30]. If conditioning on $\boldsymbol{S}$ is found to make X and Y independent, then $\boldsymbol{S}$ is recorded in $\textbf{Sepset}(\{X,Y\})$ [69]. In order to test whether $X \perp\!\!\!\perp Y \mid Z$ holds, one may compute the conditional mutual information $I(X;Y \mid Z)$ as defined in Equation 2.3, where $D_{KL}(P||Q)$ gives the *Kullback-Leibler divergence* between two probability distributions P and Q. The KL-divergence can be seen as a measure of difference between two probability distributions, and thus the conditional mutual information measures the expected difference between $P_{X,Y|Z}$ and $P_{X|Z} \times P_{X|Z}$, which is equal to zero if and only if $X \perp\!\!\!\perp Y \mid Z$.

$$I(X;Y \mid Z) = \mathbb{E}_Z[D_{KL}(P_{X,Y|Z})||P_{X|Z} \times P_{Y|Z})] \tag{2.3}$$

Once the above procedure has run all the necessary conditional independence tests, we are left with the undirected skeleton of the output CPDAG. Directionality may be inferred in a number of edges using a number of rules [69]. First, for any three nodes X,Y,Z in which X,Y are adjacent and Y,Z are adjacent but X,Z are not adjacent, we can orient the structure $X - Y - Z$ as $X \rightarrow Y \leftarrow Z$ if Y is not contained in $\mathbf{Sepset}(\{X, Z\})$. The reason we may do this is because X and Z would independent conditional on Y if $X - Y - Z$ were to have any structure other than $X \rightarrow Y \leftarrow Z$, known as **common effect** or **v-structure**, and described in Section 2.1.1. Once the initial common effects have been identified, the following two rules are repeatedly applied until no more edges can be directed:

- If a structure $X \rightarrow Y - Z$ exists, and X, Z are not adjacent, then orient $Y - Z$ as $Y \rightarrow Z$.
- If there is a directed path $X \rightarrow ... \rightarrow Y$ and an undirected edge $X - Y$, then orient $X - Y$ as $X \rightarrow Y$.

Having applied these rules, what remains is a CPDAG defining the equivalence class containing $\mathcal{G}^*$, the true underlying DAG, assuming that the conditional independence tests were correct. This CPDAG is then returned by the algorithm.

2.1.3.2 Greedy Equivalence Search

Greedy equivalence search (GES) [14] is a score-based algorithm that can identify the true DAG up to its MEC in the limit of the data, assuming faithfulness, sufficiency, and acyclicity [30]. This algorithm uses the Bayesian information criterion (BIC) as a scoring function, to choose between candidate CPDAG structures. The BIC is defined in Equation 2.4 [14], where $\boldsymbol{D}$ is the observed data set, $\hat{\boldsymbol{\theta}}$ are the maximum-likelihood values for the network parameters, d is the number of parameters in the network, and m is the number of observed samples in $\boldsymbol{D}$.

$$S_{BIC}(\mathcal{G}, \boldsymbol{D}) = \log p(\boldsymbol{D} \mid \hat{\boldsymbol{\theta}}, \mathcal{G}) - \frac{d}{2} \log m \tag{2.4}$$

The BIC thus aims to balance the likelihood that a model assigns to the observed data with the number of parameters in the model, such that if two models M and M' have equal likelihood but M has fewer parameters, then $S_{BIC}(M) > S_{BIC}(M')$. The BIC is a *locally consistent scoring function* [14], which means that in the limit of m, if adding an edge to G removes an independence assertion $X \perp\!\!\!\perp Y \mid Z$, then the score will increase if $X \perp\!\!\!\perp_{p^*} Y \mid Z$, and it will decrease if $X \not\perp\!\!\!\perp_{p^*} Y \mid Z$, where $\perp\!\!\!\perp_{p^*}$ denotes that the independence holds in the underlying generative distribution p^*, and $\not\perp\!\!\!\perp_{p^*}$ that it does not hold in p^*.

GES consists of two main phases. In the first phase, we begin with an empty graph and greedily add the edge which maximises the BIC until we reach a local maximum. This can be thought of as a discrete traversal of CPDAG space, in which adding a new edge reaches a new CPDAG state. It has been shown [14] that the equivalence class reached at this local maximum must be an I-map for p^*. This is because the BIC would have decreased if we removed any independencies in p^*, in accordance with BIC's local consistency. In the second phase, edges are greedily removed until a local maximum has been reached, at which point the current CPDAG is I-equivalent to the true underlying DAG generating p^*. This I-equivalence follows from a proof of Meek's conjecture [14], which shows that for any pair of DAGs $\mathcal{G}$ and $\mathcal{H}$ such that $\mathcal{H}$ is an I-map of $\mathcal{G}$, there exists a sequence of edge removals and reversals that can be applied to $\mathcal{H}$ such that $\mathcal{G} = \mathcal{H}$.

GES has been shown to be optimal in the limit of data samples, however, because this may not be feasible in real-world scenarios, it was found that repeating these two phases iteratively, along with an additional *turning phase* in which edges may be re-oriented, can yield better results [29].

2.1.3.3 NOTEARS

The NOTEARS [81] algorithm takes an alternative approach to score-based structure learning, opting to use continuous optimisation on the adjacency weight matrix of the DAG rather than a discrete search-based optimisation, as in GES. This is achieved by formulating the network as a linear structural equation model (SEM), in which each variable X_i is a linear function of the other variables with some added noise, i.e. $X_i = \boldsymbol{w}_i^\top \boldsymbol{X} + \boldsymbol{z}_i$, where $\boldsymbol{w}_i$ is a column from the weights matrix W, and $\boldsymbol{z}_i$ is random noise. In order to enforce acyclicity in the model structure, the function $h(W)$ was introduced. It is defined in Equation 2.5, where $\mathrm{tr}(.)$ gives the trace of a matrix, and d is the number of variables.

$$h(W) = \mathrm{tr}(e^{W \circ W}) - d \tag{2.5}$$

It has been shown [81] that $h(W) = 0 \iff W$ is a DAG. This means that $h(W) = 0$ can be used as an equality constraint while minimising the loss of the SEM to ensure that it remains acyclic, as desired. Thus, the problem of structure learning becomes the following equality-constrained program:

$$\begin{aligned} &\min_{W \in \mathbb{R}^{d \times d}} F(W) \\ &\text{subject to:} \quad h(W) = 0 \end{aligned} \tag{2.6}$$

Where $F(W)$ is a score function, and in the case of a linear SEM, is defined as in Equation 2.7, using least squares loss with an ℓ_1 regularisation term weighted by λ, where n is the number of samples.

$$F(W) = \frac{1}{2n}||\boldsymbol{X} - \boldsymbol{X}W||^2 + \lambda||W||_1 \tag{2.7}$$

Note that, in general, X_i can be modelled as a generalised linear model of the other variables, and need not necessarily be a linear SEM [81]. NOTEARS has recently been criticised as a *causal discovery* method [32, 63], and so in this project is only used to learn associational models, with PC being the preferred choice for causal discovery.

2.1.4 Parameter Estimation

Maximum likelihood estimation (MLE) is a method for fitting model parameters in which the parameters that maximise the likelihood of the observed data under this model are chosen. This is summarised by Equation 2.8, in which $\hat{\boldsymbol{\theta}}$ are the optimal parameters, D is the observed data, and L is the likelihood function [38].

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmax}} L(\boldsymbol{\theta}; D) \tag{2.8}$$

In the case of BNs, the likelihood of some parameters $\boldsymbol{\theta}$ can be formulated as in Equation 2.9, where $X_i[m]$ denotes the value of X_i in the m^{th} sample, and $\mathrm{pa}_{X_i}[m]$ denotes the values of X_i's parameters in the m^{th} sample [38].

$$L(\boldsymbol{\theta}; D) = \prod_m \prod_i P(X_i[m] \mid \mathrm{pa}_{X_i}[m]; \boldsymbol{\theta}) \tag{2.9}$$

The two products in Equation 2.9 can be swapped such that the likelihood becomes a product of likelihoods for each CPD $P(X_i \mid \mathrm{pa}_{X_i})$. In this sense, the global likelihood function can be decomposed into *local* likelihood functions, the parameters of which may be optimised independently in order to reach a global optimum [38]. In the case of discrete BNs, as used in this project, each of the CPDs is a multinomial distribution. Fortunately, there is a simple method for calculating maximum likelihood parameters for multinomial distributions, given in Equation 2.10, where $\#D[\mathrm{pa}_{X_i}, X_i]$ is the number of observed occurrences of a given value of X_i along with a given set of values for X_i's parents, and $\#D[\mathrm{pa}_{X_i}]$ is the number of occurrences of those values of X_i's parents.

$$P(X_i \mid \mathrm{pa}_{X_i}; \hat{\boldsymbol{\theta}}) = \hat{\boldsymbol{\theta}}_{X_i|\mathrm{pa}_{X_i}} = \frac{\#D[\mathrm{pa}_{X_i}, X_i]}{\#D[\mathrm{pa}_{X_i}]} \tag{2.10}$$

These parameters are easily interpretable to non-experts, assigning a specific probability to each possible event, with a direct translation to natural language. For example: " The probability of *admittance to ICU* given that the patient is *above 80 years of age* is *0.14* ". These CPDs can be visualised as in Appendix F.

2.1.5 Inference

Inference in BNs, as in many statistical and machine learning models, aims to estimate the distribution $p(Y \mid E = e)$, where Y is some outcome variable of interest and e is some instance of a evidence variables E which should be used to predict Y. Many algorithms for performing inference in BNs exist. In this section, we will describe two such algorithms, one that performs *exact inference* in which we calculate the exact values for $P(Y \mid E = e)$, and one that performs *approximate inference*, in which we approximate the probabilities. We also discuss causal inference, in which we can infer estimates of causal effects of X on Y, under the assumptions discussed in Section 2.1.2.

2.1.5.1 Variable Elimination

The variable elimination algorithm [38] is used to compute marginal distributions from BNs, exploiting the network factorisation in order to make the computation more efficient. The efficiency comes from formulating a sum over all the variables in the joint distribution to a product of local sums over the variables relevant smaller products of relevant factors. See Equation 2.11 for a toy example of this sum of products to product of sums trick [38]:

$$\begin{aligned} P(D) &= \sum_C \sum_B \sum_A P(A)P(B \mid A)P(C \mid B)P(D \mid C) \\ &= \sum_C P(D \mid C) \sum_B P(C \mid B) \sum_A P(A)P(B \mid A) \end{aligned} \tag{2.11}$$

This trick helps in reducing the exponential blowup caused by summing over all combinations of values for each variable in the distribution, but is still exponential in the local factor sums. Marginal inference is the key operation necessary to compute the conditional $P(Y \mid E = e)$ because it is defined as $\frac{P(Y,e)}{P(e)}$, two marginal distributions. Variable elimination is employed in this way to compute both $P(Y,e)$ and $P(e)$, and hence $P(Y \mid E = e)$. This method thus inherently handles missing data by summing out variables with missing values, i.e. variables not in the evidence set E.

2.1.5.2 Likelihood Weighting

The likelihood weighting algorithm [38] uses importance-sampling to approximate $P(Y \mid E = e)$, rather than compute it exactly as in variable elimination. The benefit of this approach is that it scales better with the density of the graph. This algorithm works by *forward sampling* from the BN, and calculating a weighting for the sample, w, as a product of the probabilities of the weighted samples given the observed evidence e. For example, if we have a network $P(A)P(B \mid A)$, and evidence that $B = b_0$, we can sample $a_0 \sim P(A)$ and then calculate our sample weight as the probability that we could have observed $B = b_0$ given a_0, $w = P(B = b_0 \mid A = a_0)$. We continue forward sampling in this way, updating the weight for each factor with observed evidence. This process is repeated M times, generating M samples with corresponding weights. $P(Y \mid E = e)$ can then be computed as in Equation 2.12, where $w[m]$ is the weight assigned to the m^{th} sample, $\mathbf{1}(.)$ is the indicator function, and y^* is the value of Y whose conditional probability we want to compute.

$$\hat{P}_D(y^* \mid e) = \frac{\sum_m w[m]\mathbf{1}\{y[m] = y^*\}}{\sum_m w[m]} \tag{2.12}$$

2.1.5.3 Causal Inference

Under the causal assumptions defined in Section 2.1.2, BNs can be used to simulate interventions on variables of interest. Interventional queries of this form involve setting some variable X to a specific value x_0 to estimate the effect that would have on some other variable Y. For example, this could be used to estimate the causal effect of being vaccinated against COVID-19 on various health outcomes. The *do-operation* is a piece of notation introduced by Pearl [50] to denote interventions, where $p(Y \mid do(X = x_0))$ gives a probability distribution over Y given that X has been set to x_0. Given some DAG $\mathcal{D}$ for an observational distribution $p(X, Z_1, ..., Z_K, Y)$, applying $do(X = x_0)$ has the effect of removing X's parent connections, resulting in a new DAG $\mathcal{D}'$ corresponding to the interventional distribution $p(Z_1, ..., Z_K, Y \mid do(X = x_0)$. Removing X's parent connections in $\mathcal{D}$ corresponds with removing the factors $p(X \mid \text{pa}(X))$ in the factorisation specified by $\mathcal{D}$, as X's value becomes deterministic, resulting in a *truncated factorisation* [51] corresponding to $\mathcal{D}'$. Fortunately, the truncated factorisation is defined in terms of factors that can be estimated from observational data, providing a way to compute $P(Y \mid do(X = x_0)$ from observational data. A toy example [51] is given in Equation 2.13.

$$\begin{aligned} P(X,Z,Y) &= p(Z)p(X \mid Z)p(Y \mid Z,X) \quad \text{(chain rule factorisation)} \\ p(Z,Y \mid do(X = x_0)) &= p(Z)p(Y \mid Z,x_0) \quad \text{(truncated factorisation)} \\ p(Y \mid do(X = x_0)) &= \sum_Z p(Z)p(Y \mid Z,x_0) \quad \text{(desired distribution)} \end{aligned} \tag{2.13}$$

Removing the parent connections means that no information can flow from X to Y other than via causal pathways, thus isolating the causal effect. Similarly, conditioning on a set of variables that d-separates X and Y only along non-causal paths can be used to isolate the causal effect – a technique known as *backdoor adjustment* [47]. Once these interventional distributions have been calculated, they can be used to compute statistics useful for understanding the causal effects of X on Y, such as the *average treatment effect* (ATE) on Y of X being equal to x_0 as opposed to x_1 [47]:

$$\mathbb{E}[Y(x_1) - Y(x_0)] = \sum_y y \cdot P(y \mid do(X = x_1)) - \sum_y y \cdot P(y \mid do(X = x_0)) \tag{2.14}$$

2.2 Logistic Regression

Logistic regression is a kind of **generalised linear model** which can be used to model binary outcome variables. A generalised linear model can be specified via two components [19]: (i) a probability distribution for the outcome variable, Y, and (ii) an equation linking the expected value of Y to a linear combination of the dependent variables, of the form in Equation 2.15:

$$g[\mathbb{E}(Y)] = \beta_0 + \beta_1 X_1 + \ldots + \beta_m X_m \tag{2.15}$$

In the case that Y is binary, it can be modelled as a Bernoulli distribution, $Y \sim \text{Bern}(p)$, where p is the probability of success. $\mathbb{E}(Y) = p$, in this case, and so the model applies the sigmoid function, $\sigma(x) = \frac{1}{1+e^{-x}}$, to the linear combination of input variables in order to ensure that $\mathbb{E}(Y)$ is a valid probability in $[0,1]$. Hence, when written in the form of Equation 2.15, the linking function $g(.)$ becomes the *logit* function, $\log(\frac{p}{1-p})$, which can be interpreted as the *log-odds* of a success event in Y. MLE is typically used to optimise the parameters $\boldsymbol{\beta}$ by maximising the log-likelihood function given in Equation 2.16 [64], where $Y[m]$ is the m^{th} sample in our dataset, $\boldsymbol{X}[m]$ is a vector of the input variables in the m^{th} sample, and $\boldsymbol{\beta}$ is the parameter vector.

$$\ell(\boldsymbol{\beta}) = \sum_{m} Y[m] \log p[m] + (1 - Y[m]) \log(1 - p[m]) \tag{2.16}$$

$$p[m] = \sigma(\boldsymbol{X}[m]^{\top} \boldsymbol{\beta}) \tag{2.17}$$

When the input variables are categorical, they can be modelled using *dummy variables*, where each input variable X_i is split up into K new binary variables, $X_i^1, \ldots, X_i^K$, one for each of the K values that X_i can take on. Now $X_i^k = 1$ if $X_i = k$, and 0 otherwise. This enables an interpretation of the parameter β_i^k as the increase in the log-odds of observing a success in Y due to observing that $X_i = k$ [27].

2.3 Data Imputation

Missing data is an important problem in health data science which must be handled appropriately in order to minimise bias in data analyses and predictive models [27]. If there is no pattern to the missingness in the observed data, then those missing values are said to be *missing completely at random* (MCAR) [27], in which case the samples with missing values may simply be ignored, as if we had observed fewer samples in the first place. However, it is often the case that there is some pattern to the missingness, in which case the missing values are considered *missing at random* (MAR). In this case, if one were to remove samples with missing values, the apparent distribution of the observations would change, causing bias in subsequent analyses [27]. Data imputation such as *Multivariate imputation by chained equations* (MICE) have been designed to alleviate this problem. This algorithm consists of a few simple steps [8]: First fill in missing values for each variable with its mean value, known as *mean imputation*. For a particular variable X_i, set the mean imputations back to missing. Then, fit a regression model (or classification model if the data are discrete) to the observed samples for X_i, using the other variables $X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_d$ to predict the missing values for X_i. Continue this process for each variable, using the other variables to impute its missing values. This may be repeated iteratively to improve imputations.

Chapter 3

Methodology

This chapter focuses on sets of experiments that were carried out to shed light on the primary research questions of this paper, detailed in Chapter 1. The chapter begins with a description of the dataset in Section 3.1. Following this, a set of experiments described in Section 3.2 aims to provide an answer to the first primary research question, characterising and comparing the kinds of structures produced by the PC, GES, and NOTEARS algorithms, and exploring how they are affected by the incorporation of prior knowledge. Insights from these experiments should serve as a guide to the application of these algorithms in risk prediction, with particular relevance to the interpretability of the learned structure. Section 3.3 will then focus on experiments related to inference, exploring the effects of missing data on classification performance, and comparing BNs to logistic regression via a number of standard evaluation procedures. This section will also demonstrate the ability to perform causal inference with BNs, an important feature which distinguishes BNs from standard machine learning models. Causal inference is of particular significance in risk prediction, where clinicians may want to take real-world actions based on relations identified by the model. In particular, this experiment uses *interventional queries* to estimate the causal effect of being vaccinated on various health outcomes, such as death, requiring oxygen, or being admitted to ICU. The results to the experiments in Section 3.3 should answer the second primary research question, highlighting the conditions under which one might prefer to use BNs over logistic regression or similar. The results to these experiments, along with corresponding analyses, are presented in Chapter 4.

3.1 Dataset

The data used for the experiments in this project were collected as part of a the International Severe Acute Respiratory and Emerging Infections Consortium (ISARIC) World Health Organization (WHO) Clinical Characterisation Protocol UK (CCP-UK) prospective cohort study [20]. The dataset includes information on: patient demographics, such as age and sex; relevant clinical details, such as pre-existing comorbities and current symptoms; and outcomes, such as mortality or admittance to ICU [37]. Some statistics summarising the dataset are available in a paper by Docherty et al. [20] and its supplementary material. For example, it was found that 26% of patients in the study died in hospital, and some associated risk factors identified include increased age, male sex, and comorbidities such as cardiac disease [20].

This dataset was used in the development of the *4C Mortality Score* [37], which is used to predict risk of in-hospital mortality in patients admitted to hospital with COVID-19 from a small set of easily-measurable features. Features included in the final score are the following: age, sex, number of comorbidities, respiratory rate, peripheral oxygen saturation, level of consciousness, urea level, and C-reative protein [37]. Some initial pre-processing was performed on these variables in developing the 4C score [1]. For example, the numerical variables have been discretized into factors, with levels corresponding to intervals in the original variable. These have been kept simple, with two or three levels per variable, except for age, which is partitioned into five intervals. Some samples with extreme values were also removed.

Three datasets, *small* (9 variables), *medium* (30 variables), and *large* (78 variables), were created using subsets of the variables from the ISARIC WHO CCP-UK set. The small dataset contains the same variables used for the 4C score, with the same discretization. The medium set expands the variable counting the number of comorbidities into individual comorbidities. The large dataset then adds additional information on patient symptoms, such as chest pain and cough. In this sense, each dataset is a subset of the next: $small \subset medium \subset large$. Full lists of the variables included in each set are specified in Appendix D. For the medium and large datasets, numeric variables were turned into factors using a discretization algorithm that automatically determines which intervals should become levels in the resulting factors. The *Hartemink* discretization algorithm [28], as implemented in the `bnlearn` R package [61] was used to this end. A

[1] https://github.com/SurgicalInformatics/4C_mortality_score/blob/master/01_data_prep.R

completed version of the small dataset was created using multiple imputation with the `mice` R package [75], as NOTEARS and GES require fully-observed data for structure learning.

3.2 Structure Learning

A key goal of this project is to explore the feasibility of automatically learning BN structures from observational data. Accordingly, a set of desiderata for structure learning algorithms has been proposed: a good structure learning algorithm should be *scalable*, it should produce *sparse DAGs*, and it should be *reliable* in discovering the underlying causal structure, as opposed to fitting patterns of noise in the data (i.e. overfitting). A number of experiments have been devised to evaluate the degree to which the PC, GES, and NOTEARS algorithms satisfy these desiderata. The following sections describe the setup for each of those experiments. Section 3.2.3 discusses further experiments designed to explore the effects of specifying prior structural constraints on the DAGs produced by these algorithms.

Many implementations exist for popular structure learning algorithms such as PC. Here, `bnlearn`'s `pc.stable` implementation was used. PC stable [16] iterates on the original PC algorithm, making it independent of the input variable ordering, and more conservative in the edge-orientation phase. For GES, an implementation from the `pcalg` [33] R package was used. Finally, for NOTEARS, the *causalnex* python implementation [12] was used, with the `reticulate`[2] R package serving as an interface between R and python.

3.2.1 Scalability

It is important that structure learning algorithms scale well both in the number of variables (horizontally) and the number of samples (vertically). Horizontal scalability facilitates the automatic learning of large, complex DAGs, which would be very labour-intensive for domain experts to produce by hand. Vertical scalability offers statistical gains, enabling the algorithms to make use of large datasets, leading to more confident statistical estimates. Both horizontal and vertical scalability were evaluated in terms of algorithm runtime on a number of datasets of increasing sizes. For horizontal scalability, the small, medium, and large datasets were used, and for vertical scalability, subsets of

[2] `https://cran.r-project.org/web/packages/reticulate/index.html`

the small dataset with increasing numbers of samples (10^3, 10^4, 10^5) were used. See Section 4.1 for the results of this experiment.

3.2.2 Edge Statistics

In order to characterise the typical sparsity and reliability of the DAGs produced by the structure learning algorithms, we ran them on 100 subsets of size 10,000, randomly sampled from the small set, and took statistics on the edges of the produced CPDAGs. In order to measure the typical sparsity of DAGs produced by a given algorithm, we simply measure the mean number of edges present in the set of 100 CPDAGs. We propose *average edge variance* as an indicator of algorithm reliability. This statistic may be calculated as follows: first, we model edge presence as a Bernoulli variable, $E_{i,j} \sim \text{Bern}(p_{i,j})$, where $E_{i,j} = 1$ if there is the edge $X_i \rightarrow X_j$ is present in the graph, and $E_{i,j} = 0$ if that edge is not present. We can estimate $\hat{p}_{i,j}$ using MLE as in Equation 3.1 , where $\#E_{i,j}$ is the number of times that $X_i \rightarrow X_j$ was present in n CPDAGs fit on sets of data sampled from the same distribution.

$$\hat{p}_{i,j} = \frac{\#E_{i,j}}{n} \tag{3.1}$$

Once $\hat{p}_{i,j}$ has been estimated for each possible edge location, we compute its variance [13]: $\mathbb{V}(E_{i,j}) = \hat{p}_{i,j} \cdot (1 - \hat{p}_{i,j})$. This variance will be high if the presence of $E_{i,j}$ is inconsistent throughout the trials, i.e. if $\hat{p}_{i,j}$ is near 0.5, and the variance will be low if $E_{i,j}$ is more consistently either present or absent in the trials, i.e. if $\hat{p}_{i,j}$ is near 0 or 1. Finally, the average edge variance for the CPDAG is computed by taking the mean of $\mathbb{V}(E_{i,j})$ for each pair of variable indices i, j. The intuition for this metric is that a structure learning algorithm that reliably discovers a similar underlying DAG in datasets sampled from the same distribution will produce CPDAGs with low average edge variance. Conversely, an algorithm that overfits, i.e. finds spurious edges due to noise in the data, will have high average edge variance, and might thus be considered less reliable.

3.2.3 Prior Knowledge Constraints

While structure learning algorithms aspire to learn DAGs from data alone, it is also typically possible to incorporate existing expert knowledge into the learning process, in the form of *edge constraints*. In fact, this ability to encode prior knowledge in this way is a key advantage of BNs [7]. In addition to narrowing the search space of DAGs and thus making the algorithms more efficient, these constraints can help ensure that the learned DAGs abide by common sense, and are more causally plausible [35]. For example, temporal knowledge can be encoded by prohibiting the existence of an edge $X \rightarrow Y$ if Y precedes X temporally, as causality is typically understood only to operate in the direction of time [40]. Edge constraints are typically specified in the form of *whitelisted* or *blacklisted* edges, where if an edge is in the whitelist then it must be present in the learned DAG, and if it is in the blacklist then it must not be present in the learned DAG. It should be noted that while blacklist and whitelist constraints are symmetrical in some sense, there is a subtle asymmetry which is worth highlighting. Consider the possible relationships that may hold between X and Y in a CPDAG: either (1) $X \quad Y$ (X and Y are not directly related); (2) $X \rightarrow Y$; (3) $X \leftarrow Y$; or (4) $X - Y$ (bidirected). Blacklisting $X \rightarrow Y$ prohibits (1) and (4), but permits (2) and (3), thus leaving some option for the algorithm to decide whether or not X and Y should have some relationship (e.g. if $X \rightarrow Y$ is blacklisted, but the conditional independence tests find that in fact X and Y are dependent, then there is still the option to set $X \leftarrow Y$). However, when $X \rightarrow Y$ is whitelisted, there is no option but for X and Y to be related in some way, be it $X \rightarrow Y$ or $X - Y$, even if the conditional independence tests show that X and Y should be independent. This asymmetry is summarised by Table 3.1.

Table 3.1: An asymmetry between blacklisting and whitelisting.

Relationship	**Edge type**	**Blacklist** $X \rightarrow Y$	**Whitelist** $X \rightarrow Y$
Related	$X \rightarrow Y$	✗	✓
Related	$X \leftarrow Y$	✓	✗
Not related	$X \quad Y$	✓	✗
Related	$X - Y$	✗	✓

A number of experiments were developed to explore the effects of specifying such prior knowledge on learned structures, using a blacklist and whitelist for the small dataset that were specified by a domain expert for the purposes of this project, and are provided in Table 3.2. The PC algorithm was used for these experiments. One set of these experiments measures edge statistics (mean number of edges, mean edge variance) as increasing amounts of prior knowledge are provided to the learning algorithm. For each set of prior knowledge constraints, edge statistics are calculated on CPDAGs learned from 100 samples of size 10,000 from the small dataset. Results are provided in Section 4.1.3.

Another experiment aims to determine whether adding *some* prior knowledge increases the likelihood that the learned structure will satisfy other edge constraints hitherto *unseen* by the algorithm. This was evaluated by running PC with increasing amounts of blacklist constraints, and measuring the number of blacklist constraints that are satisfied by the learned CPDAGs. The degree to which the learned CPDAGs match the constraints is measured using the *F1-score* metric, which balances precision and recall on the constraints. A blacklist constraint $X \not\to Y$ is deemed satisfied by a CPDAG $\mathcal{D}$ if $X \to Y$ is *not* present in the $\mathcal{D}$. For each set of blacklist constraints used during learning, the resulting CPDAG will not contain *at least* those edges, so the F1-score will monotonically increase as more constraints are provided. As a baseline for comparison, we compare to the F1-score of a CPDAG that has increasing amounts of constraints imposed *after* the learning procedure, i.e. the CPDAG learned without blacklist constraints, and *then* a set of edges are removed. By comparing the F1-scores for CPDAGs learned by including constraints before versus after learning, we can see whether learning with prior knowledge constraints makes the algorithm more likely to satisfy other unseen constraints. For example, if learning with the constraint that $death \not\to sex$ makes the algorithm less likely satisfy $death \not\to age$. The results for this experiment are presented and discussed in Section 4.1.3.

3.2.4 Selecting DAGs for Inference

While it was possible to perform many of the experiments previously discussed on CPDAGs, it is necessary to refine these structures down to individual DAGs in order to perform standard BN parameter estimation and inference. It is possible to narrow down the set of DAGs implied by a CPDAG by manually orienting bidirected edges, however there may be so many bidirected edges in the CPDAG that the domain expert is required to specify almost as many edges as if they had just manually specified the DAG from in the first place, defeating the original purpose of the structure learning algorithm. One approach to reducing the number of bidirected edges a given CPDAG is to use *bootstrap sampling* to get confidence measures for the presence of each edge, and then filter out edges which have less than some confidence threshold α. This approach has been taken in a number of previous works applying BNs [9, 67]. In this case, 100 subsets of size 10,000 were randomly sampled from the small data set, and a CPDAG was learned from each one. The confidence for a given edge $X_i \to X_j$ was then computed as the

Table 3.2: Edge blacklist and whitelist for the small dataset.

Blacklist	
From	**To**
death	no_comorbid
death	sex
death	age
death	rr_vsorres
death	oxy_vsorres
death	daily_gcs_vsorres
death	daily_bun_lborres
death	daily_crp_lborres
age	sex
rr_vsorres	sex
rr_vsorres	no_comorbid
rr_vsorres	age
oxy_vsorres	sex
oxy_vsorres	no_comorbid
oxy_vsorres	age
daily_gcs_vsorres	sex
daily_gcs_vsorres	no_comorbid
daily_gcs_vsorres	age
daily_bun_lborres	sex
daily_bun_lborres	no_comorbid
daily_bun_lborres	age
daily_crp_lborres	sex
daily_crp_lborres	no_comorbid
daily_crp_lborres	age

Whitelist	
From	**To**
no_comorbid	death
age	death
age	no_comorbid
sex	death
sex	no_comorbid

MLE estimate for the probability of that edge being present, $\hat{p}_{i,j}$, which edge presence modelled as a Bernoulli variable, as in Section 3.2.2. an R ShinyApp[3] was developed to determine a suitable threshold α which could be used to reduce density and bidirected edges in the CPDAGs without extensive manual labour. The app used the `networkD3` R package to create interactive visualisations of the CPDAGs with edge confidence $\hat{p}_{i,j}$ being indicated by opacity. Sliders were used to change the value for α, which would update the DAG visualisation live. A screenshot of the app interface is provided in the Appendix A Figure A.1.

A desirable value for α is one that makes the DAGs sparse and removes bidirected

[3] `https://shiny.rstudio.com/`

edges while retaining useful information, making them more interpretable and efficient while maximally accurate. $\alpha = 0.5$ was chosen as a threshold for all graphs, maintaining some of their individual characteristics, e.g. NOTEARS creating a slightly denser graph, while making them usable for inference and interpretation. Some minor changes were still necessary in order to remove remaining bidirected edges, but it was possible to resolve most of these with common sense, e.g. C-reactive protein $\not\to$ sex, and the rest with relative edge confidence. Detailed diagrams for each selected DAG are given in Appendix B. Note that prior knowledge constraints have not been applied to these DAGs other than to orient bidirected edges, and so the DAGs are likely to be causally infeasible – they are thus only used in experiments missing data and predictive performance where causal assumptions need not be made, and the model can be considered a purely associational BN.

3.2.5 DAG Comparison

One way to understand the differences between structure learning algorithms is to compare the DAGs they produce. *Structural Hamming Distance* [73] (SHD) is a distance metric for DAGs [18] that is calculated between two DAGs $\mathcal{D}$ and $\mathcal{D}'$ as the number of structural operations required to transform $\mathcal{D}$ into $\mathcal{D}'$. These structural operations are: (1) adding an edge; (2) removing an edge; and (3) reversing an edge orientation. The SHD between each pair of DAGs produced via the method described in Section 3.2.4 was computed, and is reported and discussed in Section 4.1.4.

3.3 Inference

3.3.1 Prior Knowledge Constraints

Some experiments on prior knowledge constraints were discussed in Section 3.2, but these mostly focused on the effects on structural properties. Further experiments were designed to evaluate the effect of adding prior knowledge constraints on prediction accuracy. The experimental setup is similar to those described in Section 3.2.3: sets of increasing amounts of prior knowledge constraints are created, and a CPDAG is fit to the small data set for each set of constraints. Then, for each DAG specified by a given CPDAG, parameters are estimated on the entire small data training set, and the *death* variable is predicted for a held-out validation set of size 1000 using likelihood weighting. Finally, the mean and standard deviation deviation of the accuracy of those predictions are recorded for each CPDAG. The results to this experiment are presented and discussed in Section 4.2.1.

3.3.2 Missing Data

As mentioned in Chapter 1, the ability to handle missing data during inference without using external imputation models is a key advantage of BNs. In order to quantify how well these models can handle missing data, the following experiment was designed. First, the small dataset was split into a $75 : 25$ train : validation split. Missing values in the train set were imputed using MICE, so that differences in model performance could arise only from inference. Parameters were estimated on this imputed train set using MLE for each of the BN structures selected in Section 3.2.4 and a logistic regression model. Then, a number of copies of the validation set were created, and increasing amounts of values were randomly removed to create artificial missing data, MCAR. For example, the validation set with 50% missing data had a 0.5 probability of replacing any given cell in the dataset with `NA`. Each model then made predictions on the *death* variable for each of these missing-data validation sets. The logistic regression model expects each feature to be observed during inference, so the MICE model fit to the training set was reused to impute the missing values during inference. This was done using the `mice.reuse` R function[4]. Because the *death* variable is imbalanced (75% *no*, 25% *yes*), we report *balanced accuracy*, which takes the mean of the individual class-level accuracies. Results provided in Section 4.2.2.

3.3.3 Classification Performance

A primary objective in risk stratification is accurately predicting the probability of various health outcomes for a given patient. In order to assess the predictive power of BNs, we consider a number of standard procedures for evaluating classification models. These procedures were computed using the small dataset train/validation split with each of the BNs selected in Section 3.2.4, as well as a logistic regression model, as was used in the 4C-score model. The evaluation methods computed are the following:

- **Receiver operating characteristic** (ROC) curve: this method plots the model's *sensitivity* and *specificity* for each possible decision boundary. This curve provides a means of visually evaluating a model's ability to discriminate between the two

[4] `https://www.rdocumentation.org/packages/NADIA/versions/0.4.1/topics/mice.reuse`

classes in the variable being predicted, where a model is a better discriminator if it has a larger area under the curve (AUC) [39].

- **Calibration curve**: this plot gives an indication of the reliability of the probabilities output by a model [48]. This method asks the question ”In what proportion of the samples for which Y was predicted to be 1 with probability p was Y actually equal to 1?” If that proportion is always equal to p, then the calibration curve is a straight line, and the output probabilities can be considered to reliably indicate the probability that $Y = 1$.

- **Decision curve analysis**: this method indicates the *net benefit* resulting from the decision to administer some treatment for each output probability *decision threshold* [77]. In the health context, one could imagine a model which predicts the probability p of some disease being present, that could be cured with a certain treatment T. The decision threshold, p_d, is the value of p at which T would be administered. If T is likely to cause adverse effects, then p_d will be high, because we will only want to administer T if we are very certain that the disease is present [78]. Net benefit is then calculated as in Equation 3.2 using p_d to weight the negative effect of false positives, where high p_d indicates a large negative effect of false positive. TPR is the true positive rate, and FPR is the false positive rate. A line showing the net benefit if *all* patients are treated is also plotted as a baseline.

$$\text{Net benefit} = TPR - FPR\left(\frac{p_d}{1-p_d}\right) \tag{3.2}$$

3.3.4 Causal Inference

A causal DAG was manually specified by a domain expert in order to make a causal query about vaccination effectiveness using the do-operator, as described in Section 2.1.5.3. The variables included are those from the small set, along with a variable indicating patient vaccination status, whether they were admitted to ICU, and whether they required oxygen. The specified DAG is illustrated in Figure 3.1, with *severity of illness on admission* abstracting the variables indicating respirator rate, oxygen saturation, Glasgow coma score, urea, and C-reactive protein. Parameters for this model were estimated using the expectation maximisation algorithm, which can estimate parameters for both the observed variables and an unobserved variable representing mediators from COVID-

19 to outcomes. Note that unobserved confounders of $vaccination \rightarrow outcome$ could not be accounted for in the causal effect estimation, leaving potentially unobserved backdoor paths between $vaccination$ and outcomes. The `pgmpy` [6] python library was used to specify this model, estimate parameters, and run the interventional query. This gave estimates for interventional distributions of the form $P(outcome \mid do(vaccinated = yes))$ and $P(outcome \mid do(vaccinated = no))$, for each outcome: $death$, any_icu, any_oxy. These estimates were then used to compute the ATE of the vaccine on the outcomes, as defined in Section 2.1.5.3. For comparative purposes, a naïve estimation of the ATE was computed as follows, without using causal inference:

$$\text{Naïve ATE} = \mathbb{E}[P(outcome \mid vaccinated = yes)] - \mathbb{E}[P(outcome \mid vaccinated = no)]$$

The results to this experiment have not been validated by domain experts, and should thus be considered demonstrative. The unobserved confounders, marked in yellow in Figure 3.1, introduce further bias to the estimates.

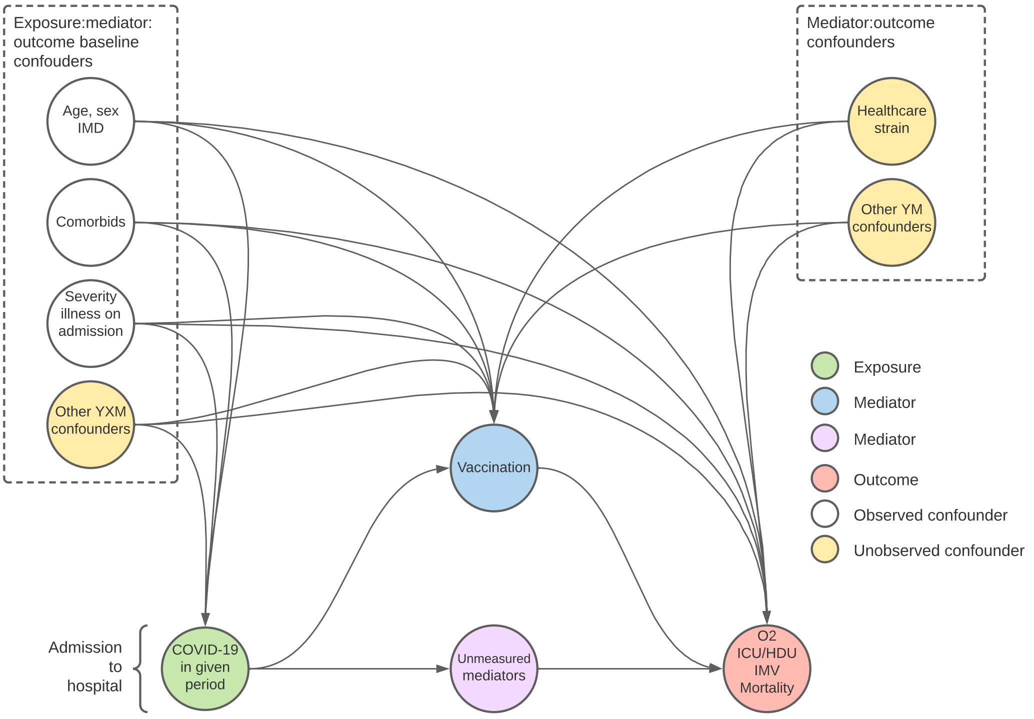


Figure 3.1: A causal DAG modelling the mechanism by which vaccination mediates the effect of COVID-19 on various health outcomes. The dashed boxes contain both observed and unobserved confounders.

Chapter 4

Results & Discussion

4.1 Structure Learning

4.1.1 Scalability

Results for the experiments described in Section 3.2.1 are given in Table 4.1, and plotted in Figures 4.1 and 4.2. It is clear from these plots that NOTEARS is signficantly slower than GES or PC for any of the dataset sizes considered. As the number of samples increases, runtime for GES and PC increases approximately linearly, whereas NOTEARS' runtime increases approximately exponentially. Conversely, as the number of variables increases, PC and GES become exponentially slower, whereas NOTEARS slows down approximately linearly. This asymmetry might lead one to prefer NOTEARS for very large datasets, however its absolute runtime is so long that will quickly become prohibitively expensive to use – note that the horizontal scaling experiments use a dataset with 1000 samples here; we can estimate that a dataset with 80 variables and 100,000 samples would take approximately 2 days to fit. GES is the fastest algorithm overall, scaling very well in the number of samples, and similarly to PC with the number of variables. PC scales slightly worse with the number of samples, likely due to the conditional independence test subroutine.

4.1.2 Edge Statistics

It is clear from the results in Table 4.2 that PC produces that sparsest DAGs, GES slightly less sparse, and NOTEARS produces that least sparse DAGs. Thus, under an associational (non-causal) interpretation, PC is producing DAGs with the most assumptions about the data, and NOTEARS is producing those with the least assumptions.

Table 4.1: Runtimes for each structure learning algorithm both as the number of samples increases (vertical scalability) and as the number of variables increases (horizontal scalability).

	Vertical Scalability			**Horizontal Scalability**		
	1000	**10000**	**100000**	**9**	**30**	**78**
PC	0.019	0.177	3.349	0.019	0.135	2.56
GES	0.017	0.027	0.081	0.017	0.181	2.117
NOTEARS	7.12	9.233	53.907	7.12	537.379	4847.333

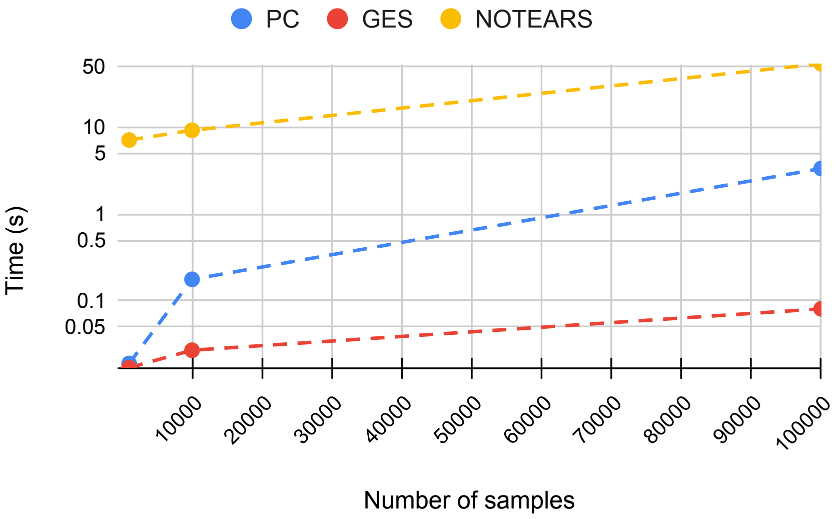


Figure 4.1: Runtimes for each structure learning algorithm as the number of variables increases.

Generally speaking, however, it is desirable that the learning algorithm produces sparse DAGs, both for improved computational efficiency and interpretability.

The average edge variance in the DAGs produced by these algorithms exhibit a much wider relative range in values, with PC and GES having an order of magnitude higher variance than NOTEARS. Overall, however, the average variances are quite low, corresponding to average probability of around 0.9 or 0.1 in PC and GES, and 0.99 or 0.01 in NOTEARS. This result is interesting because both PC and GES have statistical consistency guarantees [69, 14], meaning that they should find the true underlying DAG up to its MEC in the limit of data samples. These trials exhibit relatively low variance, and hence some level of convergence on an individual level, even at datasets with 10,000

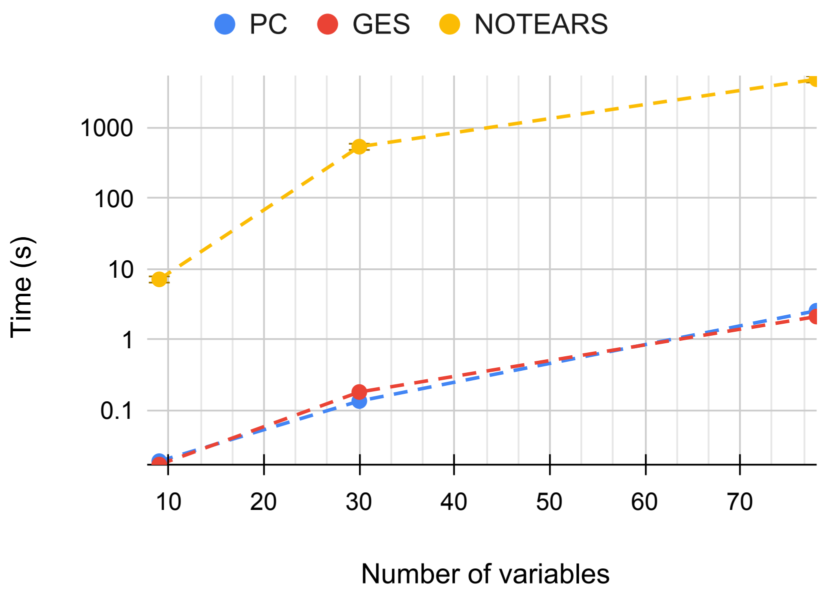


Figure 4.2: Runtimes for each structure learning algorithm as the number of samples increases.

samples, however the PC and GES don't tend to converge any more to one another than to NOTEARS, as discussed in Section 4.1.4.

Table 4.2: Edge statistics from CPDAGs fit to 100 randomly sampled subsets of the small dataset.

	Avg. No. Edges	**Avg. Edge Variance**
PC	18.14	0.086
GES	20.59	0.12
NOTEARS	26.04	0.0087

4.1.3 Prior Knowledge Constraints

Figure 4.3 shows the effect of adding blacklisted edges on mean number of edges and mean edge variance of CPDAGs produced by PC, as described in Section 3.2.3. Figure 4.3 (a) shows that increasing the number of blacklisted edges has a small effect on sparsity, decreasing the average sparsity from ~ 16.6 to ~ 15.6 by adding 24 blacklist constraints. This effect is perhaps smaller than one might expect, indicating the PC tends to compensate for removed edges by adding new edges that would not otherwise have been present. In fact, what is happening in this case is that if there is some association $X - Y$, but the blacklist contains $X \rightarrow Y$, then PC is simply orienting $X - Y$ as $X \leftarrow Y$. This is a desirable behaviour as it will enhance model interpretability, ensuring that the orientation of associational relationships $X - Y$ identified by PC are more likely to accord with common sense. Figure 4.3 (b) shows that blacklisting edges results in a significant decrease in mean edge variance in the resulting CPDAGs. This result indicates that blacklisting certain edges does not increase the variance of the remaining non-blacklisted edges, but rather causes a stable decrease in variance as more and more edges are added to the blacklist. A decrease in variance is desirable as it corresponds to more reliable DAGs that are robust to noise in the input. When examining the results on the effects of whitelisting in Figure 4.4, it is harder to observe clear trends as the whitelist was quite small, containing only 5 constraints. We can observe a slight trend towards an increased mean number of edges in Figure 4.4 (a), however, moving from an average of 17 edges with 0 constraints to an average of 19 with 5 whitelist constraints. This is expected, as whitelisting an edge $X \rightarrow Y$ ensures its presence whether or not PC finds that there should be some association between X and Y (see Table 3.1).

Figure 4.5 shows the effects on blacklist F1-score of providing PC with the constraints before learning versus imposing them on the resulting CPDAG afterwards. No clear difference between the two is evident in the plot, indicating that, beyond increased computational efficiency, there may not be significant benefits to providing the algorithm with these constraints before learning, at least in constraint-based algorithms.

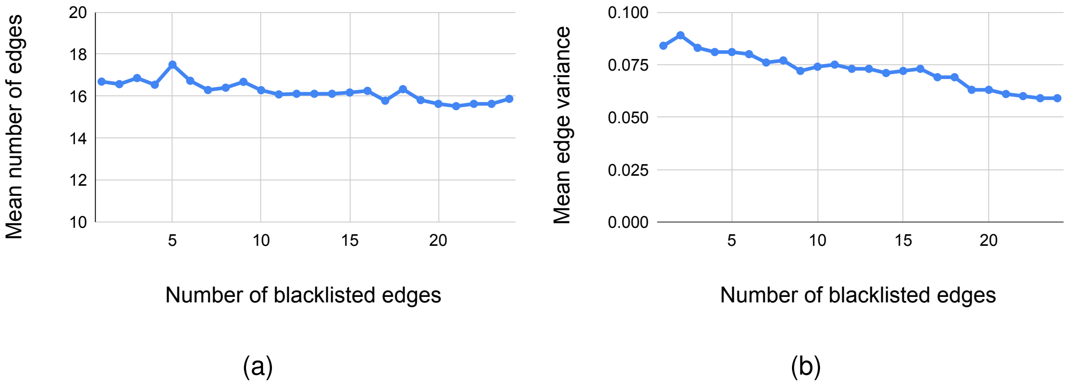
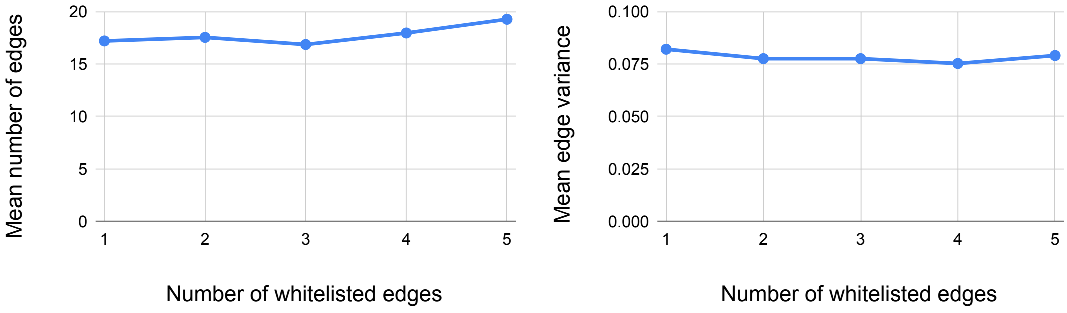


Figure 4.3: The effects of increasingly large blacklists on edge statistics.



(a) (b)

Figure 4.4: The effects of increasingly large whitelists on edge statistics.

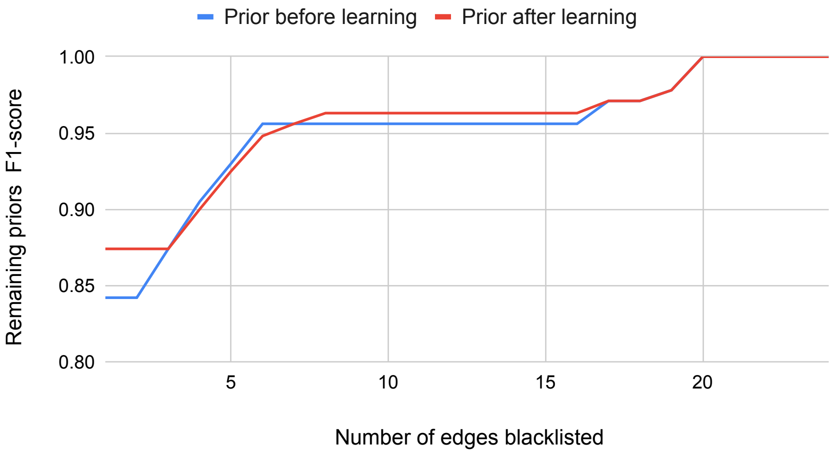


Figure 4.5: The F1-score on the remaining edge blacklist when priors are included before versus after structure learning.

4.1.4 DAG Comparison

Table 4.3 shows the structural Hamming distance between each pair of the DAGs created for the experiments on inference. It is interesting to note that each of these DAGs differ significantly from one another, with approximately 20 edge operations to convert any one DAG into any other. This suggests that learned structure can be quite sensitive to the algorithm chosen, a result which is corroborated by previous work in this area [36]. It is perhaps surprising that PC and GES have produced such different DAGs, when they should find the same MEC in the limit of data samples due to their consistency [25] – perhaps the assumptions necessary for this guarantee (e.g. sufficiency) have not been met, or there are simply too few samples.

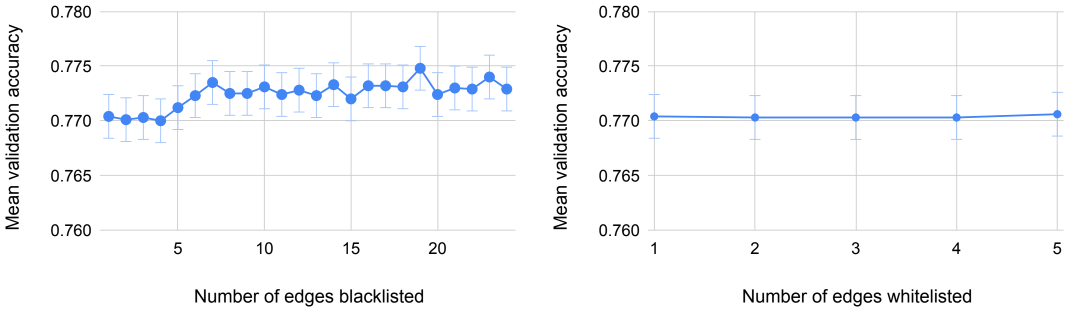
Table 4.3: Structural Hamming distance between DAGs learned by each algorithm.

	PC	**GES**	**NOTEARS**
PC	0		
GES	22	0	
NOTEARS	21	23	0

4.2 Inference

4.2.1 Prior Knowledge Constraints

The results to the experiment described in 3.3.1 are plotted in Figure 4.6. In Figure 4.6 (a) we can see an increase in mean accuracy when the blacklist increases from 4 to 7 constraints. By examining the blacklist in Table 3.2 it is clear that the first 8 constraints relate to the *death* variable, which is the variable being predicted in this case. These constraints prevent *death* from being the parent to any other variable (in accordance with temporality) and hence any relationship $X - death$ identified by PC must be oriented as $X \rightarrow death$, giving *death* more parents as more constraints are added. It is this increase in size of the parent set that is likely responsible for the increase in accuracy. No clear pattern emerges in Figure 4.6 (b) as edges are whitelisted, although this may be due to the small size of the whitelist: one could infer that whitelisting edges of the form $X \rightarrow death$ that increase the size of the parent set of *death* would increase accuracy.



(a)

(b)

Figure 4.6: Validation accuracy mean and standard deviation for CPDAGs fit with an increasing number of blacklisted edges.

4.2.2 Missing Data

Figure 4.7 shows the results to the experiment described in Section 3.3.2. The naive baseline plots the accuracy achieved by a model that predicts a single class for any input, receiving 100% accuracy on one class and 0% on the other, resulting in a 50% balanced accuracy. In examining the plot in Figure 4.7 it appears that each of the models, BNs and logistic regression alike, behave similarly in the face of missing data, the accuracy for each model decaying linearly towards the baseline as additional missing data is added. Of note in these results is that the BN models handle the missing data just as well as logistic regression without using any external imputation models, confirming that this is a strong point for BNs.

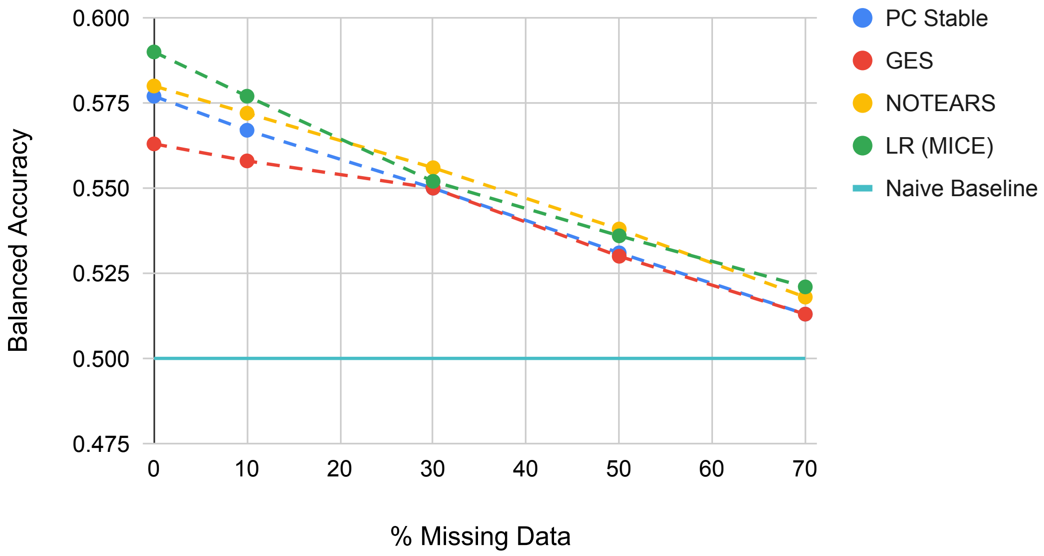


Figure 4.7: The effect of increasing amounts of missing data on validation accuracy for each of the selected DAGs (Section 3.2.4) and logistic regression.

4.2.3 Classification Performance

This section will present results to an evaluation of classification performance in BNs and logistic regression as described in Section 3.3.3. ROC curves for each BN model and the logistic regression are plotted in Figure 4.8. It is clear in examining this plot that logistic regression consistently outperforms the BNs, all of which have an almost identical ROC curve. The AUC for each model is provided in Table 4.4. Calibration curves are plotted in Figure 4.9. Again, the BNs' curves are almost identical, and overlap with one another. The curves for both Logistic Regression and the BNs are almost perfectly straight, apart from a slight deviation between 0.2 and 0.4, where logistic regression overestimates the probability and the BNs underestimate it. Overall, this result indicates that the output probabilities in both kinds of model are reliable. The output probability distributions for logistic regression and the PC BN are plotted as histograms in Figure 4.10. Note that the distribution produced by the Bayesian net is somewhat inconsistent, as compared with that of the logistic regression, which shows a smooth monotonic decrease in frequency as the probability increases, in line with the imbalance of positive samples for the dependent variable. Figure 4.11 plots a decision curve for each model. In general, logistic regression is provides as much or more net benefit than the BNs for any given risk threshold, with the two being similar at very high or low risk thresholds, and around 0.4. In aggregate, these results indicate that while the classification performance of logistic regression and discrete BNs is similar, logistic regression consistently shows a small improvement on standard metrics, corroborating findings from similar experiments in previous works [24, 18]. This result is not hugely surprising, as logistic regression is fit specifically to perform this classification task, whereas BNs are fit to model the entire joint distribution. In fact, it has been shown that, as discriminative classifiers, BNs constitute a subset of logistic regression models [56].

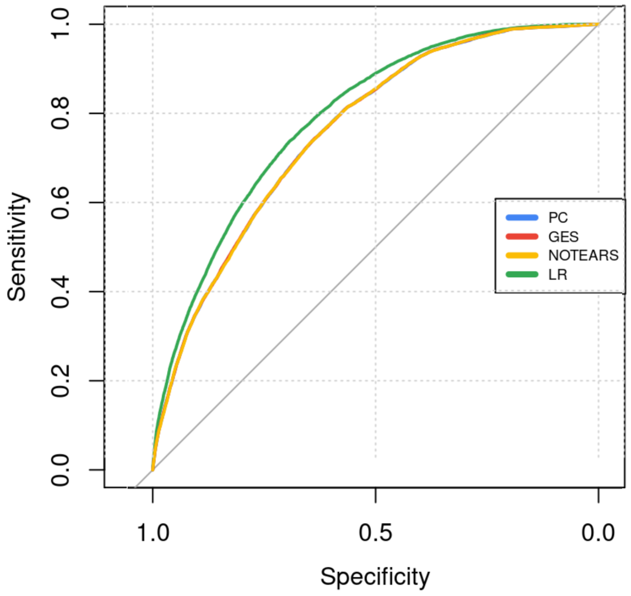


Figure 4.8: ROC curves for each BN and logistic regression.

Table 4.4: Area under ROC curves in Figure 4.8.

	AUC
LR	0.7746
PC	0.7603
GES	0.7604
NOTEARS	0.7602

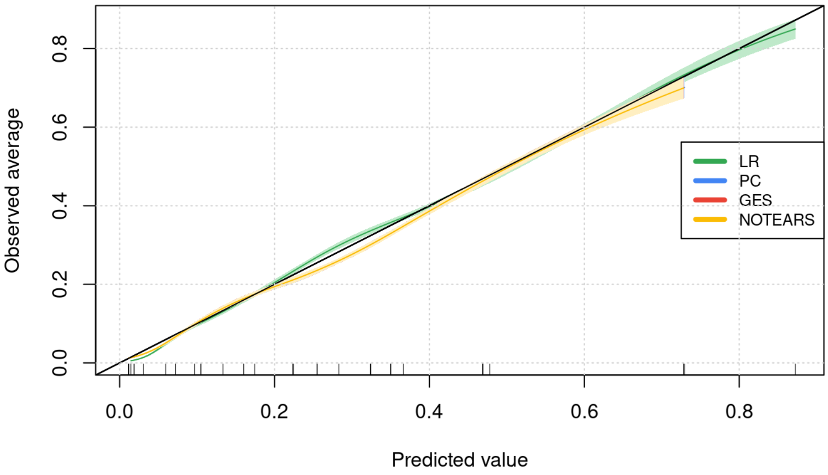
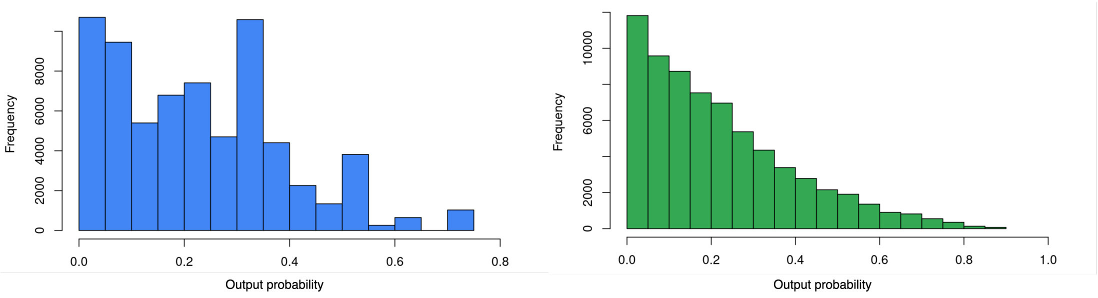


Figure 4.9: Calibration curve



(a)

(b)

Figure 4.10: Histograms showing distribution of probabilities predicted by the PC fitted BN (a) and logistic regression (b).

4.2.4 Causal Inference

The ATE was calculated for each health outcome using both the causal BN and a naïve observational approach, as described in Section 3.3.4. The resulting ATE estimates are presented in Table 4.5, where if outcome *death* has an ATE of 0.0457, then it is expected

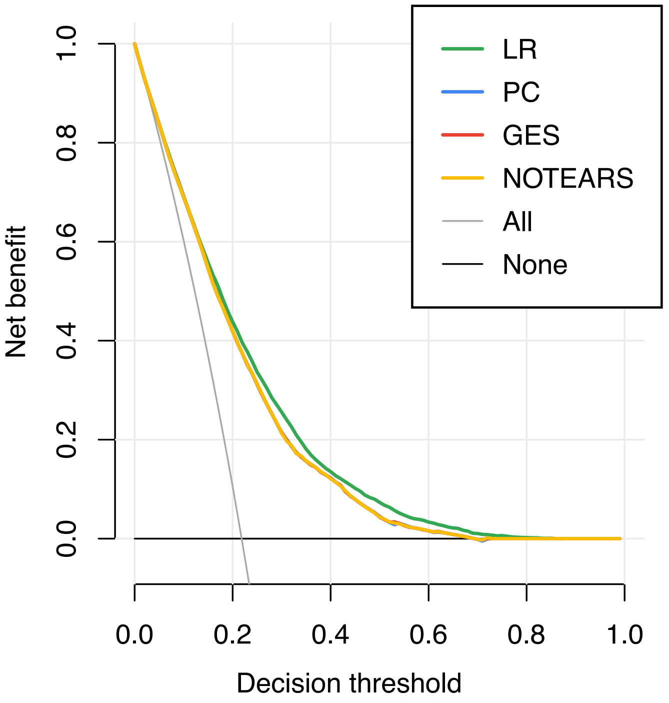


Figure 4.11: Decision curve for each BN and logistic regression.

that, on average, taking the vaccine will decrease probability of death by 0.0457. It is noteworthy that the naïve ATE estimate systematically overestimates the effect of the treatment, likely due to *vaccine* → *outcome* confounders. If, for example, older people are both more likely to receive the vaccine, and it is more effective for them at preventing negative health outcomes, then the conditional probability distribution $P(death \mid vaccine = yes)$ will be disproportionately represented by older people, and the effectiveness of the vaccine will appear greater. The expected values for each interventional and conditional distribution used to compute the ATE estimates are provided in Appendix C.

Table 4.5: Average treatment effects computed both naïvely and using a causal BN. The statistics indicate the expected *decrease* in probability of the associated health outcome caused by taking the vaccine.

Outcome	**Causal BN ATE**	**Naïve ATE**
death	0.0457	0.0556
any_icu	0.1277	0.1645
any_oxy	0.1445	0.177

4.3 A Finalised DAG

The following is an example of a general *workflow* for BN modelling, guided by the project's experimental results, that should produce interpretable and accurate DAGs (illustrated in Appendix E). Subject to validation by clinicians, the resulting DAG could be deployed for COVID-19 mortality risk prediction.

- The PC algorithm was chosen to learn the structure, because of its efficiency, transparent learning process and causal interpretation, and tendency to produce sparse DAGs.
- All available prior knowledge (in this case, specified in Section 3.2.3) was provided to the algorithm. This ensures that each DAG in the learned CPDAG is *causally plausible*[1]. Including prior knowledge about the variable to be predicted ($death$) that will increase the size of its parent set will help achieve high accuracy.
- Edge bootstrapping with $\alpha = 0.7$, as described in Section 3.2.4, was used to cut the CPDAG down to contain only the strongest edges.
- Finally, relative edge strength was used to orient two remaining bidirected edges, setting $oxy\_vsorres.factor \rightarrow rr\_vsorres.factor$ and $daily\_crp\_lborres.factor \rightarrow oxy\_vsorres.factor$.

The final DAG is plotted in Figure 4.12. In analysing the DAG structure, we see that *sex* and *age* are found to play a key role in mortality risk prediction, having both direct effects on mortality as well as through mediating factors including the number of comorbidities and blood urea nitrogen, aligning with the salient features identified by previous works [23, 34, 26]. C-reactive protein is also found to indirectly affect mortality through oxygen saturation. Interestingly, Glasgow Coma Score (GCS), measuring degree of consciousness, is not connected to any other variable, which contradicts findings by Gao et al. [23]. In this case it was found that the edge strength for each potential connection to or from GCS was approximately 0.2, and was thus cut from the CPDAG – this 'cutting' is analogous to feature selection via, e.g., LASSO.

[1] If the CPDAG implies a DAG that is *not* causally plausible, then this implies that there is some prior knowledge that hasn't been included.

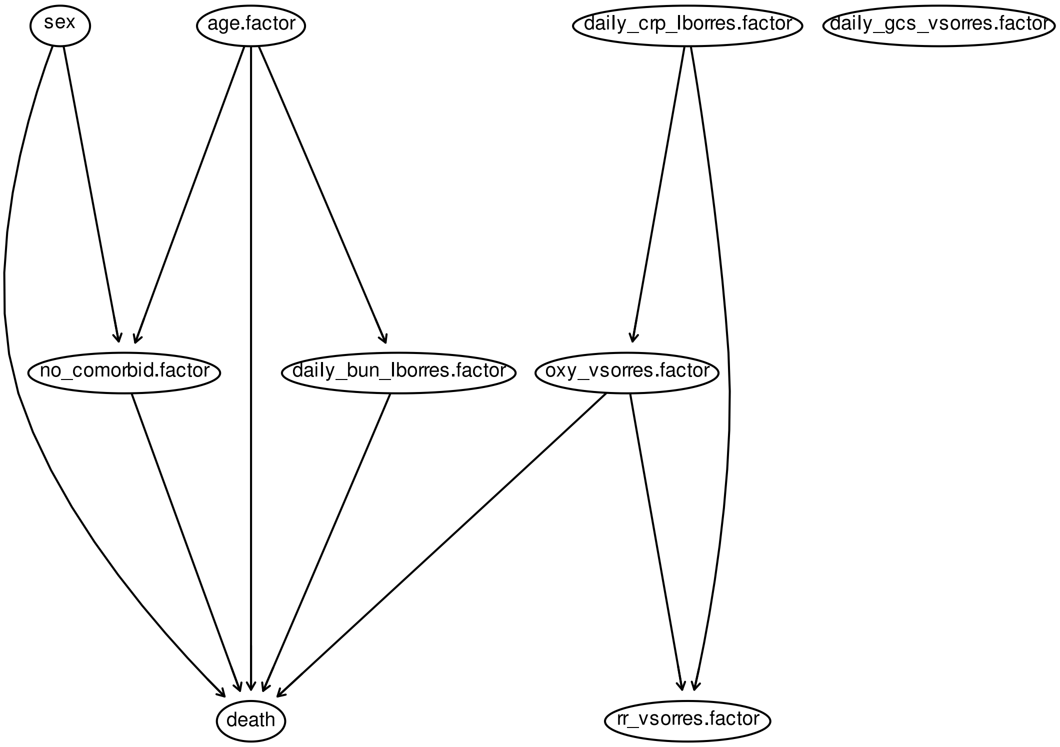


Figure 4.12: Plot of finalised DAG learned using PC stable, with all available prior knowledge and edge bootstrapping.

Chapter 5

Conclusion

In conclusion, this project has demonstrated that BN models are a viable alternative to standard methods of risk prediction, with some interesting properties that make them particularly suitable to the clinical domain. We showed that interpretable models of the domain can automatically be learned from data via structure learning, with prior expert knowledge being used to ensure that the learned model remains causally plausible. Three popular structure learning algorithms – PC, GES, and NOTEARS – were evaluated in terms of computational efficiency, and the properties of the structures they produced. These experiments identified the PC algorithm as a well-balanced choice, producing reliable, sparse DAG structures, and scaling well in the size of the dataset. In examining the effects of constraining the learning process with prior expert knowledge, it was found that edge blacklisting helps to produce more reliable, interpretable structures. The predictive performance of these models was also evaluated in a number of ways. One experiment found that BNs, without any external data imputation models, performed similarly to a logistic regression model using MICE imputation, even with large amounts (up to 70%) of the data missing. Using standard evaluation procedures for classification, such as computing AUROC, calibration curves, and decision curves, it was found that BNs are generally outperformed by logistic regression for classification, but only by a small amount (typically around 1%). A further experiment explored using a BN for performing causal inference, manually specifying a causal DAG with domain knowledge and then using interventional queries to compute average treatment effects of being vaccinated on various health outcomes. The treatment effects computed via this method were observed to be smaller than those seen using correlation, due to the interventional queries accounting for confounding bias. Finally, a generic workflow for BN modelling was proposed and employed in making a finalised model of the small dataset, which could potentially be deployed for real-world risk prediction, subject to further validation.

While the aforementioned experiments reveal some interesting results, they are not without limitation. In the experiment with prior knowledge specification before versus after the learning process, we only considered PC, a constraint-based algorithm. It would have been instructive also to try this with a score-based algorithm, which would likely behave differently. In general, the experiments with prior knowledge constraints were somewhat limited by the small size of the whitelist, making it hard to observe general patterns. It would also have been useful to conduct some experiments with larger graphs, in particular the missing data experiments, looking at the effect on accuracy of data missingness, although this was limited by compute time.

Many interesting research directions remain to be explored on this topic. For example, it would be interesting to explore ways of parameterising a BN model that can handle both discrete and continuous (mixed) data, and the impact that this can have on interpretability and predictive power. While discretising variables can approximate complex distributions well if the number of levels is high, this comes with a computational trade-off in discrete BNs. Opting instead for continuous distributions can drastically reduce the number of parameters necessary, and creates a new way to encode prior knowledge in the model. Some work has been done in this area, e.g. [72], and could be integrated with existing R code from this project. Further, it could be interesting to study the effects on predictive performance of modelling non-linear relationships between variables and their parents using more complex but nonetheless interpretable methods such as splines (e.g. [65]). It could also be interesting to develop parameter estimation and inference algorithms which operate on CPDAGs rather than DAGs. More work could be done in using BNs for causal inference. One experiment, for example, could involve a domain expert specifying as much causal domain knowledge as they can, and using the PC algorithm to learn an equivalence class in which each DAG could plausibly be the correct causal DAG (each should plausible because if it were not, then there would be some prior knowledge not encoded in the structure learning process). Then, intervention-calculus when the DAG is absent [45] could be used to estimate bounds on the causal effect of one variable on another. One could also move beyond BNs to make more complex causal queries, such as counterfactual queries with structural equation models [51]. Recently, causal machine learning methods such as the causal effect variational auto-encoder [44] have been developed, and claim to significantly improve in accuracy of estimating treatment effects.

Bibliography

[1] Muhammad Aurangzeb Ahmad, Carly Eckert, and Ankur Teredesai. Interpretable machine learning in healthcare. In *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics*, pages 559–560, 2018.

[2] Norah Alballa and Isra Al-Turaiki. Machine learning approaches in covid-19 diagnosis, mortality, and severity risk prediction: a review. *Informatics in Medicine Unlocked*, 24:100564, 2021.

[3] Shanmukh Alle, Akshay Kanakan, Samreen Siddiqui, Akshit Garg, Akshaya Karthikeyan, Priyanka Mehta, Neha Mishra, Partha Chattopadhyay, Priti Devi, Swati Waghdhare, et al. Covid-19 risk stratification and mortality prediction in hospitalized indian patients: Harnessing clinical data for public health benefits. *PloS one*, 17(3):e0264785, 2022.

[4] Paul D Allison. *Missing data*. Sage publications, 2001.

[5] Steen A Andersson, David Madigan, and Michael D Perlman. A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997.

[6] Ankur Ankan and Abinash Panda. pgmpy: Probabilistic graphical models using python. In *Proceedings of the 14th python in science conference (scipy 2015)*, volume 10. Citeseer, 2015.

[7] Paul Arora, Devon Boyne, Justin J Slater, Alind Gupta, Darren R Brenner, and Marek J Druzdzel. Bayesian networks for risk prediction using real-world data: a tool for precision medicine. *Value in Health*, 22(4):439–445, 2019.

[8] Melissa J Azur, Elizabeth A Stuart, Constantine Frangakis, and Philip J Leaf. Multiple imputation by chained equations: what is it and how does it work? *International journal of methods in psychiatric research*, 20(1):40–49, 2011.

[9] Zofia Baranczuk, Janne Estill, Sara Blough, Sonja Meier, Aziza Merzouki, Marloes H Maathuis, and Olivia Keiser. Socio-behavioural characteristics and hiv: findings from a graphical modelling analysis of 29 sub-saharan african countries. *Journal of the International AIDS Society*, 22(12):e25437, 2019.

[10] David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[11] Bryce A Basques, Ryan P McLynn, Michael P Fice, Andre M Samuel, Adam M Lukasiewicz, Daniel D Bohl, Junyoung Ahn, Kern Singh, and Jonathan N Grauer. Results of database studies in spine surgery can be influenced by missing data. *Clinical Orthopaedics and Related Research®*, 475(12):2893–2904, 2017.

[12] Paul Beaumont, Ben Horsburgh, Philip Pilgerstorfer, Angel Droth, Richard Oentaryo, Steven Ler, Hiep Nguyen, Gabriel Azevedo Ferreira, Zain Patel, and Wesley Leong. Causalnex, 10 2021. *URL https://github.com/quantumblacklabs/causalnex.*

[13] Dimitri Bertsekas and John N Tsitsiklis. *Introduction to probability*, volume 1. Athena Scientific, 2008.

[14] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

[15] Federico Cismondi, André S Fialho, Susana M Vieira, Shane R Reti, João MC Sousa, and Stan N Finkelstein. Missing data in medical databases: Impute, delete or classify? *Artificial intelligence in medicine*, 58(1):63–72, 2013.

[16] Diego Colombo, Marloes H Maathuis, et al. Order-independent constraint-based causal structure learning. *J. Mach. Learn. Res.*, 15(1):3741–3782, 2014.

[17] A Philip Dawid. Beware of the dag! In *Causality: objectives and assessment*, pages 59–86. PMLR, 2010.

[18] Martijn de Jongh and Marek J Druzdzel. A comparison of structural distance measures for causal bayesian network models. *Recent Advances in Intelligent*

Information Systems, Challenging Problems of Science, Computer Science series, pages 443–456, 2009.

[19] Annette J Dobson and Adrian G Barnett. *An introduction to generalized linear models*. Chapman and Hall/CRC, 2018.

[20] Annemarie B Docherty, Ewen M Harrison, Christopher A Green, Hayley E Hardwick, Riinu Pius, Lisa Norman, Karl A Holden, Jonathan M Read, Frank Dondelinger, Gail Carson, et al. Features of 20 133 uk patients in hospital with covid-19 using the isaric who clinical characterisation protocol: prospective observational cohort study. *bmj*, 369, 2020.

[21] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[22] Norman E Fenton, Scott McLachlan, Peter Lucas, Kudakwashe Dube, Graham A Hitman, Magda Osman, Evangelia Kyrimi, and Martin Neil. A bayesian network model for personalised covid19 risk assessment and contact tracing. *MedRxiv*, pages 2020–07, 2021.

[23] Yue Gao, Guang-Yao Cai, Wei Fang, Hua-Yi Li, Si-Yuan Wang, Lingxi Chen, Yang Yu, Dan Liu, Sen Xu, Peng-Fei Cui, et al. Machine learning based early warning system enables accurate mortality risk prediction for covid-19. *Nature communications*, 11(1):1–10, 2020.

[24] L Mary Gladence, M Karthi, and V Maria Anu. A statistical comparison of logistic regression and different bayes classification methods for machine learning. *ARPN Journal of Engineering and Applied Sciences*, 10(14):5947–5953, 2015.

[25] Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

[26] Xin Guan, Bo Zhang, Ming Fu, Mengying Li, Xu Yuan, Yaowu Zhu, Jing Peng, Huan Guo, and Yanjun Lu. Clinical and inflammatory features based machine learning model for fatal risk prediction of hospitalized covid-19 patients: results from a retrospective cohort study. *Annals of Medicine*, 53(1):257–266, 2021.

[27] Ewen Harrison and Pius Riinu. *R for Health Data Science*. Chapman and Hall/CRC, 2020.

[28] Alexander Hartemink and DK Gifford. *Principled computational methods for the validation and discovery of genetic regulatory networks. Massachusetts Institute of Technology*. PhD thesis, Ph. D. dissertation, 2001.

[29] Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13(1):2409–2464, 2012.

[30] Christina Heinze-Deml, Marloes H Maathuis, and Nicolai Meinshausen. Causal structure learning. *arXiv preprint arXiv:1706.09141*, 2017.

[31] Julia Hippisley-Cox, Carol AC Coupland, Nisha Mehta, Ruth H Keogh, Karla Diaz-Ordaz, Kamlesh Khunti, Ronan A Lyons, Frank Kee, Aziz Sheikh, Shamim Rahman, et al. Risk prediction of covid-19 related death and hospital admission in adults after covid-19 vaccination: national prospective cohort study. *bmj*, 374, 2021.

[32] Marcus Kaiser and Maksim Sipos. Unsuitability of notears for causal graph discovery. *arXiv preprint arXiv:2104.05441*, 2021.

[33] Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H Maathuis, and Peter Bühlmann. Causal inference using graphical models with the r package pcalg. *Journal of statistical software*, 47:1–26, 2012.

[34] Sujoy Kar, Rajesh Chawla, Sai Praveen Haranath, Suresh Ramasubban, Nagarajan Ramakrishnan, Raju Vaishya, Anupam Sibal, and Sangita Reddy. Multivariable mortality risk prediction using machine learning for covid-19 patients at admission (aicovid). *Scientific reports*, 11(1):1–11, 2021.

[35] Neville K Kitson, Anthony C Constantinou, Zhigao Guo, Yang Liu, and Kiattikun Chobtham. A survey of bayesian network structure learning. *arXiv preprint arXiv:2109.11415*, 2021.

[36] Neville Kenneth Kitson and Anthony C Constantinou. Learning bayesian networks from demographic and health survey data. *Journal of Biomedical Informatics*, 113:103588, 2021.

[37] Stephen R Knight, Antonia Ho, Riinu Pius, Iain Buchan, Gail Carson, Thomas M Drake, Jake Dunning, Cameron J Fairfield, Carrol Gamble, Christopher A Green,

et al. Risk stratification of patients admitted to hospital with covid-19 using the isaric who clinical characterisation protocol: development and validation of the 4c mortality score. *bmj*, 370, 2020.

[38] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[39] Rajeev Kumar and Abhaya Indrayan. Receiver operating characteristic (roc) curve for medical researchers. *Indian pediatrics*, 48(4):277–287, 2011.

[40] Douglas Kutach. The asymmetry of influence. 2011.

[41] Evangelia Kyrimi, Kudakwashe Dube, Norman Fenton, Ali Fahmi, Mariana Raniere Neves, William Marsh, and Scott McLachlan. Bayesian networks in healthcare: What is preventing their adoption? *Artificial Intelligence in Medicine*, 116:102079, 2021.

[42] Colleen L Lau, Helen J Mayfield, Jane E Sinclair, Samuel J Brown, Michael Waller, Anoop K Enjeti, Andrew Baird, Kirsty R Short, Kerrie Mengersen, and John Litt. Risk-benefit analysis of the astrazeneca covid-19 vaccine in australia using a bayesian network modelling framework. *Vaccine*, 39(51):7429–7440, 2021.

[43] Chang-Ju Lee and Kun Jai Lee. Application of bayesian network to the probabilistic risk assessment of nuclear waste disposal. *Reliability Engineering & System Safety*, 91(5):515–532, 2006.

[44] Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. *Advances in neural information processing systems*, 30, 2017.

[45] Marloes H Maathuis, Markus Kalisch, and Peter Bühlmann. Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, 37(6A):3133–3164, 2009.

[46] Scott McLachlan, Kudakwashe Dube, Graham A Hitman, Norman E Fenton, and Evangelia Kyrimi. Bayesian networks in healthcare: Distribution by medical condition. *Artificial Intelligence in Medicine*, 107:101912, 2020.

[47] Brady Neal. Introduction to causal inference from a machine learning perspective. *Course Lecture Notes (draft)*, 2020.

[48] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632, 2005.

[49] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

[50] Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

[51] Judea Pearl. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.

[52] Kazem Rahimi, Derrick Bennett, Nathalie Conrad, Timothy M Williams, Joyee Basu, Jeremy Dwight, Mark Woodward, Anushka Patel, John McMurray, and Stephen MacMahon. Risk prediction in patients with heart failure: a systematic review and analysis. *JACC: Heart Failure*, 2(5):440–446, 2014.

[53] Hans Reichenbach. *The direction of time*, volume 65. Univ of California Press, 1956.

[54] Alexander Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the simulated dag! causal discovery benchmarks may be easy to game. *Advances in Neural Information Processing Systems*, 34:27772–27784, 2021.

[55] Robert W Robinson. Counting labeled acyclic digraphs. *New directions in the theory of graphs*, pages 239–273, 1973.

[56] Teemu Roos, Hannes Wettig, Peter Grünwald, Petri Myllymäki, and Henry Tirri. On discriminative bayesian network classifiers and logistic regression. *Machine Learning*, 59(3):267–296, 2005.

[57] J Russell Stuart and Peter Norvig. *Artificial intelligence: a modern approach*. Prentice Hall, 2009.

[58] Mauro Scanagatta, Antonio Salmerón, and Fabio Stella. A survey on bayesian network structure learning from data. *Progress in Artificial Intelligence*, 8(4):425–439, 2019.

[59] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

[60] Marco Scutari, Catharina Elisabeth Graafland, and José Manuel Gutiérrez. Who learns better bayesian network structures: Accuracy and speed of structure learning algorithms. *International Journal of Approximate Reasoning*, 115:235–253, 2019.

[61] Marco Scutari and Robert Ness. bnlearn: Bayesian network structure learning, parameter learning and inference. *R package version*, 3:805, 2012.

[62] Andrew Selbst and Julia Powles. "meaningful information" and the right to explanation. In *Conference on Fairness, Accountability and Transparency*, pages 48–48. PMLR, 2018.

[63] Jonas Seng, Matej Zečević, Devendra Singh Dhami, and Kristian Kersting. Tearing apart notears: Controlling the graph prediction via variance manipulation. *arXiv preprint arXiv:2206.07195*, 2022.

[64] Cosma Shalizi. Advanced data analysis from an elementary point of view. 2013.

[65] Charupriya Sharma and Peter van Beek. Scalable bayesian network structure learning with splines. *arXiv preprint arXiv:2110.14626*, 2021.

[66] Jiang Shen, Fusheng Liu, Man Xu, Lipeng Fu, Zhenhe Dong, and Jiachao Wu. Decision support analysis for risk identification and control of patients affected by covid-19 based on bayesian networks. *Expert Systems with Applications*, 196:116547, 2022.

[67] Xinpeng Shen, Sisi Ma, Prashanthi Vemuri, and Gyorgy Simon. Challenges and opportunities with causal discovery algorithms: application to alzheimer's pathophysiology. *Scientific reports*, 10(1):1–12, 2020.

[68] Karamjit Singh, Garima Gupta, Vartika Tewari, and Gautam Shroff. Comparative benchmarking of causal discovery techniques. *arXiv preprint arXiv:1708.06246*, 2017.

[69] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search.* MIT press, 2000.

[70] Daniel Straub. Natural hazards risk assessment using bayesian networks. In *9th International Conference on Structural Safety and Reliability (ICOSSAR 05)*, 2005.

[71] Navdeep Tangri, Georgios D Kitsios, Lesley Ann Inker, John Griffith, David M Naimark, Simon Walker, Claudio Rigatto, Katrin Uhlig, David M Kent, and Andrew S Levey. Risk prediction models for patients with chronic kidney disease: a systematic review. *Annals of internal medicine*, 158(8):596–603, 2013.

[72] Michail Tsagris, Giorgos Borboudakis, Vincenzo Lagani, and Ioannis Tsamardinos. Constraint-based causal discovery with mixed data. *International journal of data science and analytics*, 6(1):19–30, 2018.

[73] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

[74] Laura Uusitalo. Advantages and challenges of bayesian networks in environmental modelling. *Ecological modelling*, 203(3-4):312–318, 2007.

[75] Stef Van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 45:1–67, 2011.

[76] Abhinav Vepa, Amer Saleem, Kambiz Rakhshan, Alireza Daneshkhah, Tabassom Sedighi, Shamarina Shohaimi, Amr Omar, Nader Salari, Omid Chatrabgoun, Diana Dharmaraj, et al. Using machine learning algorithms to develop a clinical decision-making tool for covid-19 inpatients. *International journal of environmental research and public health*, 18(12):6228, 2021.

[77] Andrew J Vickers and Elena B Elkin. Decision curve analysis: a novel method for evaluating prediction models. *Medical Decision Making*, 26(6):565–574, 2006.

[78] Andrew J Vickers, Ben van Calster, and Ewout W Steyerberg. A simple, step-by-step guide to interpreting decision curve analysis. *Diagnostic and prognostic research*, 3(1):1–8, 2019.

[79] Matthew J Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya like dags? a survey on structure learning and causal discovery. *ACM Computing Surveys (CSUR)*, 2021.

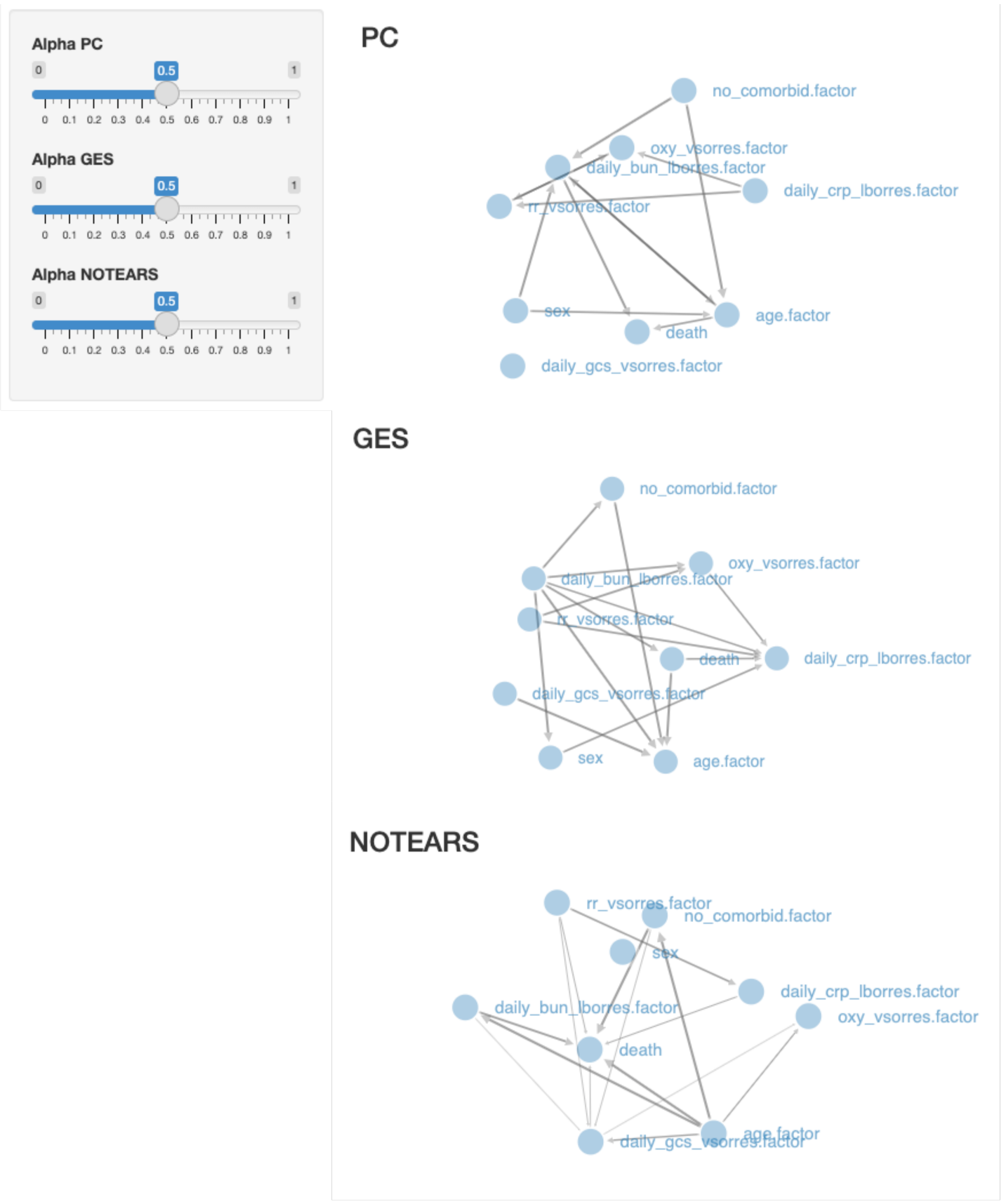
[80] Huayu Zhang, Ting Shi, Xiaodong Wu, Xin Zhang, Kun Wang, Daniel Bean, Richard Dobson, James T Teo, Jiaxing Sun, Pei Zhao, et al. Risk prediction for poor outcome and death in hospital in-patients with covid-19: derivation in wuhan, china and external validation in london, uk. *MedRxiv*, 2020.

[81] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Appendix A

ShinyApp for edge bootstrapping

See Figure A.1 on the next page.



Alpha PC
0.5

Alpha GES
0.5

Alpha NOTEARS
0.5

Figure A.1: The effect of increasing amounts of missing data on validation accuracy. This screenshot is for illustrative purposes, not to present the DAGs. For more readable DAG plots see Appendix B.

Appendix B

DAGs selected for inference experiments

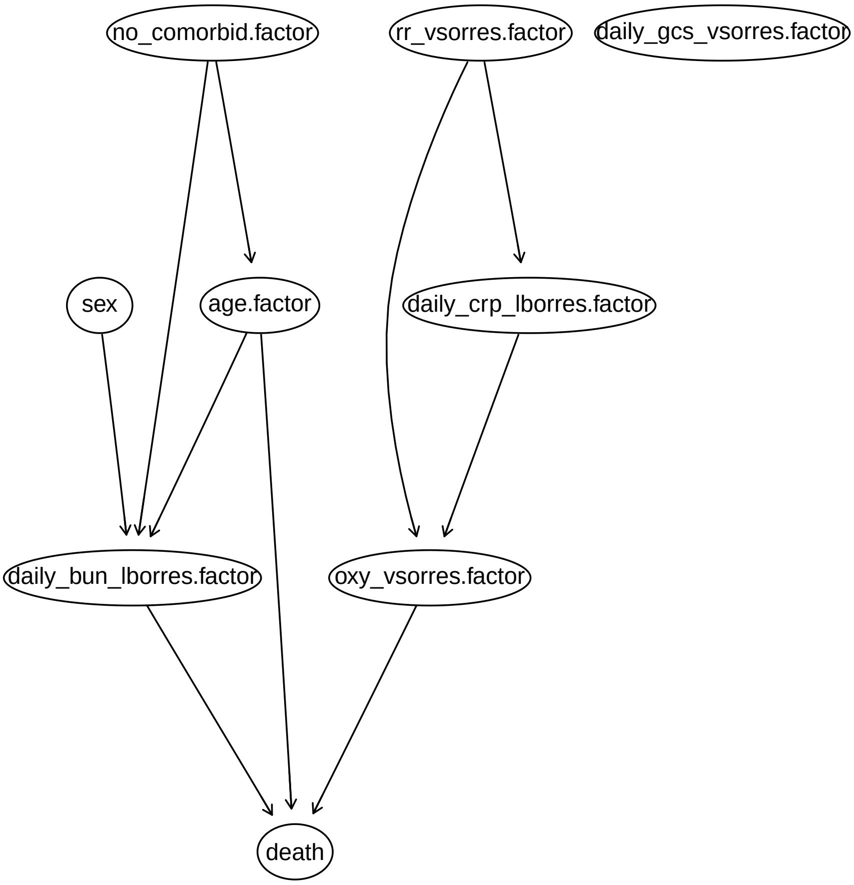


Figure B.1: Plot of selected DAG learned by PC.

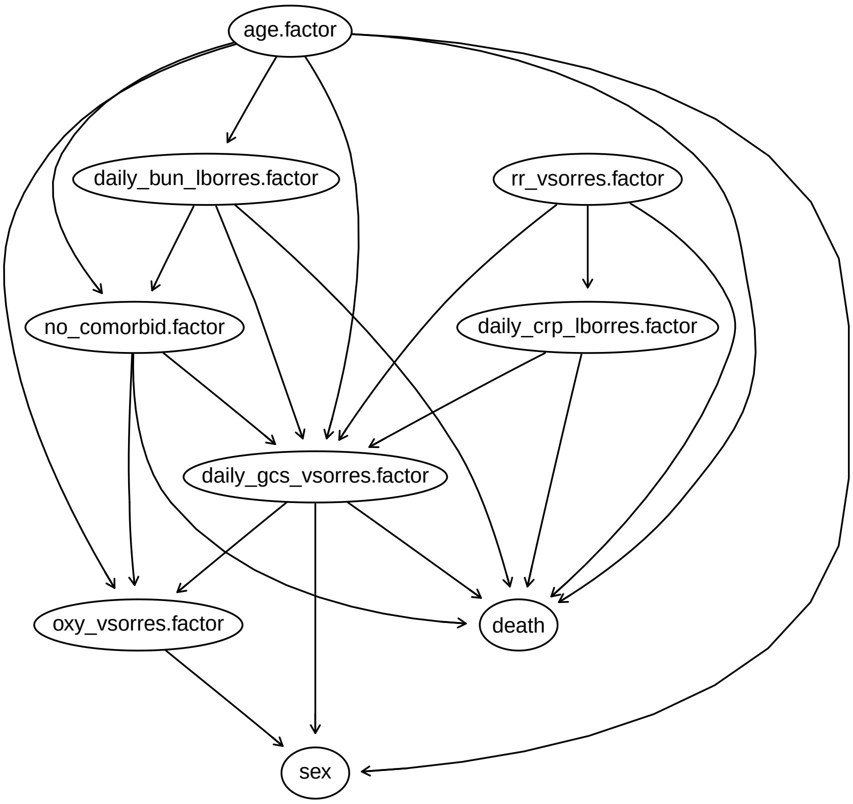


Figure B.2: Plot of selected DAG learned by NOTEARS.

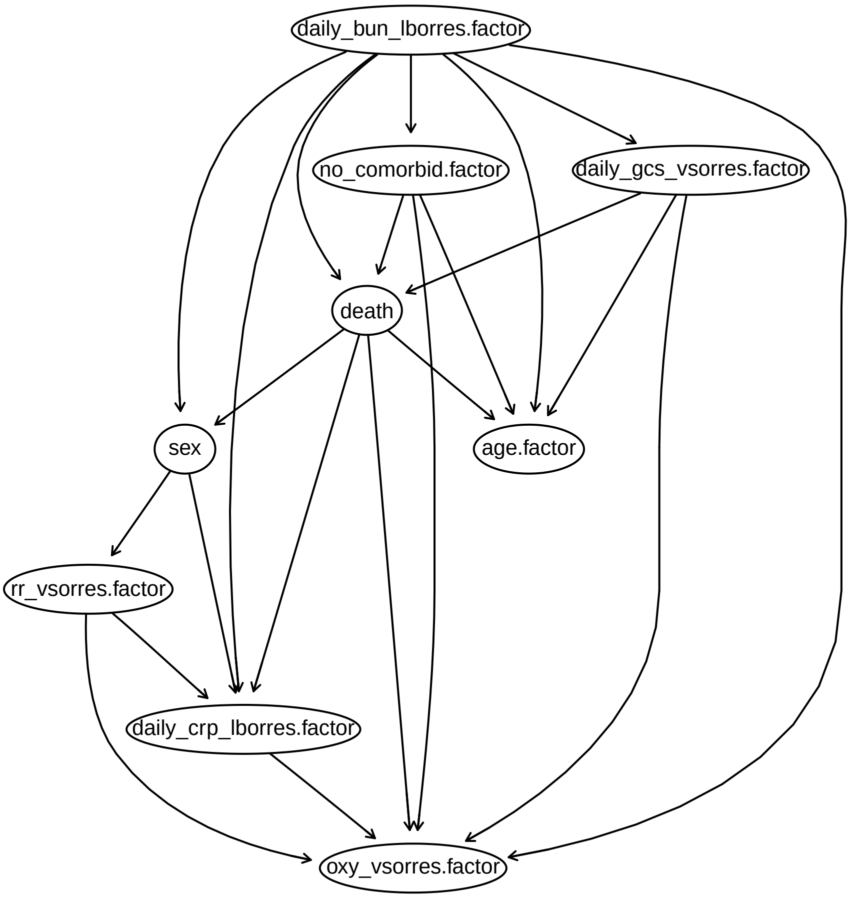


Figure B.3: Plot of selected DAG learned by GES.

Appendix C

Interventional and conditional expectations on health outcomes

Table C.1: Interventional and conditional expectations on the *death* (death) variable for various settings of the *vaccination_at_admission* (vac) variable.

death	$\mathbb{E}[\text{death} \mid do(\text{vac} = \text{yes})]$	$\mathbb{E}[\text{death} \mid do(\text{vac} = \text{no})]$	$\mathbb{E}(\text{death} \mid \text{vac} = \text{yes})$	$\mathbb{E}(\text{death} \mid \text{vac} = \text{no})$
yes	0.1585	0.2042	0.1452	0.2008
no	0.8415	0.7958	0.8548	0.7992

Table C.2: Interventional and conditional expectations on the *any_icu* (ICU) variable for various settings of the *vaccination_at_admission* (vac) variable.

ICU	$\mathbb{E}[\text{ICU} \mid do(\text{vac} = \text{yes})]$	$\mathbb{E}[\text{ICU} \mid do(\text{vac} = \text{no})]$	$\mathbb{E}(\text{ICU} \mid \text{vac} = \text{yes})$	$\mathbb{E}(\text{ICU} \mid \text{vac} = \text{no})$
yes	0.1049	0.2326	0.0745	0.2391
no	0.8951	0.7674	0.9255	0.7609

Table C.3: Interventional and conditional expectations on the *any_oxy* (oxy) variable for various settings of the *vaccination_at_admission* (vac) variable.

oxy	$\mathbb{E}[\text{oxy} \mid do(\text{vac} = \text{yes})]$	$\mathbb{E}[\text{oxy} \mid do(\text{vac} = \text{no})]$	$\mathbb{E}(\text{oxy} \mid \text{vac} = \text{yes})$	$\mathbb{E}(\text{oxy} \mid \text{vac} = \text{no})$
yes	0.5761	0.7206	0.5525	0.7295
no	0.4239	0.2794	4475	0.2705

Appendix D

Datasets

Table D.1: Generated by Spread-LaTeX

	Small dataset	**Note**
4C Score	death	-
	no_comorbid	Number of comorbidities
	age	-
	rr_vsorres	Respiratory rate
	oxy_vsorres	Oxygen saturation
	daily_gcs_vsorres	Glasgow Coma Score
	daily_bun_lborres	Urea
	daily_crp_lborres	C-reactive protein
	sex	-

Table D.2: Variables used in the medium dataset.

	Medium dataset
4C Score	death
	no_comorbid
	age
	rr_vsorres
	oxy_vsorres
	daily_gcs_vsorres
	daily_bun_lborres
	daily_crp_lborres
	sex
Further measures of general health	sysbp_vsorres
	admission_diabp_vsorres
	temp_vsorres
	hr_vsorres
	daily_hb_lborres
	daily_wbc_lborres
	daily_neutro_lborres
	daily_plt_lborres
	daily_sodium_lborres
	daily_bil_lborres
	daily_creat_lborres
	vac_at_admission
Comorbidities	chrincard
	asthma_mhyn
	modliv
	malignanteo_mhyn
	chronichaemo_mhyn
	aidshiv_mhyn
	obesity_mhyn
	diabetes_type_mhyn
	smoking_mhyn

Table D.3: Variables used in the large dataset.

	Large dataset
4C Score	death
	no_comorbid
	age
	rr_vsorres
	oxy_vsorres
	daily_gcs_vsorres
	daily_bun_lborres
	daily_crp_lborres
	sex
Further general measures of health	sysbp_vsorres
	admission_diabp_vsorres
	temp_vsorres
	hr_vsorres
	daily_hb_lborres
	daily_wbc_lborres
	daily_neutro_lborres
	daily_plt_lborres
	daily_sodium_lborres
	daily_bil_lborres
	daily_creat_lborres
	vac_at_admission
Comorbidities	chrincard
	asthma_mhyn
	renal_mhyn
	modliv
	malignanteo_mhyn
	chronichaemo_mhyn
	aidshiv_mhyn
	obesity_mhyn
	diabetes_type_mhyn
	smoking_mhyn
Further comorbidities	chronicpul_mhyn
	mildliver
	chronicneu_mhyn
	rheumatologic_mhyn
	dementia_mhyn
	malnutrition_mhyn
	vulnerable_no_nk
	vulnerable_transplant
	vulnerable_cancers
	vulnerable_copd
	vulnerable_scid
	vulnerable_immuno
	vulnerable_preg
Symptoms	dehydration_vsorres
	daily_plt_lborres
	adm_no_sypm
	fever_ceoccur_v2
	cough_ceoccur_v2
	coughsput_ceoccur_v2
	coughhb_ceoccur_v2
	sorethroat_ceoccur_v2
	runnynose_ceoccur_v2
	earpain_ceoccur_v2
	wheeze_ceoccur_v2
	chestpain_ceoccur_v2
	myalgia_ceoccur_v2
	jointpain_ceoccur_v2
	fatigue_ceoccur_v2
	shortbreath_ceoccur_v2
	aguesia_ceoccur_v2
	lowerchest_ceoccur_v2
	headache_ceoccur_v2
	confusion_ceoccur_v2
	seizures_ceoccur_v2
	abdopain_ceoccur_v2
	vomit_ceoccur_v2
	diarrhoe_ceoccur_v2
	conjunct_ceoccur_v2
	rash_ceoccur_v2
	skinulcers_ceoccur_v2
	lymp_ceoccur_v2
	bleed_ceoccur_v2
	bleed_ceterm_v2
	anosmia_ceoccur_v2
Outcomes	any_trach
	any_icu
	any_oxygen
	any_noninvasive
	any_invasive

Appendix E

Modelling Workflow

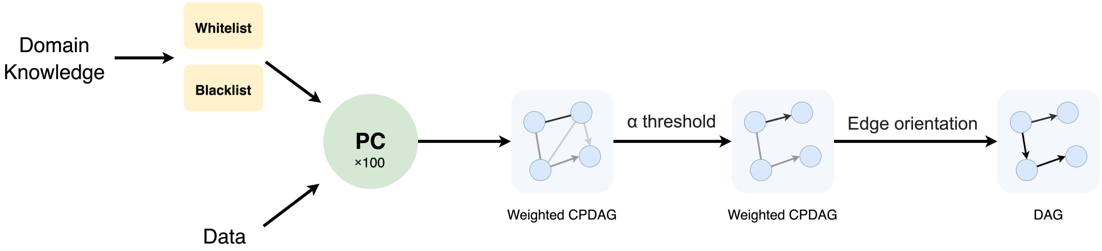


Figure E.1: Illustration of Bayesian network modelling workflow. PC algorithm is run 100 times on randomly sampled subsets of the dataset to produce a *weighted CPDAG*, in which each edge is weighted by the proportion of the 100 fitted CPDAGs it occurs in. Then edge weight thresholding is used to narrow the CPDAG down, with a final edge orientation phase to direct any remaining bidirected edges according to relative edge strength, producing a final DAG.

Appendix F

Visualising Conditional Probability Tables

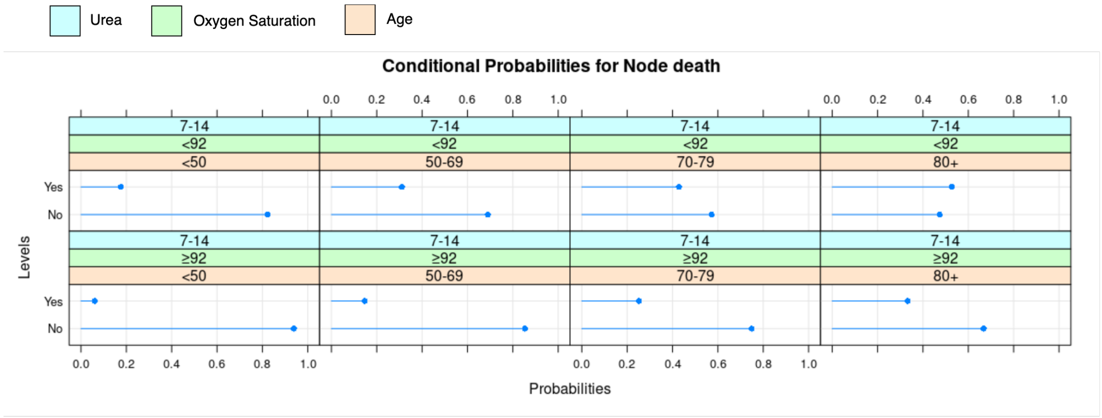


Figure F.1: In discrete BNs, each factor $P(X \mid \mathrm{pa}(X))$ is a conditional probability table, each can be visualised using intuitive plots, such as bar charts. See, for example, Figure F.1. The visualisation shows a probability for each level of the $death$ node, *yes* and *no*, for each combination of values of the parent nodes. For example, we can see on the bottom left panel of Figure F.1 that if a patient is young, has high oxygen saturation and a normal range of urea, they have a very low probability of in-hospital mortality. Note that as the size of the set of possible parent configurations grows, be that through increasing the number of parents or the number of levels in the parent factors, such diagrams become impractically large, further motivating DAG sparsity. The table depicted here comes from the *death* node for the DAG plotted in Figure B.1

.