Application of Bayesian networks to risk assessment

by

Jidapa Kraisangka

M.S., University of Pittsburgh, 2013

Submitted to the Graduate Faculty of

the School of Computing and Information in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2019

UNIVERSITY OF PITTSBURGH

SCHOOL OF COMPUTING AND INFORMATION

This dissertation was presented

by

Jidapa Kraisangka

It was defended on

April 10th 2019

and approved by

Marek J. Druzdzel, Professor, School of Computing and Information

Stephen C. Hirtle, Professor, School of Computing and Information

Paul W. Munro, Associate Professor, School of Computing and Information

Raymond L. Benza, MD, Allegheny General Hospital

Dissertation Director: Marek J. Druzdzel, Professor, School of Computing and Information

Application of Bayesian networks to risk assessment

Jidapa Kraisangka, PhD

University of Pittsburgh, 2019

Various approaches are used to estimate and predict risks. One of the most prevalent methods for risk assessment is the Cox's proportional hazard (CPH) model (Cox, 1972), a popular statistical technique used in risk estimation and survival analysis. The weaknesses of this approach are: (1) the underlying model can be only learned from data and is not readily amenable to refinement based on expert knowledge (2) the CPH model rests on several assumptions simplifying the interactions between the risk factors and the predicted outcome. While these assumptions are reasonable and the CPH model has been successfully used for decades, it is interesting to question them with a possible benefit in terms of model accuracy.

This dissertation focuses on theoretical and practical aspects of risk assessment based on Bayesian networks (Pearl, 1988) as an alternative approach to the CPH model. The dissertation makes three contributions: (1) I propose a Bayesian network interpretation of the CPH (BN-Cox) model, a process of using existing CPH models as data sources for parameter estimation in Bayesian networks when original data are not available, and discuss methods for modeling such model computationally tractable (2) I empirically demonstrate in both context-sensitivity of the strength of influences of individual risk factors on the outcome variables in both Bayesian network model and the CPH model, and finally, (3) I propose and evaluate methods for enhancing the quality of Bayesian network parameters learned from small data sets, by means of priors.

Table of Contents

List of Tables

List of Figures

Preface

This is an acknowledgment of appreciation for all the people who have supported me in my journey to achieving my doctorate degree over the years in Decisions Systems Laboratory (DSL), School of Computing and Information, University of Pittsburgh.

First, I would like to express my deepest gratitude to Dr. Marek J. Druzdzel, for the continuous mentoring advice of my Ph.D. study and research. Dr. Druzdzel provided me with every bit of guidance, assistance, and expertise that I needed to achieve every PhD milestone. I learned so much from him. I could not have imagined having a better advisor and mentor for my PhD journey.

Apart from my advisor, I would like to express my sincere gratitude to the members of my Ph.D. committee for their time and effort on my dissertation. I gratefully thank Dr. Raymond Benza from Allegheny Health Network for his mentorship and substantial support in my research. I would like to sincerely thank Dr. Stephen Hirtle and Dr. Paul Munro for their useful questions, feedback, and comments on my dissertation.

I greatly appreciate the support from all members of the PHORA project for their fruitful discussions and comments, that were broaden perspective of my research. I also want to thank my fellow lab-mates for the stimulating discussions and all the fun we had in the past years. Special thanks to Marcin Kozniewski for his helpful suggestions and commenton my work with Bayesian networks. Also Dr. Parot Ratnapinda for his useful suggestion and guidance throughout my study.

Thanks to my friends from Thailand for their supports and encouragement. I gratefully acknowledge the scholarship received towards my PhD from the Faculty of Information and Communication Technology, Mahidol University, Thailand. I greatly thank my mentor at Mahidol University, Dr. Jarernsri Mitrpanont for her sincere support and patience since my undergraduate. Last but not least, I would like to express deepest gratitude to my family, especially my mom who encouraged me to pursue my studies and always believe in me.

1.0 Introduction

Risk is often referred to the *probability* of occurrence for an undesirable outcome, such as the probability of patients developing a disease, the probability of patients dying from a disease, or the probability of patients being hospitalized in the next six months, etc. The process of describing and quantifying risks is called risk assessment (Covello and Merkhoher, 2013), and it often involves prediction of an outcome based on a set of risk factors.

Various approaches are used to estimate and predict risks including statistical methods, such as., survival analysis. One of the most prevalent methods for risk assessment is the Cox's proportional hazard (CPH) model (Cox, 1972), a popular statistical technique used in risk estimation and survival analysis. While CPH models are widely used, their weaknesses are: (1) the underlying model can be only learned from data and is not readily amenable to refinement based on expert knowledge (2) the CPH model rests on several assumptions simplifying the interactions between the risk factors and the predicted outcome. While these assumptions are reasonable and the CPH model has been successfully used for decades, it is interesting to question them with a possible benefit in terms of model accuracy.

In the scope of this dissertation, I propose an alternative approach to risk assessment based on Bayesian network (BN) (Pearl, 1988) models. Bayesian networks are acyclic directed graphs in which vertices represent random variables and directed edges between pairs of vertices capture direct influences between the variables represented by the vertices. The network captures the joint probability distribution among a set of variables both intuitively and efficiently, modeling explicitly independencies among them. A representation of the joint probability distribution allows for calculation of probability distributions that are conditional on a subset of variables. This typically amounts to calculating the probability distributions over variables of interest given observations of other variables (e.g., probability of one-year survival given a set of observed risk factors). There is also a well developed theory expressing the relationship between causality and probability and often the structure of a Bayesian network is given a causal interpretation. This is utmost convenient in terms of user interfaces, notably knowledge acquisition and explanation of results.

The structure of the dissertation is as follows. Chapter 2 introduces terms and concepts that are necessary for the remaining chapters. Chapter 3 focuses on my first attempt to risk assessment with Bayesian networks: a Bayesian network interpretation of the CPH (BN-Cox) model (Kraisangka and Druzdzel, 2014, 2018). I describe the use the CPH models as data sources in the process of parameter estimation for Bayesian networks. I successfully replaced the use of the CPH model in the REVEAL risk score calculator (Benza et al., 2010) with an BN-Cox-based risk score calculator (Kraisangka et al., 2016; Kraisangka and Druzdzel, 2018). The BN-based calculator reproduces the results of the REVEAL risk score calculator exactly. However, one of the challenges to applying the BN-Cox model is an exponential growth of the conditional probability tables (CPT) corresponding to the survival variables, as the number of risk factors increases (Kraisangka and Druzdzel, 2016, 2018). I evaluated two approaches to mitigate this problem: (1) decomposition of the underlying Bayesian network known as parent divorcing, and (2) simplifying the network structure by removing least influential risk factors. The BN-Cox model seems to be not decomposable and approximating of decomposition leads to high loss of accuracy. Hence, simplifying the network structure by removing the least influential risk factors by any statistical variable selection methods is recommenced when we have a data set to refit the simplified model. However, when data are not available, we can simplify the model by removing least influential risk factors based on both the value of β coefficients and the statistical significance. When removing risk factors, we suggest marginalization, as it leads to smallest error on the average.

In Chapter 4, I demonstrate that the assumptions of context invariance of hazard ratios in the CPH model is unrealistic. Bayesian networks model correctly varying magnitude of influence of risk factors as other factors are observed. I empirically compare the influence of risk factors in two models.

Chapter 5 discusses methods for enhancing the quality of Bayesian network parameters, as learned from small data sets, by means of different priors: priors from experts knowledge and priors from simplified probabilistic models, such as Tree-Augmented Naïve Bayes. I discuss and provide empirical evaluation of the proposed methods, which are useful in practice when we need to improvement model accuracy for Bayesian network in risk assessment. Finally, Chapter 6 summarizes the dissertation, limitations, and directions of future work.

2.0 Background

This chapter introduces concepts that are necessary for my dissertation: (1) survival analysis techniques (Section 2.1) including Kaplan-Meier estimates and Cox's proportional hazard model and (2) Bayesian networks (Section 2.2).

2.1 Survival analysis

Survival analysis is a set of statistical methods that aim at modeling the relationship between a set of predictor variables and an outcome variable and, in particular, prediction of the time when an event occurs (Allison, 2010). For example, researchers may focus on time-to-death of patients with a specific disease, failure time of machines, or time to rearrest of prisoners who have been released. Survival analysis can be used to estimate time-to-event for a group, to compare risks among study groups, or to study the relationship between variables to the predicted events.

The probability of an individual surviving beyond a given time t, i.e., the survivor function, is defined as

$$S(t) = Pr(T > t) \ . \tag{2.1}$$

T is a variable denoting the time of occurrence of an event of interest. The survival probability at the beginning, i.e., t_0, may be equal to 1 or to some baseline survival probability, which will drop down to zero over time. While survivor function represents the probability of survival, the hazard function represents the risk of event occurrence at time t. The hazard is a measure of risk at a small time interval $\triangle t$ which can be considered as a rate (Allison, 2010). The hazard function is given by

$$\lambda(t) = \lim_{\triangle t \to 0} \frac{Pr(t \leq T < t + \triangle t \mid T \geq t)}{\triangle t} \ , \tag{2.2}$$

where T is also a time variable. The relationship between the hazard function and the survivor function is described as

$$\lambda(t) = -\frac{d}{dt} \log S(t) \tag{2.3}$$

or as

$$S(t) = \exp \int_0^t \lambda(u)\, du\ . \tag{2.4}$$

Hence, we can estimate the survival probability from the hazard function, and vice versa. Several techniques has been used to estimate the hazard function or the survivor function which are broadly classified into parametric regression model, non-parametric model, or semi-parametric models. Parametric regression model assume certain distribution underlying the hazard function. The distributions can follow normal, uniform, exponential, Weibull, or log-normal distributions. On the other hand, a non-parametric model does not have any assumptions for distribution, however, the model, such as Kaplan-Meier estimates, is widely used to depict the structure of survival data. Semi-parametric models restricts partial assumptions about the models, for example, Cox's proportional hazard model assume the ratio between the baseline hazard and the hazard with a specific risk factor is constant over time.

The focus of this dissertation is to investigate the application of Bayesian network in risk assessment against traditional survival analysis techniques which is Cox's proportional hazard model. However, I also use Kaplan-Meier estimates to depict a survival data in the experiment. Both survival analysis methods are widely used particularly in medicine which I will provide more details in the following sections.

2.1.1 Cox's proportional hazard model

The Cox's proportional hazard model (Cox, 1972) is a set of regression methods used in the assessment of survival based on its risk factors or explanatory variables. The risk factors can be time-independent (e.g., race or sex) or time-dependent, which can change throughout the study (e.g., blood pressure at different points of study time). In the scope of this dissertation, I focus only on the CPH model with time-independent risk factors. This

model allows researchers to evaluate and control factors that affects the time to event (Klein and Moeschberger, 2003).

As defined originally by Cox (1972), the hazard function is expressed as

$$\lambda(t) = \lambda_0(t) \exp(\beta' \cdot \mathbf{X}) \ . \tag{2.5}$$

The function is composed of two main parts: the baseline hazard function, $\lambda_0(t)$, and the set of risk factors, $\beta' \cdot \mathbf{X} = \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n$. The baseline hazard function determines the risks at an underlying level of explanatory variables, i.e., when all risk factors are absent. According to Cox (1972), this $\lambda_0(t)$ can be unspecified or follow any distribution, which makes the CPH model semi-parametric. The βs are coefficients corresponding to the risk factors, $\mathbf{X_i}$. The coefficient represents the effect of the risk factor in the model.

CPH models can handle both continuous and discrete variables (Allison, 2010). The CPH model treats these risk factors as numerical variables, so that the model can estimate the parameter coefficients, β. Researchers can treat risk factors as they are defined in the data set or do some data preprocessing. For example, in case of categorical variables with n categories, researchers need to create a set of dummy binary variables capturing $n-1$ categories, e.g., we can code a variable *color* having values as red, green, blue, as two binary variables (e.g., *color-red* and *color-green*). Some continuous variables, e.g., number of days in a hospital, can also be discretized. Once all risk factors have been established, β parameters are estimated by means of the Maximum Partial Likelihood technique.

Application of the CPH model relies on the assumption that the hazard ratio of two observations is constant over time (Cox, 1972). The hazard ratio is defined as γ:

$$\gamma = \frac{\lambda_2(t)}{\lambda_1(t)} = \frac{\exp(\beta' X_2)}{\exp(\beta' X_1)} \ . \tag{2.6}$$

In the most situation, hazard ratio is used to define the effect of an interested group to the baseline group. For example, in the study of patients with pulmonary arterial hypertension (PAH) (Benza et al., 2010), the hazard ratio of a group of PAH patients having renal insufficiency to a group of patients without renal insufficiency (baseline group) is reported as 1.90. This means that those patients with renal insufficiency have a 90% higher risk of

dying from PAH disease than patients without renal insufficiency. The ratio represents the relative risk of these two observations with different state of risk factors at time t.

Once we know the hazard ratio, we can estimate their survival probability at time t of the group of interest relative to baseline group from

$$S(t) = S_0(t)^{\gamma} = S_0(t)^{\exp(\beta' \cdot \mathbf{X})} \ . \tag{2.7}$$

$S_0(t)$ is the baseline survival probability estimated from data, i.e., when all risk factor are absent or at their baseline value ($X = 0$) at any time t, while γ is the hazard ratio of the group of interest to the baseline group.

In medicines, CPH models is commonly used for evaluating treatment effect and predicting patient prognosis . For example, the Seattle Heart Failure Model (Levy et al., 2006) uses a CPH model to predict 1-, 2-, and 3-year survival of heart failure patients. The Registry to Evaluate Early and Long-Term Pulmonary Arterial Hypertension (PAH) Disease Management (REVEAL) (Benza et al., 2010) uses also a CPH model at the foundation of to its Risk Score Calculator, that determines the probability of a PAH patient survival.

Example 1. A classical example application of the CPH model is an experimental study of recidivism of prisoners by Rossi et al. (1980). The data set was collected in the course of an experimental study of 432 male prisoners, who were under one year observation after being released from prison. The event of interest in this analysis is *arrest*, i.e., whether the prisoner is re-arrested during the period of study or not. The Recidivism data set is quite likely the most widely used example data set for survival analysis (Allison, 2010; Fox, 2002), especially for the CPH model.

The original data set consists of 62 variables (Rossi et al., 1980), including:

- **week**: the week when a prisoner was rearrested after having been released from prison.
- **arrest**: the rearrest status of a prisoner (rearrested = 1, never-rearrested = 0).
- **fin**: financial aid status after being released (no-financial-ai or has-financial-aid).
- *age*: the age in year at the time of being released.
- **race**: prisoner's race (others or black).
- **wexp**: status of having prior full-time working experience (yes or no).

Table 1: A list of risk factors with their parameters estimated for the CPH model

Variables	β	exp(β)	lower .95	upper .95	p-value
fin	-0.3899	0.6771	0.4664	0.9829	0.0403
$race$	0.2591	1.2958	0.7110	2.3617	0.3974
$wexp$	0.5249	0.5916	0.4038	0.8667	0.0071
$prio$	0.3330	1.3951	0.8462	2.3001	0.1918

- *mar*: marital status at the time of being released (single or married).
- *paro*: status of being released on parole (yes or no).
- **prio**: number of prior convictions.
- *educ*: level of education.
- $emp1 - emp52$: a list of variables indicating employment status of each week.

For the sake of simplicity, I selected only four risk factors (highlighted in bold) from the seven risk factors in the original Recidivism data set and preprocessed them into binary variables. The selected variables included the financial aid status *fin* (*no*=0, *yes*=1), prisoner's *race* (*other*=0, *black*=1), having prior full-time work experience *wexp* (*yes*=0, *no*=1), and number of prior convictions *prio* (*five and below*=0, *more than five*=1). The time variable in this data set is *week*, which is the week when a prisoner was rearrested during the observation period of one year (52 weeks). The survival variable is *arrest* indicating the rearrest status of a prisoner (*rearrested*=1, *never-rearrested*=0). I used R with the package *survival* and package *survminer* to create a CPH model and visualize survival curves. The function *coxph* was used to model the survival variable *arrest* with the selected risk factors based on each *week*. Table 1 shows the parameters of the constructed CPH model.

The β of each variable represents the coefficient in the model while the exp(β) is the multiplicative effect of the hazard (Fox, 2002), i.e., hazard ratio. The lower and upper bounds of the 95% confidence interval are in the third and fourth columns respectively. The fifth column indicates statistical significance of the β coefficient of the risk factor. From

the model estimation, fin and $wexp$ are significant ($p < 0.05$), while $race$ and $prio$ are not ($p > 0.05$). However, the overall test of the model are significant including likelihood ratio test ($p = 0.00313$), the Wald test ($p = 0.002736$), and the logrank test ($p = 0.002313$). Based on these parameters, the survivor function can be written as:

$$S\left(t\right) = S_0(t)^{\exp\left(-0.3899fin+0.2591race+0.5249wexp+0.3330prio\right)}\ , \tag{2.8}$$

where $S_0(t)$ is a vector of baseline probabilities estimated from the data set from the beginning of the observation period until the end of the 52nd week. The baseline survival probability, $S_0(t)$, is the probability measured when all risk factors are absent ($fin = 0, race = 0, wexp = 0$, and $prio = 0$) at time t. For example, the baseline survival probabilities for the first five weeks are $S_0(1), S_0(2), S_0(3), S_0(4), S_0(5) = 0.9984, 0.9968, 0.9951, 0.9935, 0.9919$.

Examples of the survival curves along with their 95% confidence interval estimated from the CPH model are shown in Figure 1. The grey line represents the baseline survival curve which is a vector of baseline survival probabilities, $S_0(t)$, measured when all risk factors are absent ($fin = 0, race = 0, wexp = 0, andprio = 0$) at time t. When other cases than the baseline are analyzed, the survival probability can be estimated based on this baseline and respective hazard ratios. For example, the survival probability of a prisoner group with $fin = 0, race = 1, wexp = 1$, and $prio = 0$ relative to the baseline group at any time t can be calculated from

$$S\left(t\right) = S_0(t)^{\exp\left(-0.3899(0)+0.2591(1)+0.5249(1)+0.3330(0)\right)} = S_0(t)^{\exp(0.784)}\ . \tag{2.9}$$

The baseline survival probability at the first week, $S_0(1)$, is 0.9984. If we want to assess the survival probability of the selected prisoner group with $fin = 0, race = 1, wexp = 1$, and$prio = 0$ in the first week, we can compute the survival probability of the first week as $S(1) = 0.9984^{exp(0.784)} = 0.9965$. By repeating the same steps, we can obtain survival probabilities for each week relative to the baseline. As a result, we have a vector of survival probabilities of the selected group shown as the blue line in Figure 1. ■

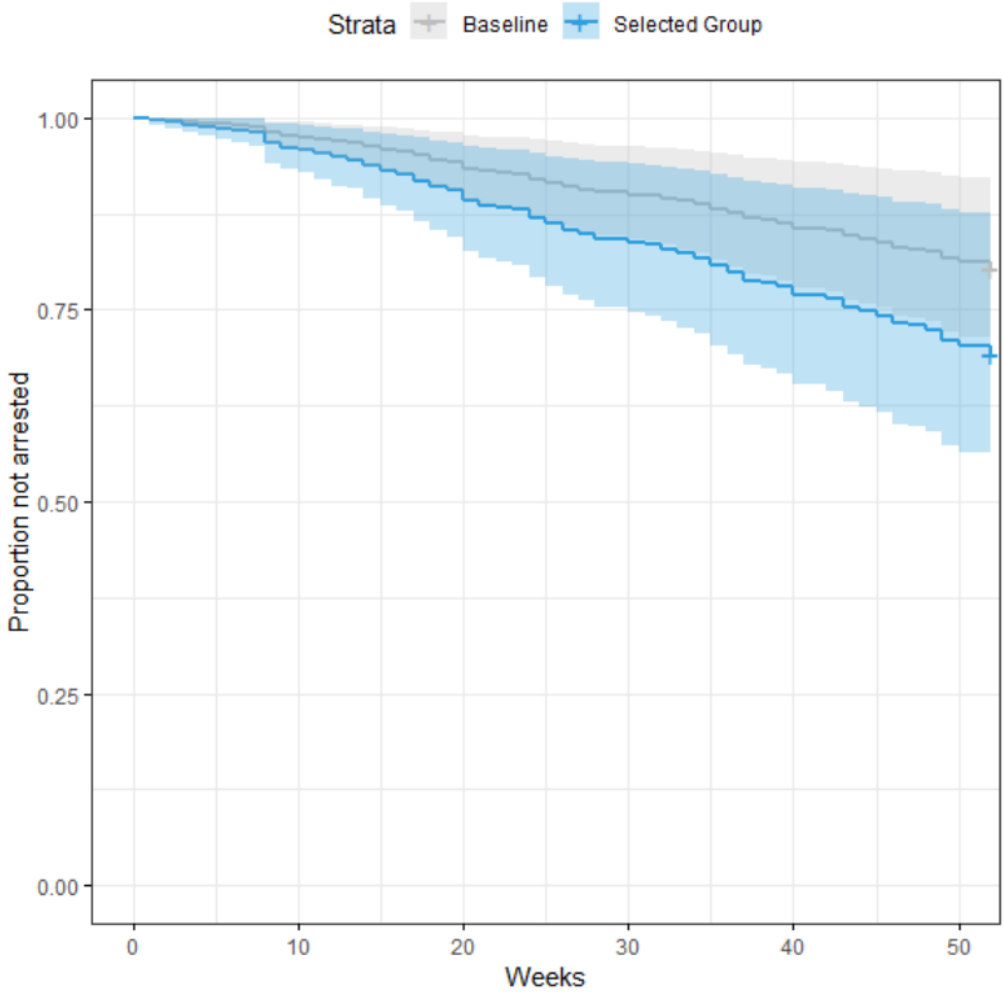


Figure 1: Survival curves along with their 95% confidence intervals from the CPH model reported in Table 1: baseline vs. selected group

2.1.2 Kaplan-Meier estimates

The Kaplan-Meier (K-M) estimator (Kaplan and Meier, 1958) is an alternative method of depicting the survival curve. It amounts simply to calculating the survival probability for each time interval t based on the event occurrences at that time. From the data, the survival probabilities are estimated as follows,

$$S(t) = \prod_{t_i \leq t} (1 - \frac{d_i}{n_i}) \,, \tag{2.10}$$

where n_i is the number of subjects at risk at the beginning of the time interval t_i and d_i is the number of subjects who have not survived during the time interval t_i.

Unlike the CPH model, K-M does not include any risk factors and parameter estimation in the model, which make the K-M estimate a non-parametric method. The K-M method is learned directly from the observed survival data without the assumption of an underlying probability distribution. The observed survival data means the sub-group in the survival data given by a combination of risk factors. When there are enough data records to learn from, the K-M estimates provide good predicted survival curve. However, there could be few data records for each combination of risk factors. When there are not enough data records to learn from, the K-M estimates provide poor quality of survival curve.

Example 2. In this example, I also used the Recidivism data set (Rossi et al., 1980) to demonstrate the K-M model. Four risk factors ($fin, race, wexp$, and $prio$ were selected and discretized in the same way as in Example 1. Similarly, I used the R *survival* package to create the K-M model. The result of the model is a set of 16 survival curves estimated from the data, each for one combination of risk factors, e.g., $fin = 0$, $race = 1$, $wexp = 1$,and $prio = 0$ shown in Figure 2.

■

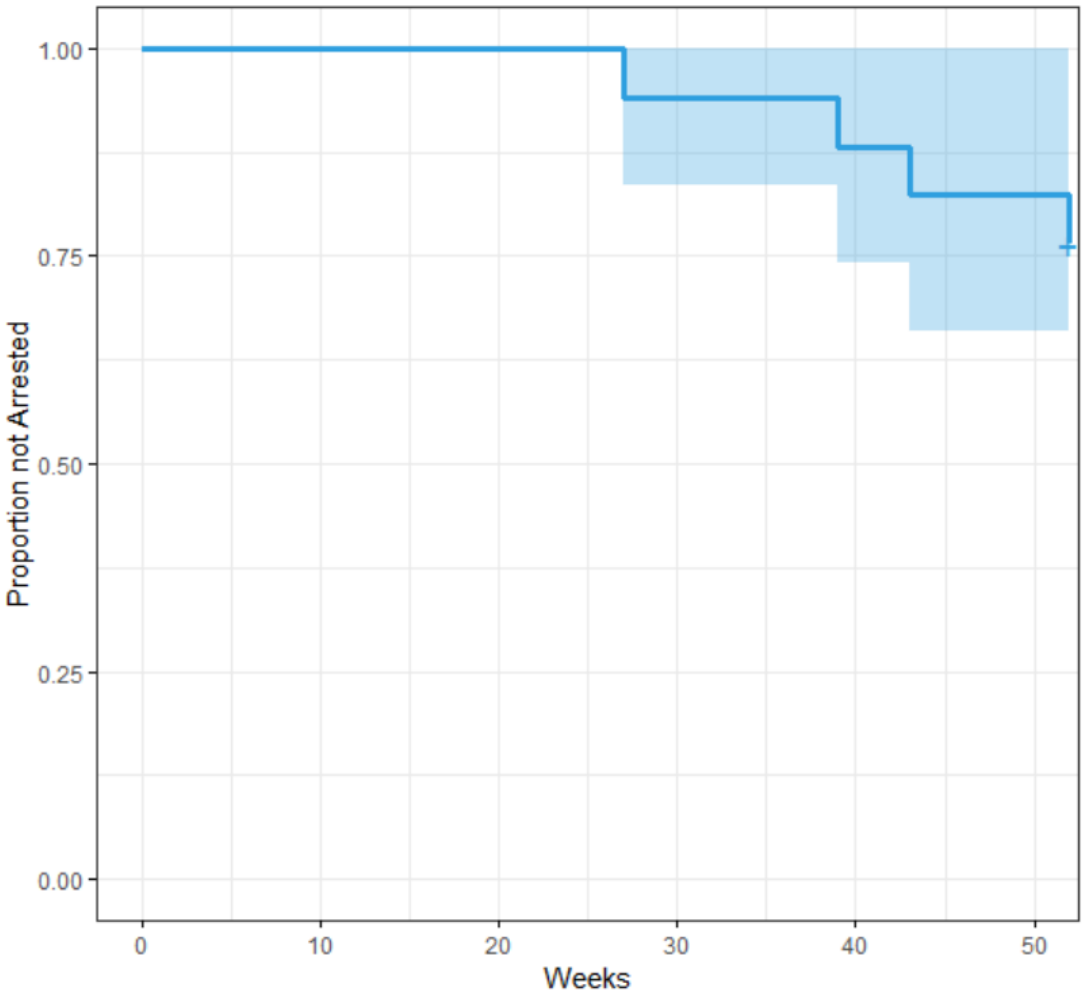


Figure 2: The survival curve along with its 95% confidence interval from the K-M model of the selected prisoner groups, i.e., when $fin = 0$, $race = 1$, $wexp = 1$,and $prio = 0$

2.2 Bayesian networks

Bayesian networks Pearl (1988) are probabilistic graphical models capable of modeling the joint probability distribution over a finite set of random variables. The structure of a BN is an acyclic directed graph in which each node corresponds to a single variable and directed arcs denote direct dependencies between pairs of variables. A conditional probability table (CPT) of a variable X contains probability distributions over the states of X for all combinations of states of X's parents. The joint probability distribution over all variables of the network can be calculated by taking the product of all prior and conditional probability distributions, i.e.,

$$\Pr(\mathbf{X}) = \Pr(X_1, \ldots, X_n) = \prod_{i=1}^{n} \Pr(X_i | Pa(X_i)) \ . \tag{2.11}$$

BNs have been used in numerous practical applications and because they are capable of deriving the posterior marginal probability distribution over a variable of interest, given values of other variables in the model, it is quite natural to apply them to survival analysis. BNs are compact and intuitive, while also being theoretically sound Husmeier et al. (2005). They can be based purely on literature or expert knowledge, can be learned from data, or a combination of the two. Calculation in BNs, which worst case NP-hard, is very efficient for most practical models known.

There are two general approaches to building Bayesian networks for the purpose of risk assessment. Researchers can implement static models that predict risk or survival at a snap-shot of time. For example, Kanwar et al. (2018) developed an application of Bayesian networks to survival analysis include risk assessment models for patient data with the left ventricular assist devices (LVADs) from the INTERMACS data set Kirklin et al. (2017). Bayesian networks estimate the risk of mortalit at specific points in time including 1, 3, and 12 months with high accuracy. A more complex approach uses dynamic Bayesian networks (DBNs). van Gerven et al. (2008) implemented a DBN for prognosis of patients that suffer from low-grade midgut carcinoid tumor. Instead of analyzing each time point separately, the DBN model calculates how the state of the patient changes over time under the influence of therapy choices. This allows for modelling temporal nature of medical problems throughout

the course of care, and provides detailed prognostic predictions. However, it requires significantly more effort more during model construction, i.e., require expertise to define causal structure and temporal interaction, large amount of data, and is generally time-consuming. In the scope of this dissertation, I will focus on the first approach which is a discrete Bayesian network to predict outcome at a snap-shot of time.

3.0 Bayesian network interpretation of Cox's proportional hazard model

Cox's proportional hazards (CPH) model is quite likely the most popular modeling technique in survival analysis. While the CPH model is able to represent a relationship between a collection of risks and their common effect, Bayesian networks have become an attractive alternative with an increased modeling power and far broader applications. However, building Bayesian networks based purely on expert knowledge can be a time-consuming and costly task. Luckily, many CPH models can be found in the literature. They are typically published as a set of numerical coefficients along with their significance levels. No original data are usually available. To use the knowledge encoded in these CPH models, an interpretation of the CPH parameters is needed. In this chapter, I provide such a method of encoding knowledge from existing CPH models in the process of knowledge engineering for Bayesian networks (Section 3.1) along with its empirical evaluation (Section 3.2). Section 3.3 provides an example of the use of BN-Cox to risk assessment. Finally, Section 3.4 discusses two approaches for simplifying the BN-Cox model for the sake of representational and computational efficiency

3.1 Definition

As I mentioned earlier, the process of building Bayesian networks can take a significant effort, especially when little or no data are available. In this section, I discuss how to use parameters from existing CPH models to create Bayesian networks (we will call it the BN-Cox model). This approach is especially useful when very little or no data are available. I assume that the CPH model's assumptions are not violated and the risk factors or random variables $\mathbf{X}$ are time-independent discrete/binary variables (Kraisangka and Druzdzel, 2018).

To create a Bayesian network, I create its structure by designating the random variables representing risk factors as parents ($\mathbf{X}$) of the outcome or survival node (S). The number of states of each random variable is the same as in the CPH model.

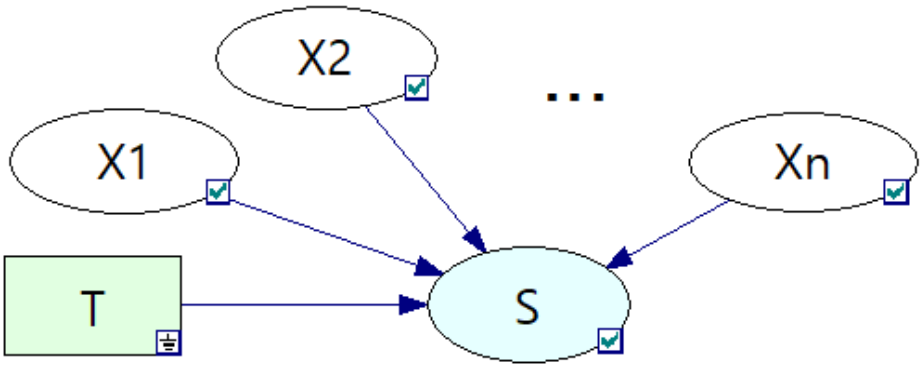


Figure 3: A structure of BN-Cox model representing interactions among variables

Unlike the CPH model, static Bayesian networks capture a snapshot of a system at a certain time. I need, thus, to represent time explicitly by adding an indexing variable (T) for *time*, capturing each discrete point in time that is of interest, e.g., every day, every two weeks, etc. This time variable can be omitted if we are interested in the prediction at one point in time, e.g., at one year. CPH models represent their relationship between individual risk factors to the outcome in the form of a multiple linear regression (in the logarithmic scale). Thus, the structure of the BN-Cox model can be interpreted as the structure of a Na ive Bayes model. Figure 4 show an example of such a model, showing the relationship between risk factors ($\mathbf{X}$), the time variable (T), and the survival node (S).

In the next step, I create the conditional probability table for the *survival* node (S). Recall that we can obtain the survival probabilities from Equation 2.7 in the CPH model. For each time snapshot captured by the variable T, we assess a set of survival probabilities, $S(t)$ from the CPH model. A set of survival probabilities here means that we configure the hazard ratio γ according to the combination of the parent states. γ is equal to hazard ratio of the conditioning case $\mathbf{X_i}$ to the baseline case $\mathbf{X_b}$, i.e., case in which all risk variables are *absent*, i.e.,

$$\gamma = \frac{exp(\beta' \mathbf{X_i})}{exp(\beta' \mathbf{X_b})} = \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_n X_{in} \ . \tag{3.1}$$

Equation 3.1 allows us to assess the survival probabilities directly from the parameters of

the CPH model. First, I configure all risk factor cases in Equation 3.1 to find all hazard ratio values. Then I obtain the baseline survival probability at the first point in time from the CPH model ($S_0(t = 1)$) and use Equation 2.7 to find the survival probability. The survival probability calculated for each combination of risk factors corresponds to the conditional probability of survival. Hence, the conditional probability to be encoded in the CPT can be estimated by

$$Pr(s \mid \mathbf{X}, T = t) = S_0(t)^{exp(\beta' \mathbf{X})} , \tag{3.2}$$

where s corresponds to the state *survived* in the survival node S, $\mathbf{X}$ are risk factors, and T is the time point. This allowed us to reproduce fully the CPH model by means of a Bayesian network.

Example 3. For this example, I will use the CPH model from Example 1 as a source to create a BN-Cox model. I used GeNIe[1] to implement its structure, and obtained survival probabilities. To create a structure of the BN-Cox model, each of the risk factors and the survival variable are converted into a random variable (*fin*, *race*, *wexp*, *prio*, and *arrest*). These random variables representing risk factors are parents of the survival node, *arrest*. For the purpose of simplicity, I reduced the number of states for the time variable *week* from 52 to 13, which amounts to analyzing the system at 4-week steps. Other random variables (risk factors) have the same states as in the CPH model from Example 1. The resulting structure of the Bayesian network are shown in Figure 4.

For each time snapshot captured in the variable *week*, a set of survival probabilities, $S(t)$, can be assessed from the CPH model, in this case, at 4-week steps. A set of survival probabilities here means that the hazard ratio γ has to be configured according to the combination of the parent states. γ is equal to the ratio of hazard of the conditioning case X_i to the baseline case X_b. Selected probabilities of survival for all combinations of states of the risk variables are shown in Table 2. ■

[1] Available at http://www.bayesfusion.com/.

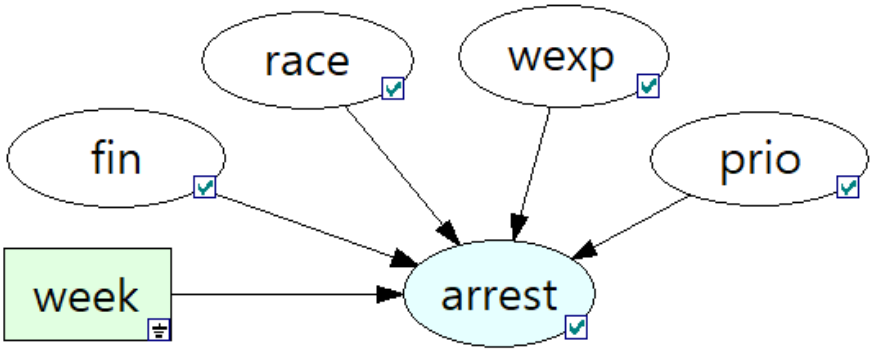


Figure 4: The structure of a BN-Cox model for the CPH model from Example 1.

3.2 Empirical evaluation

In this section, I provide an empirical evaluation of the BN-Cox model by comparing its predictive precision to the baseline survival analysis models like the CPH model and the Kaplan-Meier (K-M) estimator (Kaplan and Meier, 1958) and to Bayesian networks learned from data. I used the Recidivism data set as shown in the previous examples. I will explain how to build the models and show the result of the predictive comparison in the following sections.

3.2.1 Model construction

I constructed seven models for the purpose of the empirical evaluation. I used the BN-Cox model constructed in Example 3 I used the K-M model from Example 2and the CPH model from Example 1 as representatives from survival analysis approach. The K-M and the CPH models were created by uing the R programming environment with the *Survival* library Fox (2002).

For Bayesian network approach, I created four Bayesian networks including a Bayesian network model learning from the data set (BN-Learn) using Expectation-Maximization (EM) algorithm (Dempster et al., 1977; Lauritzen, 1995), a Bayesian network model using Naïve

Table 2: Conditional probabilities of survival for all cases at each snapshot of time. γ is calculated from Equation 3.1 and $S(1)$, $S(2)$, ..., $S(13)$ are calculated from Equation 3.2 at 4-week steps. s is the survival variable *arrest*.

$Pr(s \mid X_i)$	γ	$S(1)$	$S(2)$	...	$S(12)$	$S(13)$
$\mathbf{Pr}(s \mid f=0, r=0, w=0, p=0)$	0.0000	0.998	0.992	...	0.830	0.804
$\mathbf{Pr}(s \mid f=0, r=0, w=0, p=1)$	0.3330	0.998	0.989	...	0.771	0.738
$\mathbf{Pr}(s \mid f=0, r=0, w=1, p=0)$	0.5249	0.997	0.986	...	0.729	0.692
$\mathbf{Pr}(s \mid f=0, r=0, w=1, p=1)$	0.8579	0.996	0.981	...	0.644	0.599
$\mathbf{Pr}(s \mid f=0, r=1, w=0, p=0)$	0.2591	0.998	0.990	...	0.785	0.754
$\mathbf{Pr}(s \mid f=0, r=1, w=0, p=1)$	0.5921	0.997	0.986	...	0.714	0.675
$\mathbf{Pr}(s \mid f=0, r=1, w=1, p=0)$	0.7840	0.997	0.983	...	0.665	0.621
$\mathbf{Pr}(s \mid f=0, r=1, w=1, p=1)$	1.1117	0.995	0.976	...	0.565	0.514
$\mathbf{Pr}(s \mid f=1, r=0, w=0, p=0)$	-0.3899	0.999	0.995	...	0.881	0.863
$\mathbf{Pr}(s \mid f=1, r=0, w=0, p=1)$	-0.0569	0.998	0.992	...	0.838	0.814
$\mathbf{Pr}(s \mid f=1, r=0, w=1, p=0)$	0.1350	0.998	0.991	...	0.808	0.779
$\mathbf{Pr}(s \mid f=1, r=0, w=1, p=1)$	0.4680	0.997	0.987	...	0.742	0.706
$\mathbf{Pr}(s \mid f=1, r=1, w=0, p=0)$	-0.1308	0.999	0.993	...	0.849	0.826
$\mathbf{Pr}(s \mid f=1, r=1, w=0, p=1)$	0.2022	0.998	0.990	...	0.796	0.766
$\mathbf{Pr}(s \mid f=1, r=1, w=1, p=0)$	0.3941	0.998	0.988	...	0.758	0.724
$\mathbf{Pr}(s \mid f=1, r=1, w=1, p=1)$	0.7271	0.997	0.983	...	0.680	0.637

Bayes (BN-NB) learning algorithm, a BN model with Tree Augmented Naïve Bayes (BN-TAN) learning algorithm (Friedman et al., 1997), and a BN model with Noisy-Max (BN-NoisyMax) gates (Nowak and Druzdzel, 2014). For the BN-Learn model, I built the model in GeNIe using the same structure as in the BN-Cox model (Figure 4). The BN-Learn model was learned only the numerical parameters from data using the EM algorithm. The BN-NB model and BN-TAN were learned both structure and parameters directly from data, while BN-NoisyMax was learned using the method published in Nowak and Druzdzel (2014).

In summary, there are seven models (K-M, CPH, BN-Cox, BN-Learn, BN-TAN, BN-NB and BN-NoisyMax) with four risk factors: *fin*, *race*, *wexp* and *prio*. These four risk factors are binary variables resulting in $2^4 = 16$ combinations of risk factors. We compare the prediction accuracy of each model in the following section.

3.2.2 Prediction comparison

With four binary risk factors, there are 16 combinations of risk factors. I plotted the distribution of the number of records corresponding to these 16 cases in Figure 5, sorted in descending order. For the purpose of comparison, I selected four cases as candidates, including one with the highest number of records (102 records), one with a medium-to-high number of records (61 records), one with a medium-to-small number of records (9 records), and one with a small number of records (2 records). The dark grey color indicates the selected cases in Figure 5.

Figure 6 shows the survival probabilities predicted by each of the seven models: K-M model (round-dotted line), CPH model (square-dotted line), BN-Cox (diamond), BN-Learn (triangle), BN-TAN (red dash), BN-NB (dark blue dash), and BN-NoisyMax (orange dash). We observe an almost perfect match between the CPH and the BN-Cox model in all 16 cases. Both BN-TAN and BN-NB models are close for every case, while the BN-NoisyMax model falls in-between. The K-M and BN-Learn model are also close, although they both depart from the CPH model significantly as the number of records gets smaller (Figure 6c and Figure 6d).

BN-Cox, BN-Learn, BN-TAN, BN-NB, and BN-NoisyMax models are simplified and

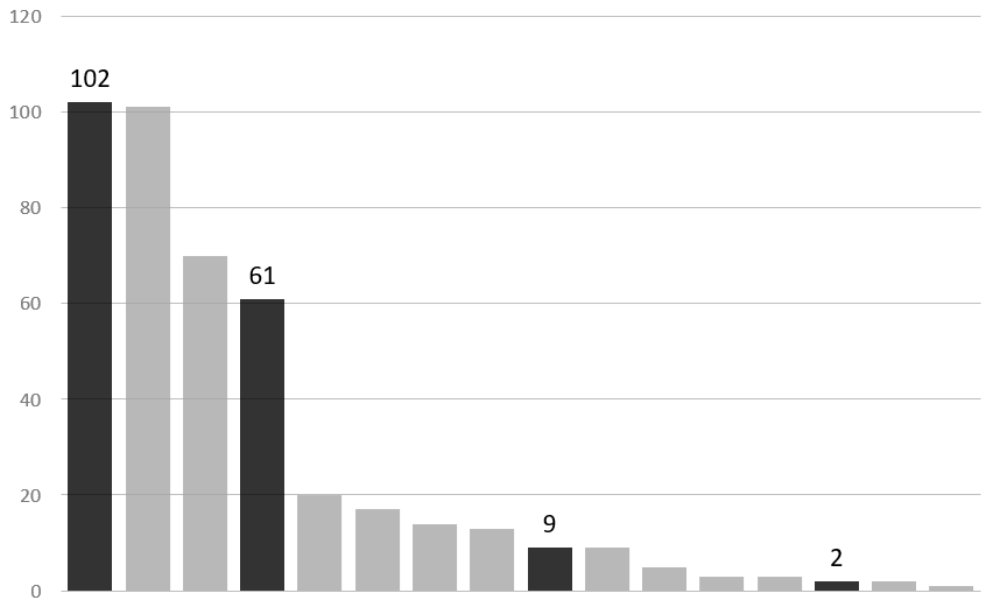
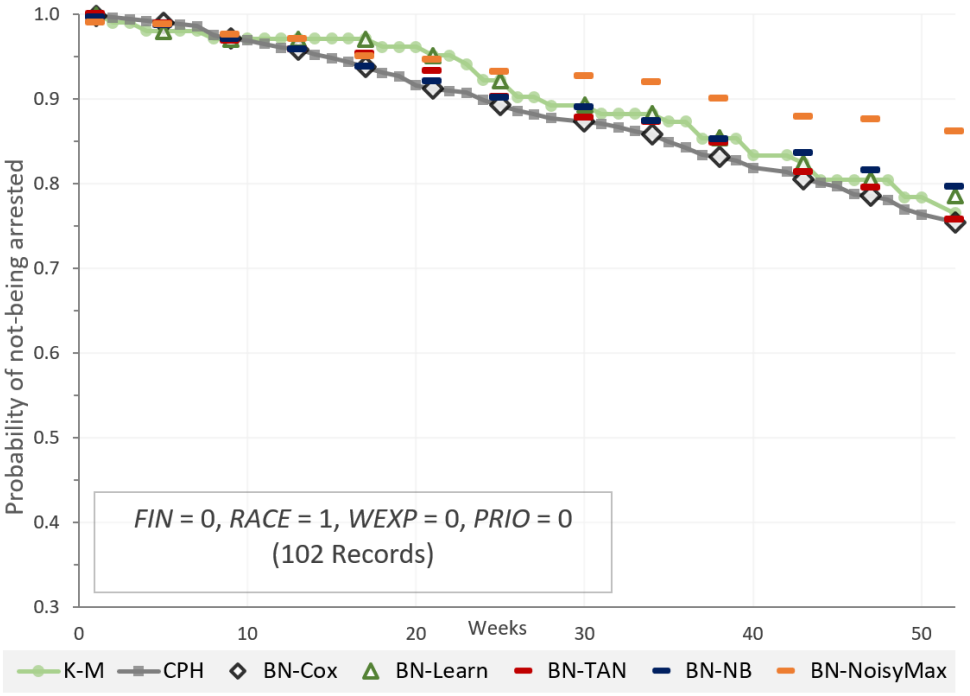


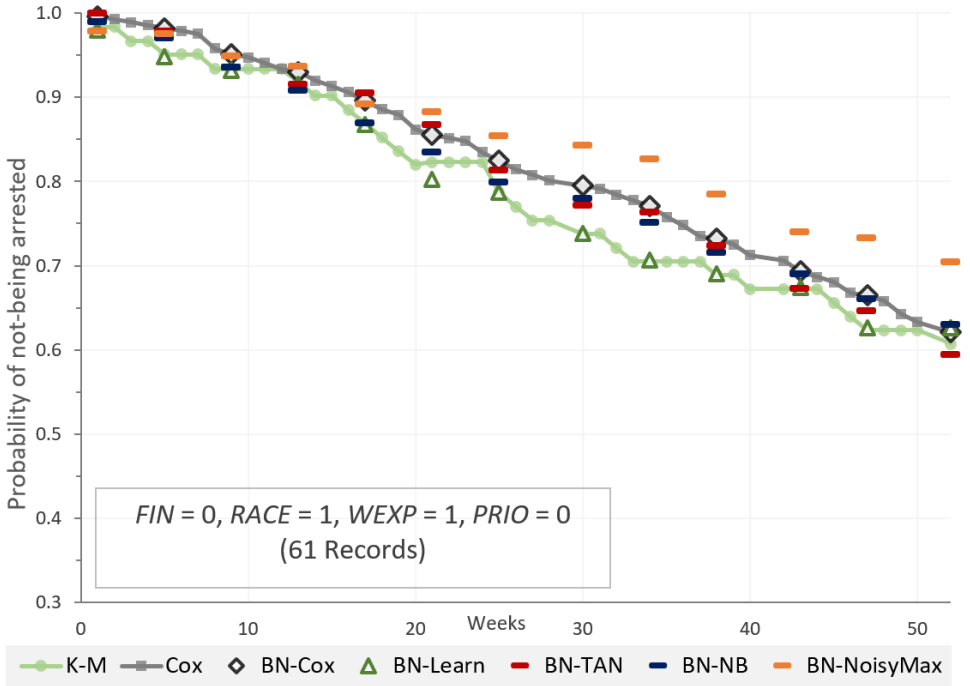
Figure 5: Distribution of the number of records in the Recidivism data set with four risk factors for each of the 16 combinations of risk factors (sorted in descending order).

produce 13 survival probabilities for each case while the K-M and CPH model produced 52 survival probabilities. We found that when we have enough data to learn, e.g., more than a hundred records, there is a remarkable agreement among all seven models. However, when there are fewer data points, we found that the curves produced by the K-M estimate and the Bayesian network learned from data (BN-Learn), while in agreement with one another, depart from the CPH model significantly. The BN-Cox model and the CPH model, which again agree perfectly, produce smoother curves. We also observed agreement between the BN-TAN and the BN-NB models producing smoother curves for cases with few or no data records. The BN-NoisyMax model predicts probabilities in-between but not so similar to the remaining models.

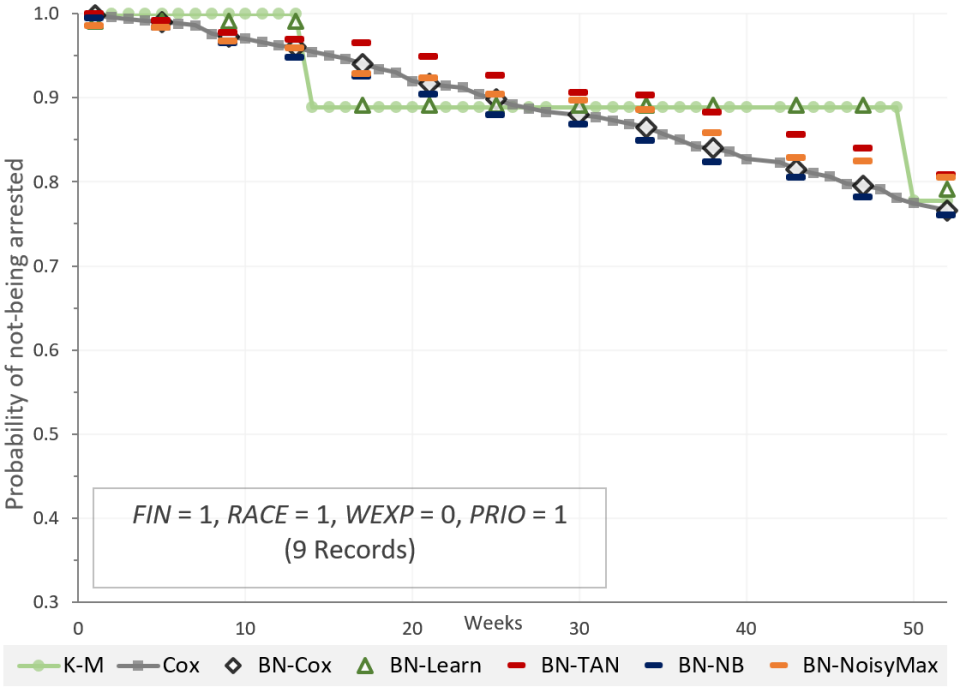
With these complete Recidivism models, there are 512 combinations of risk factors. I found that the distribution of the cases in terms of the number of records is extremely skewed. As shown in Figure 7, the case best represented in the data has only 32 records, while more than 70 percent of cases (392 cases of the total of 512 cases) have zero records.



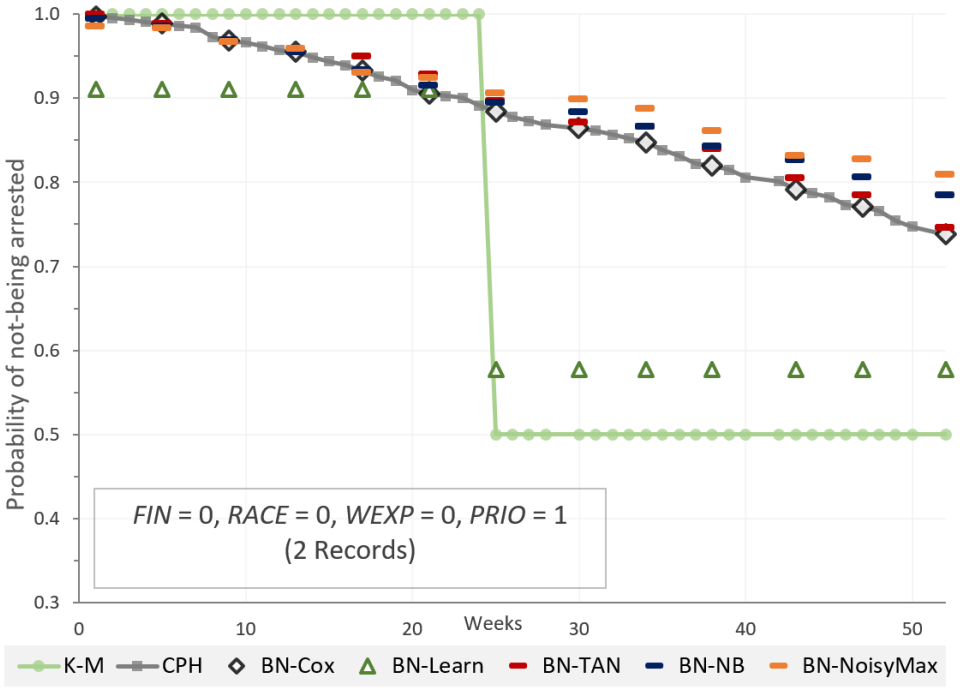
(a) Predicted survival curves of the selected group with 102 records



(b) Predicted survivals of the selected group with 61 records



(c) Predicted survival curves of the selected group with 9 records



(d) Predicted survival curves of the selected group with 2 records

Figure 6: Comparison of the predicted survival curves in the four-risk-factor models

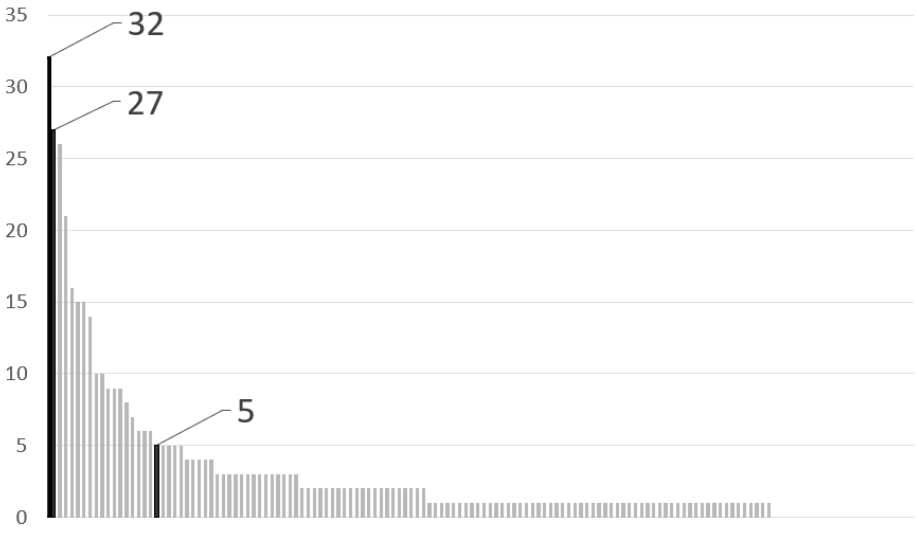
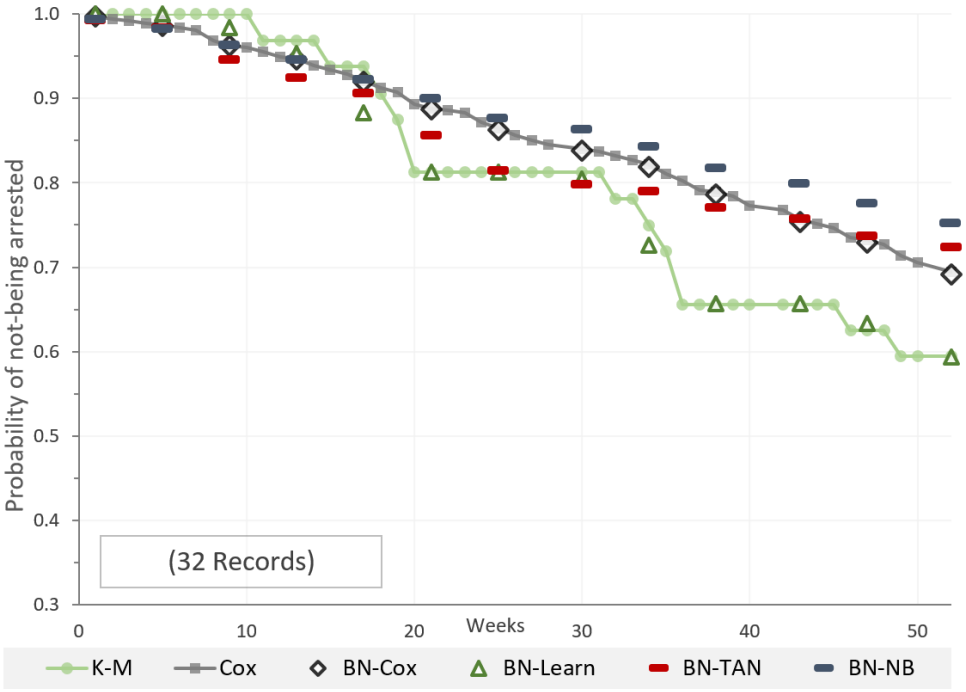


Figure 7: Distribution of the number of records in the Recidivism data set with all risk factors for each of the 512 combinations of risk factors (sorted in descending order).

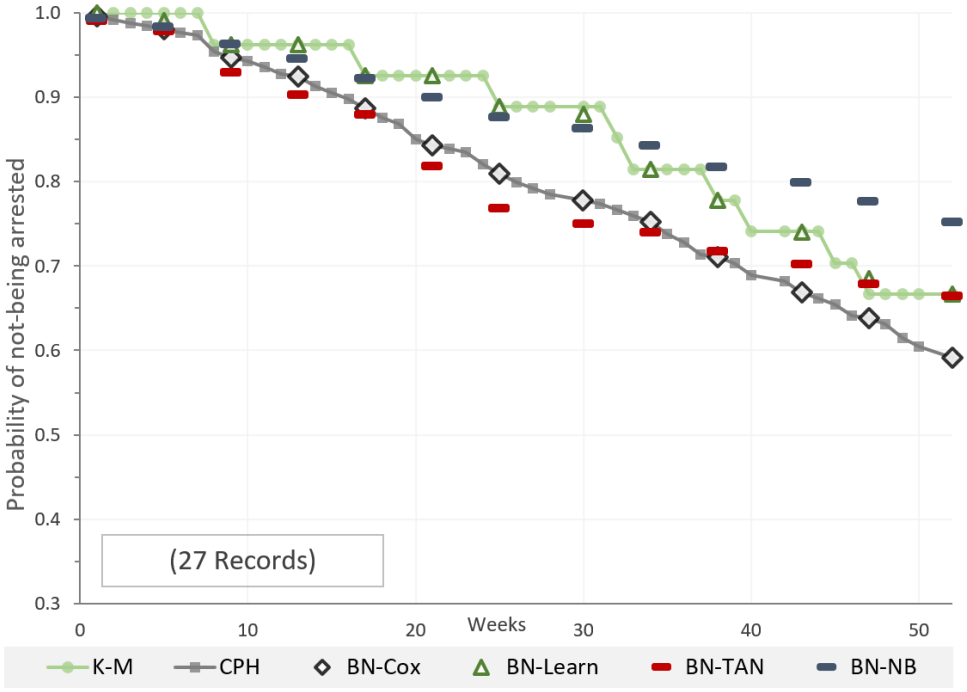
Hence, I selected four cases, with 32, 27, 5, and 0 records respectively, for the purpose of the comparison.

The results were similar to those of the simplified models. The survival probabilities predicted by the BN-Cox model were identical to those of the CPH model. The BN-Learn, the K-M model, the BN-TAN, and the BN-NB models produced similar trends, but the BN-Learn had an overall lower predicted survival probability. We can see larger differences in the predicted probability when there are few data records to learn from. The K-M model, BN-Learn, and BN-TAN produce different results only when the number of data records is small or zero. In this case, the CPH and the BN-Cox models agree perfectly.

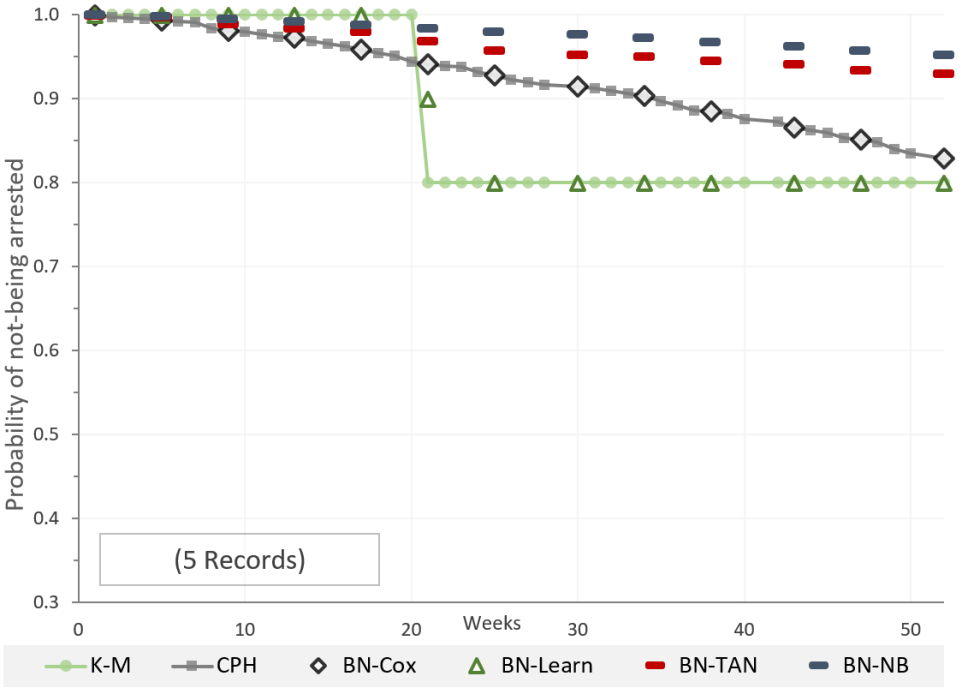
In addition to the simplified, four-risk-factor model, I also created a complete Recidivism model with all eight risk factors using the same techniques for the four-risk-factor model. The complete Recidivism model consists of seven binary and one categorical variable (see all variable details in Example 1). However, I only created six models: K-M, CPH, BN-Cox, BN-Learn, BN-TAN, and BN-NB, since the Noisy-Max algorithm cannot handle non-binary variables.



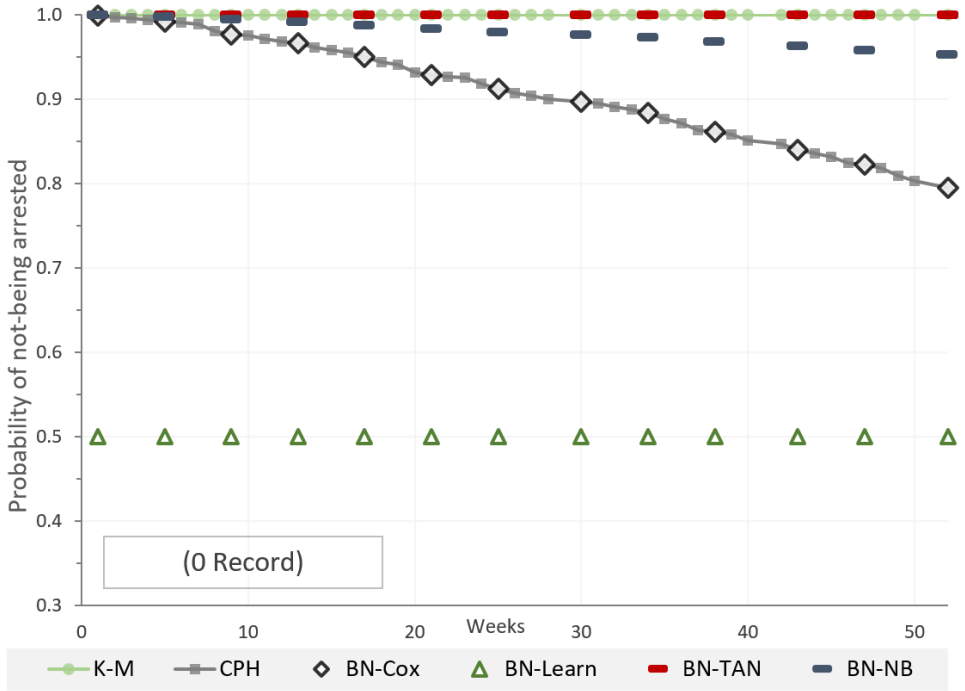
(a) Predicted survival curves of the selected group with 32 records



(b) Predicted survivals of the selected group with 27 records



(c) Predicted survival curves of the selected group with 5 records



(d) Predicted survival curves of the selected group with no record

Figure 8: Comparison of the predicted survival curves in the all-risk-factor models

Table 3: Performance of Bayesian network models with four risk factors and all risk factors

Performance	BN-Cox	BN-Learn	BN-TAN	BN-NB	NoisyMax
(Four-risk-factors models)					
Accuracy (ACC)	0.8759	0.8769	0.8769	0.8761	0.8769
Area under ROC (AUC)	0.7605	0.7609	0.7536	0.7514	0.7421
(All-risk-factors models)					
Accuracy (ACC)	0.8797	0.8803	0.8764	0.8748	0.8769
Area under ROC (AUC)	0.8322	0.8345	0.7926	0.7635	0.5646

I also compared the accuracy (ACC) and the area under the receiver operating characteristic (ROC) curve (AUC) for the BN-Cox, the BN-Learn, the BN-TAN, the BN-NB, and the BN-NoisyMax models (for four-variable models and all-variables models) using 10-fold cross validation (Table 3). For the four-risk-factor models, each model produced very similar accuracy (ACC) and the area under ROC (AUC). Both BN-Cox and BN-Learn performed similarly. BN-TAN and BN-NoisyMax are unable to correctly predict the re-arrest. We also observed similar performance for all-variables model. BN-Learn offered the best accuracy among all methods for the four-variables and all-variables models, while BN-NoisyMax was the least accurate. However, the differences in accuracy among the models are not significant (McNemar's test $p > 0.05$).

In summary, our results show that when we do not have any data to learn from but only have an existing model, i.e., the CPH model, we can create a BN-Cox model to get similar performance. The BN-Cox model will relax the assumption of the multiplicative character of interactions between the risk factors and the survival variable.

3.3 Application of BN-Cox to risk assessment

In this section, I provide an example of the use of BN-Cox in risk assessment for pulmonary arterial hypertension. Pulmonary arterial hypertension (PAH) is a chronic and life-changing disease originating from an increase in pulmonary vascular resistance, and leading to high blood pressure in the lung. One of the most widely used tools in prognosis and management of PAH is the REVEAL risk score calculator Benza et al. (2010), which assesses the risk of death of a PAH patient based on various risk factors. With no access to the REVEAL Registry data, I replaced the CPH model by a BN-Cox model constructed from the CPH parameters reported in Benza et al. (2010).

The core of the REVEAL risk score calculator by Benza et al. (2012) was based on the multivariate CPH model. The model is comprised of 19 demographic, functional, laboratory, and hemodynamic parameters (reproduced from the original paper in Table 4. The risk factors X includes PAH associated with portal hypertension (APAH-PoPH), PAH associated with connective tissue disease (APAH-CTD), family history of PAH (FPAH), being male aged over 60 years, having renal insufficiency, modified New York Heart Association (NYHA)/World Health Organization (WHO) functional class I, III, and IV, systolic blood pressure (SBP), heart rate, 6-minute walking distance (6MWD), brain natriuretic peptide (BNP), presence of pericardial effusion on echocardiogram, percentage predicted diffusing capacity of lung for carbon monoxide (Dlco), mean right atrial pressure (mRAP) and pulmonary vascular resistance (PVR). Most of the risk factors were associated with increasing mortality rate (indicated by positive sign in β), while only four factors were associated with increased one-year survival (indicated by negative sign in β). The baseline probability of survival was reported as $S_0(1) = 0.9698$.

By following the method outlined in Section 3.1, I created a BN-Cox model shown in Figure 9. In this case, we omitted the *time* variable, as the purpose of the REVEAL risk score calculator is to capture the risk at one point in time (one year). This by itself offers no advantages over a CPH model-based calculator but it was the first step toward a better calculator that relaxes some of the CPH assumptions and is capable of representing a generalized structure of interactions between risk factors and the survival variables.

Table 4: A list of 19 binary risk factors, their corresponding coefficients β, hazard ratios $exp(\beta)$ and p-values reported for the CPH model from Benza et al. (2010).

Risk factors X_i	β	$exp(\beta)$	p-value
APAH-CTD	0.7737	1.59	<0.001
FPAH	1.2801	3.60	<0.001
APAH-PoPH	0.4624	2.17	0.012
Male aged >60 years	0.7779	2.18	<0.001
Renal insufficiency	0.6422	1.90	<0.001
NYHA/WHO FC I	-0.8740	0.42	0.039
NYHA/WHO FC III	0.3454	1.41	0.008
NYHA/WHO FC IV	1.1402	3.13	<0.001
SBP <110 mmHg	0.5128	1.67	<0.001
Heart Rate >92bpm	0.3322	1.39	0.005
6MWD ≥440 m	-0.5455	0.58	0.006
6MWD <165 m	0.5210	1.68	<0.001
BNP <50 pg/ML	-0.6922	0.50	0.003
BNP >180 pg/ML	0.6791	1.97	<0.001
Pericardial effusion	0.3014	1.35	0.014
% DLCO ≥80%	-0.5317	0.59	0.031
% DLCO ≤32%	0.3756	1.46	0.018
mRAP >20 mmHg	0.5816	1.79	0.043
PVR >32 Wood units	1.4062	4.08	<0.001

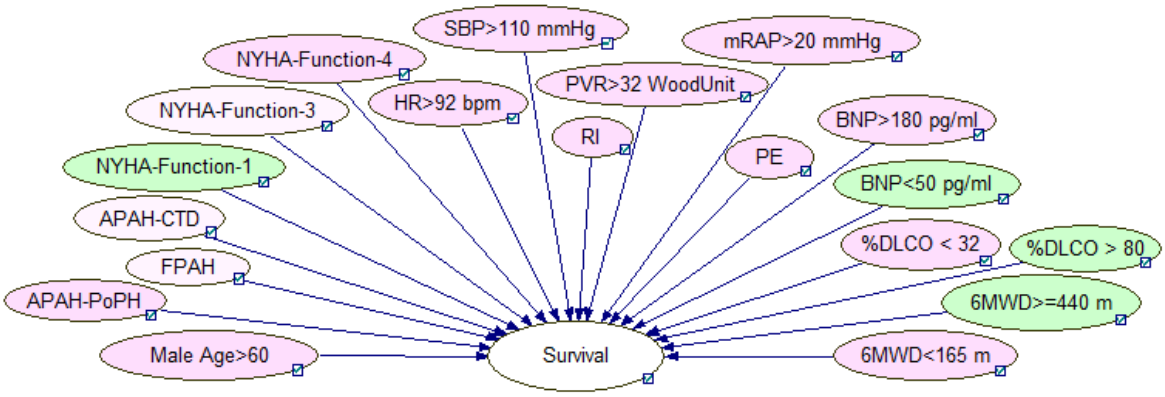


Figure 9: A BN-Cox model representing the interaction among variables for the CPH model in the REVEAL risk score calculator.

I applied the same approach from Benza et al. (2012) to create a simplified risk score calculator. Equation 3.2 captures the survival probabilities s given the states of risk factors. We can extract a hidden hazard ratio of each variable by configuring states of other risk factors to be absent. For example, the hazard ratio of a risk factor x_j can be estimated from

$$\gamma = \frac{\log(Pr(s \mid \bar{x}_1, \ldots, \bar{x}_{j-1}, \mathbf{x_j}, \bar{x}_{j+1}, \ldots, \bar{x}_n))}{\log(Pr(s \mid \bar{x}_1, \ldots, \bar{x}_{j-1}, \bar{\mathbf{x}}_\mathbf{j}, \bar{x}_{j+1}, \ldots, \bar{x}_n))} \,. \tag{3.3}$$

The term $\log(Pr(s \mid \bar{x}_1, \ldots, \bar{x}_{j-1}, \bar{\mathbf{x}}_\mathbf{j}, \bar{x}_{j+1}, \ldots, \bar{x}_n))$ is similar to the baseline survival probability in the CPH model ($S_0(1) = 0.9698$). Hence, with this equation, we can track back all hazard ratios. Then, we use the same criteria as the original REVEAL risk score calculator to convert the hazard rate to a score. Score of 2, for example, indicates at least two-fold increase in risk of mortality compared to the baseline risk.

Figure 10 shows a screen shot of the graphical user interface (GUI) of our prototype of the BN-Cox risk score calculator. The left-hand side pane allows for entering risk factors for a given patient. The right-hand side pane shows the calculated score and survival probabilities. Currently, the numerical risks produced by the BN-Cox calculator are identical to those of the original CPH-based REVEAL risk score calculator (Benza et al., 2012). However, the BN-Cox model makes CPH's assumptions explicit and will allow to relax them in the future.

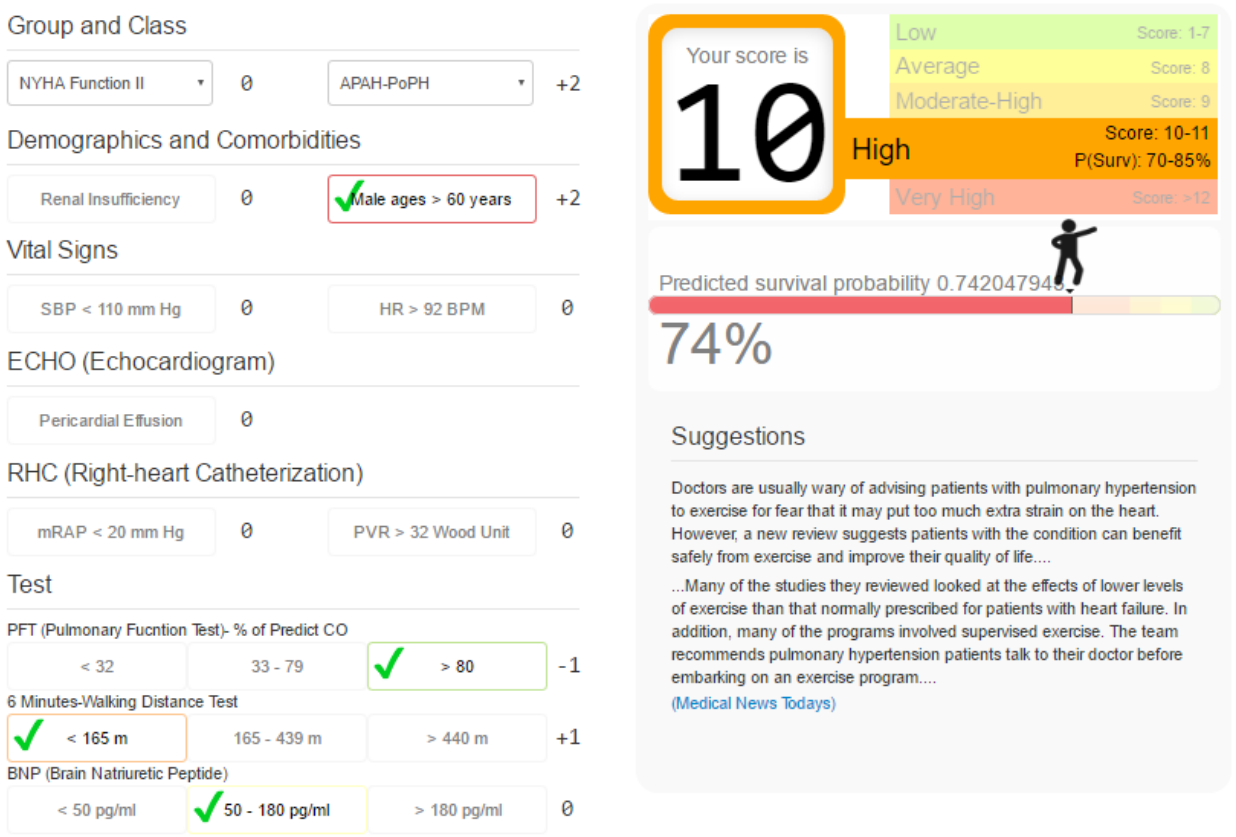


Figure 10: A prototype GUI for our BN-Cox risk score calculator for a 1-year PAH prognosis model. The left-hand pane allows for entering risk factors for a given patient case. The right-hand pane shows the calculated score and the survival probability.

One immediate advantage of the BN-Cox representation is that BNs make it possible to refine the parameters with additional data records.

3.4 Making BN-Cox tractable

One of the challenges to applying the BN-Cox model is an exponential growth of the conditional probability tables (CPT) corresponding to the survival variables, as the number of risk factors increases (Kraisangka and Druzdzel, 2016, 2018). When the number of risk factors is high, this table becomes intractable. I evaluated two approaches to mitigate this problem: (1) decomposition of the underlying Bayesian network known as parent divorcing, and (2) simplifying the network structure by removing least influential risk factors.

3.4.1 BN-Cox decomposition

In Bayesian networks, one way of reducing the complexity when the CPT of a node becomes too complex is through decomposition. This process can lead to substantial efficiency improvements in Bayesian updating (Zagorecki et al., 2006). In case of the noisy-OR gates (Díez and Druzdzel, 2006), for example, the combination function can be decomposed into a series of binary OR functions. For example, the $OR(X_1, \ldots, X_n)$ function is equivalent to $OR(X_1, OR(X_2, OR(\ldots OR(X_{n1}, X_n)\ldots)))$. Other functions, such as *AND*, *MIN*, and *MAX* can be decomposed similarly.

Decomposition of the CPH model amounts to finding a function f that is capable of expressing the survival function $S(t)$ in the following way:

$$S\left(t\right) = f\left(S_1\left(t\right)^{e^{(\beta_1 X_1 + \beta_2 X_2)}}, S_2\left(t\right)^{e^{(\beta_3 X_3 + \beta_3 X_3)}}\right) . \tag{3.4}$$

However, the survivor function describes an interaction between states of risk factors (PRESENT and ABSENT) and the probability of survival. This is different from the OR function, which describes interaction between states of variables. The following theorem,

with an elegant proof offered by Jirka Vomlel, states that there is no universal decomposition function for the BN-Cox model.

Theorem 1. There exists no universal decomposition function for parent-divorcing a BN-Cox model.

Proof. By contradiction for the simplest case with two binary risk factors X and Y, parents of the survival node S. The probability of survival is in this case expressed by the following function:

$$P(S=1) = S_0^{exp(\beta_X X + \beta_Y Y)} \ . \tag{3.5}$$

We will attempt decomposition of the survival function into $P(A_X = 1) = S_0^{exp(\beta_X X)}$, the survival probability considering X as the only risk factor, and $P(A_Y = 1) = S_0^{exp(\beta_Y Y)}$, the survival probability considering Y as the only risk factor. Decomposition using parent divorcing requires two auxiliary nodes, A_X and A_Y, parents of S, with the conditional probabilities $P(S|A_X, A_Y)$

$$\begin{aligned} c_1 &= P(S=1|A_X=0, A_Y=0) \\ c_2 &= P(S=1|A_X=0, A_Y=1) \\ c_3 &= P(S=1|A_X=1, A_Y=0) \\ c_4 &= P(S=1|A_X=1, A_Y=1) \ . \end{aligned}$$

In order to decompose the BN-Cox model using the parent divorcing method, the following must hold for all values of $X \in [0,1]$ and $Y \in [0,1]$

$$\begin{aligned} S_0^{exp(\beta_X X + \beta_Y Y)} &= c_1(1 - S_0^{exp(\beta_X X)})(1 - S_0^{exp(\beta_Y Y)}) + c_2(1 - S_0^{exp(\beta_X X)})S_0^{exp(\beta_Y Y)} \\ &\quad + c_3 S_0^{exp(\beta_X X)}(1 - S_0^{exp(\beta_Y Y)}) + c_4 S_0^{exp(\beta_X X)} S_0^{exp(\beta_Y Y)} \ . \end{aligned} \tag{3.6}$$

By substituting $(X, Y) = (0, 0)$ we get

$$S_0 = c_1(1 - S_0)(1 - S_0) + c_2(1 - S_0)S_0 + c_3 S_0(1 - S_0) + c_4 S_0 S_0 \ . \tag{3.7}$$

For the function to be universal, i.e., independent of the actual values of S_0, β_X, and β_Y, it must hold that $c_1 = 0$, $c_2 + c_3 = 1$, and $c_4 = 1$. If we substitute $c_1 = 0$ and $c_4 = 1$ into Equation 3.6 for $(X, Y) = (0, 1)$, we get:

$$\begin{aligned} S_0^{exp(\beta_Y)} &= c_2(1 - S_0)S_0^{exp(\beta_Y)} + c_3 S_0(1 - S_0^{exp(\beta_Y)}) + S_0 S_0^{exp(\beta_Y)} \\ &= c_2 S_0^{exp(\beta_Y)} - c_2 S_0^{1+exp(\beta_Y)} + c_3 S_0 - c_3 S_0^{1+exp(\beta_Y)} + S_0^{1+exp(\beta_Y)} \\ &= (1 - c_2 - c3)S_0^{1+exp(\beta_Y)} + c_2 S_0^{exp(\beta_Y)} + c_3 S_0 \ . \end{aligned}$$

From Equation 3.7, we know that $c_2 + c_3 = 1$ therefore $(1 - c_2 - c3) = 0$. Hence, we get:

$$S_0^{exp(\beta_Y)} = c_2 S_0^{exp(\beta_Y)} + c_3 S_0 \ . \tag{3.8}$$

For the function to be universal, it requires $c_2 = 1$ and $c_3 = 0$. If we substitute $c_1 = 0$ and $c_4 = 1$ into Equation 3.6 for $(X, Y) = (1, 0)$, we get:

$$\begin{aligned} S_0^{exp(\beta_X)} &= c_2(1 - S_0^{exp(\beta_X)})S_0 + c_3 S_0^{exp(\beta_X)}(1 - S_0) + S_0^{exp(\beta_X)} S_0 \\ &= c_2 S_0 - c_2 S_0^{1+exp(\beta_X)} + c_3 S_0^{exp(\beta_X)} - c_3 S_0^{1+exp(\beta_X)} + S_0^{1+exp(\beta_X)} \\ &= (1 - c_2 - c3)S_0^{1+exp(\beta_X)} + c_2 S_0 + c_3 S_0^{exp(\beta_X)} \ . \end{aligned}$$

From Equation 3.7, we know that $c_2 + c_3 = 1$ therefore $(1 - c_2 - c3) = 0$. Hence, we get:

$$S_0^{exp(\beta_X)} = c_2 S_0 + c_3 S_0^{exp(\beta_X)} \ . \tag{3.9}$$

For the function to be universal, it requires $c_2 = 0$ and $c_3 = 1$. This contradicts $c_2 = 1$ and $c_3 = 0$ from Equation 3.8 and concludes the proof. □

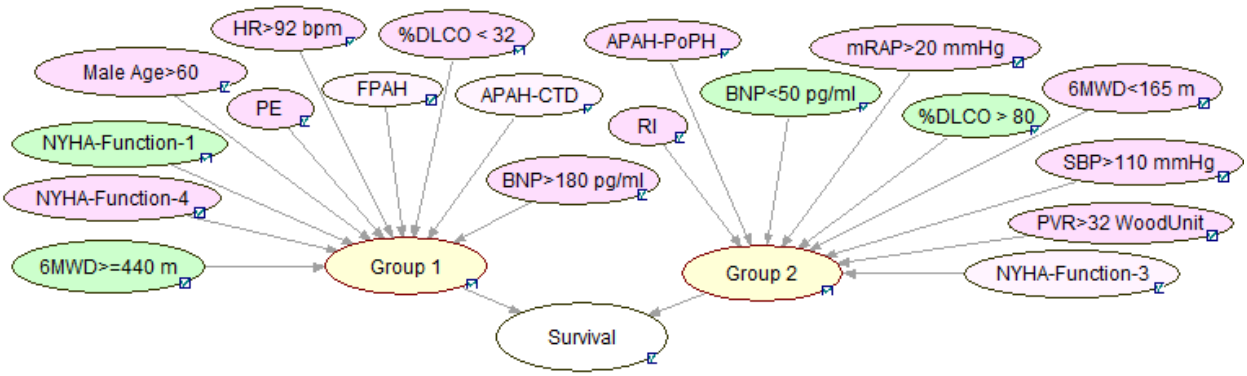


Figure 11: An example of a BN-Cox model decomposition of the a BN-Cox model from Figure 9

While BN-Cox model cannot be decomposed by means of the parent divorcing method, one suggestion offered by Jirka Vomlel was studying other decompositions. Complexity of such decompositions requires studying the rank of the CPH model along the lines of analysis for several popular canonical models (Díez and Galan, 2003; Savicky and Vomlel, 2007; Vomlel and Tichavsky, 2014). While I leave the search for other possible decomposition methods outside of this scope of my dissertation, I offered an experimental analysis of the possible approximate decompositions.

To check the quality of possible approximate decompositions, I performed a series of experiments that consisted of manually decomposing the BN-Cox model and refitting its probabilities from data. Figure 11 shows an example of the structured decomposition of the original BN-Cox model shown in Figure 9. After creating the decomposed BN-Cox model, I generated a data set from the distribution of the original BN-Cox model (at least 5 records for each combination of risk factors). Then, the decomposed BN-Cox model learned from the generated data set using the EM algorithm with intermediate nodes being unobserved (i.e., absent in the data file). Unfortunately, all of the attempts resulted in poor numerical fit and models of clearly inferior quality than the original BN-Cox model.

Figure 12 illustrates the poor quality of approximation of the decomposed model. We used the scatterplot (Figure 12a) of the survival probabilities from the decomposed BN-Cox

model against the ones produced by the original BN-Cox model. In case of perfect fit, the plot would be a perfect diagonal line from $(0, 0)$ to $(1, 1)$. Figure 12b shows the same scatterplot transformed by hexagon binning techniques (Lewin-Koh, 2018). Each hexagon is color-coded according to the number of points falling in that region. While many probabilities are similar, we see a large off-diagonal cloud that indicates poor fit.

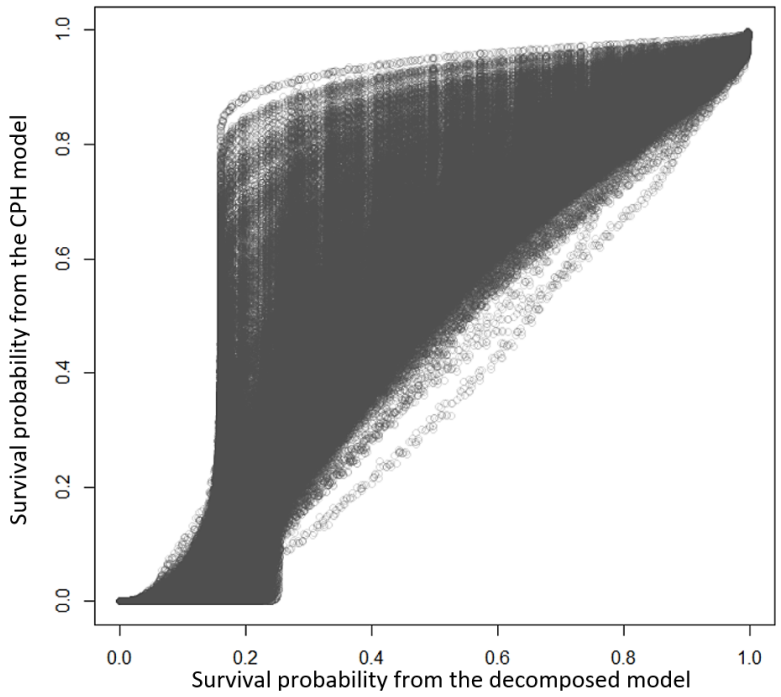
Figure 13 shows the histogram of Euclidean distances between the survival probabilities calculated by the original CPH and the decomposed model for all possible combinations of values of risk factors sorted from the smallest to the largest distance. We clearly see an overall poor fit between the decomposed and the original model.

Although, we have not tested all versions of network decomposition, we tried other decompositions with different number of groups including 4 groups, 6 groups, and 9 groups. All these decompositions confirmed poor approximation of the original model.

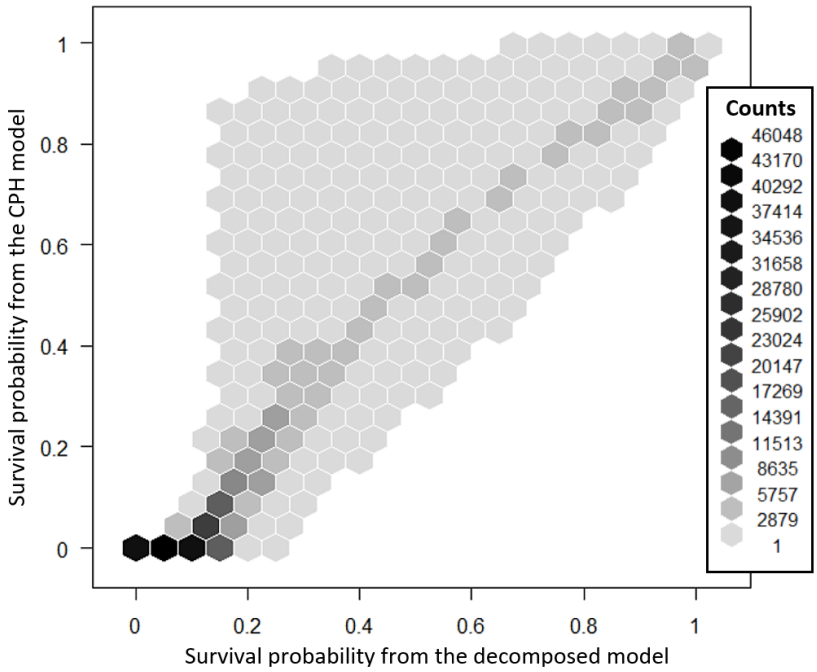
3.4.2 BN-Cox simplication by removing least influential variables

Another method of reducing the complexity of the BN-Cox model is to simplify the CPH model itself by removing the least influential risk factors. It can be expected that some of the risk factors will have minimal effect on the result and omitting them altogether will not lead to much loss of precision. On the other hand, removing each of these least influential factors will cut the size of the survival node's CPT by at least half. In practice, there are several techniques of variable selection in survival analysis (Fan and Li, 2002). We started out by evaluating the effect of removing the weakest variable. The *weakest* means the variable with the highest p value and possibly the smallest value of the β coefficient. The larger the value of p, the less certain we are that the risk factor is really affecting survival, the smaller the value of β, the weaker the effect, even if there is any.

I performed simplification experiments on the Recidivism CPH model consisting of seven binary risk factors listed in Table 5. First, I compared **the effect of removing the least significant variable against the effect of removing the most significant one**. The weakest variable in Table 5 seems *paro* with $\beta = -0.06721$ and $p = 0.7288$, while the strongest variable seems *wexp* with $\beta = 0.41055$ and $p = 0.0403$. To create a simplified model,



(a) Scatterplot



(b) Scatterplot with Hexagonal Binning

Figure 12: Survival probabilities produced by the decomposed model against survival probabilities produced by the original CPH model.

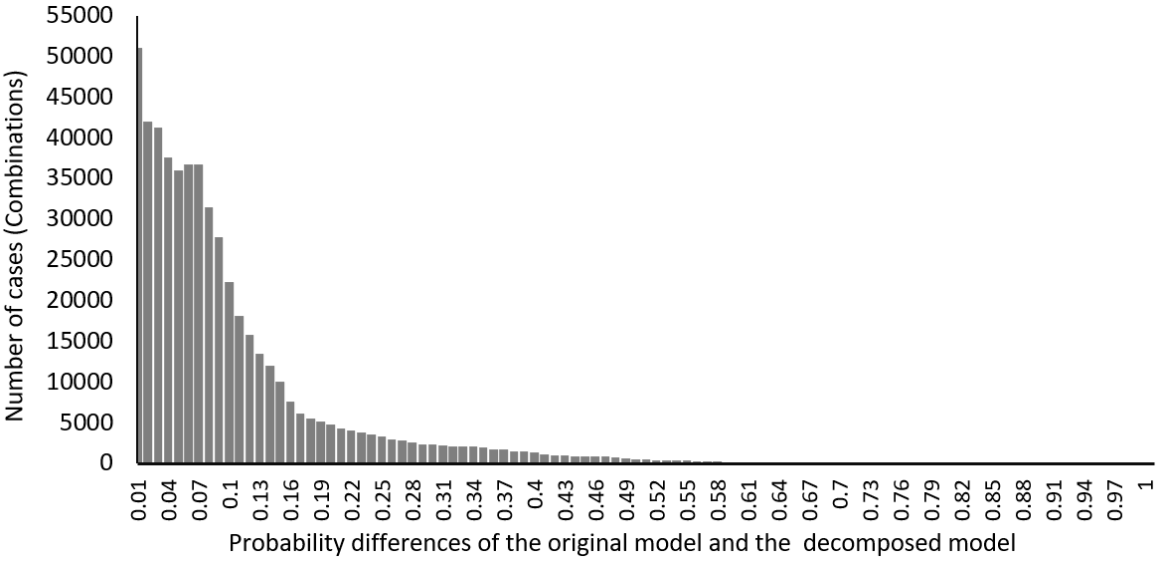


Figure 13: The histogram showing the Euclidean distance between the survival probabilities produced by the original BN-Cox model and the decomposed model sorted from the smallest to largest distance.

Risk factor (X_i)	β	$exp(\beta)$	p-value
X_1: fin	-0.40415	0.6675	0.0339
X_2: race	0.22931	1.2577	0.4549
X_3: wexp	0.41055	1.5076	0.0403
X_4: mar	-0.49926	0.6070	0.1874
X_5: paro	-0.06721	0.9350	0.7288
X_6: prio	0.28708	1.3325	0.2654
X_7: educ	-0.80736	0.4460	0.0557

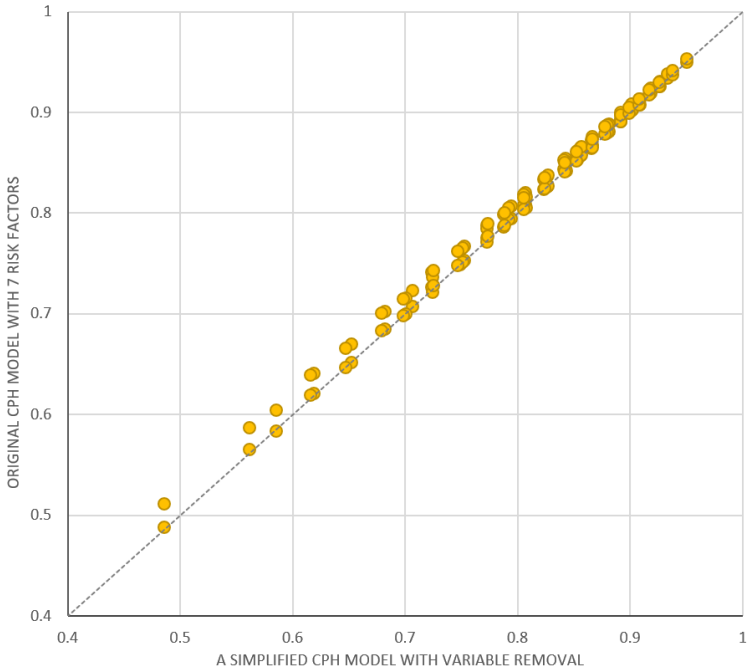
Table 5: A list of seven binary risk factors, their corresponding coefficients β, hazard ratio $exp(\beta)$, and p-value estimated from the Recidivism data set.

I removed the selected variables from the data set and refit the CPH model. Hence, we have two refitted models: (1) one with the variable *paro* (weakest) removed, and (2) one with the variable *wexp* (strongest) removed. I compared the predicted survival probabilities against the original CPH model shown in Figure 14. The original CPH model consisted of seven binary risk factors resulting in $2^7 = 128$ predicted survival probabilities. Since we removed one variable from the original CPH model, the total number of predicted probabilities in the simplified model is $2^6 = 64$. Two survival probabilities in the original CPH model correspond to one probability in the modified models. For example, the survival probabilities produced by the original model when *fin*=0, *race*=0, *wexp*=0, *mar*=0, *prio*=0, *educ*=0, *paro*=0 and when *fin*=0, *race*=0, *wexp*=0, *mar*=0, *prio*=0, *educ*=0, *paro*=1 are mapped to the survival probability produced by the *paro*-removed model when *fin*=0, *race*=0, *wexp*=0, *mar*=0, *prio*=0, *educ*=0.

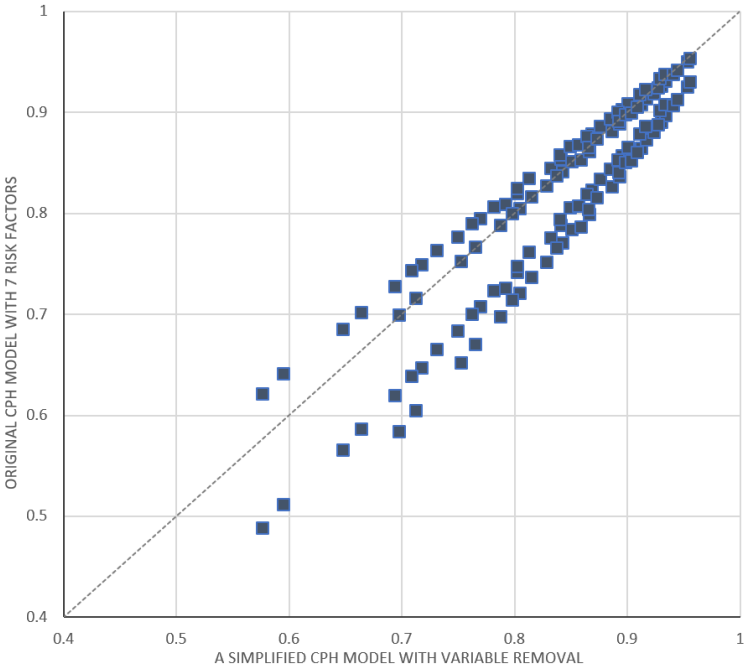
The results obtained by removing the least significant variable (Figure 14a) are closer to the original model than the results obtained by removing the strongest variable (Figure 14b). In this experiment, we identified the least/most significant variables by their β and p-values in the original CPH model. However, one can use any variable selection method here (Fan and Li, 2002).

Removing a weak variable and refitting works only when we have the original data set. In practice, however, we often have only the CPH parameters and not the data from which they were obtained. For those variables with small influences, it can be expected that setting those variables to be *absent* will be similar to removing those variables in the simplified refitted model. In a follow-up experiment, I evaluated **the effect of fixing state of the weakest variable (*paro*) to *absent* against the simplified refitted model**. I used the original CPH model (Table 5) and simplified the model by fixing the state of *paro*. As a result, we have two sets of predicted probabilities from the fixed-state model: cases when *paro* is fixed to *absent* ($paro = 0$) and cases when *paro* is fixed to *present* ($paro = 1$). Then, we compared those results to the original CPH models and the model with *paro* removed (Figure 15).

Figure 15a shows the predicted probabilities of all cases with $paro = 0$. The diagonal grey-dotted line shows ideal probabilities with all $paro = 0$ cases as produced from the original CPH model. It can be expected that all probabilities produced from the model fixing

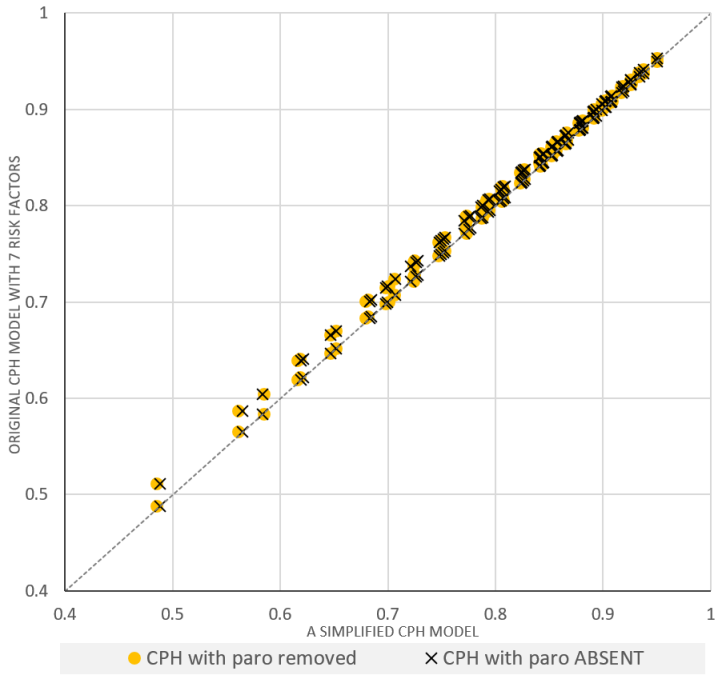


(a) Weak variable (*paro*) removed

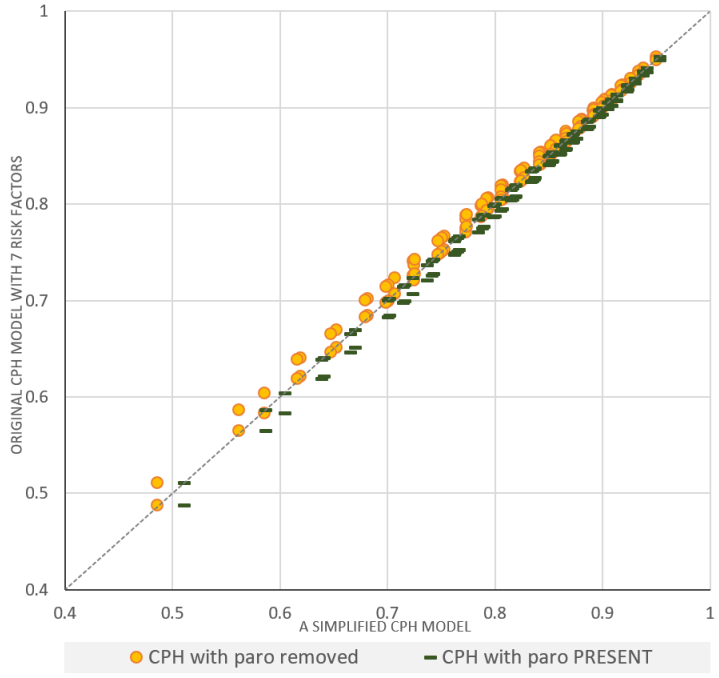


(b) Strong variable (*wexp*) removed

Figure 14: The scatterplot of the survival probability produced by the simplified models against the survival probability produced by the original CPH model.



(a) Cases when $paro = 0$



(b) Cases when $paro = 1$

Figure 15: The scatterplots shows probabilities produced by two fixed-variable models (*paro-absent* and *paro-present*) against one variable-removed model (*paro*-removed model).

paro to *absent* are perfectly on the diagonal line, while, setting *paro* to *present* produced some errors. We observed that fixing *paro* to *absent* produced results very close to the results from the *paro*-removed model. For those cases with $paro = 1$ in the original CPH model, we also observed similar trends in Figure 15b. All probabilities from the model fixing *paro* to *present* lie perfectly on the diagonal grey-dotted line, which shows ideal probabilities with all $paro = 1$ cases as produced from the original CPH model. In this case, setting *paro* to *absent* produced errors. In summary, we could approximate the simplified model by setting state of a risk factor to *absent* in the original model without refitting the model from the data set. However, fixing state of the variable to *absent* still produces errors for those original cases with the risk factor *present*, and vise versa.

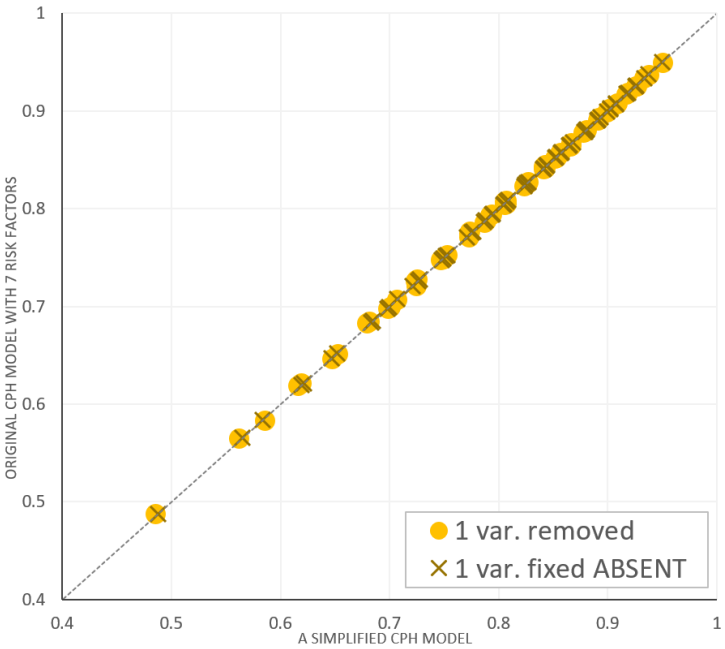
To verify this observation, I also created four models with one, two, three, and four least significant variables *absent*. I refitted corresponding four models by removing the least influential risk factors. Figure 16 shows the results for the simplified models with both fixed to be *absent* and refitted models against the original CPH model for different numbers of risk factors. We can see that removal of multiple variables, especially when their influence is larger, can lead to departure from the ideal precision (the diagonal line in the plots). We should add that removing four of the seven Recidivism variables was expected to make a large impact on the quality of the resulting model. We believe that the loss of precision will be much smaller when the number of variables removed is small.

As shown in the previous experiment, fixing the small-influence risk factors to *absent* is similar to removing those risk factors from the model but still produces error when the risk factors are *present*. In Bayesian networks, we can use marginalization to simplify a model. Marginalization amounts to removing a risk factor X_i from consideration while preserving the joint probability distribution among the remaining variables and the effect of the remaining risk factor on the survival probability, s. Marginalization of a risk factor, X_i, amounts to:

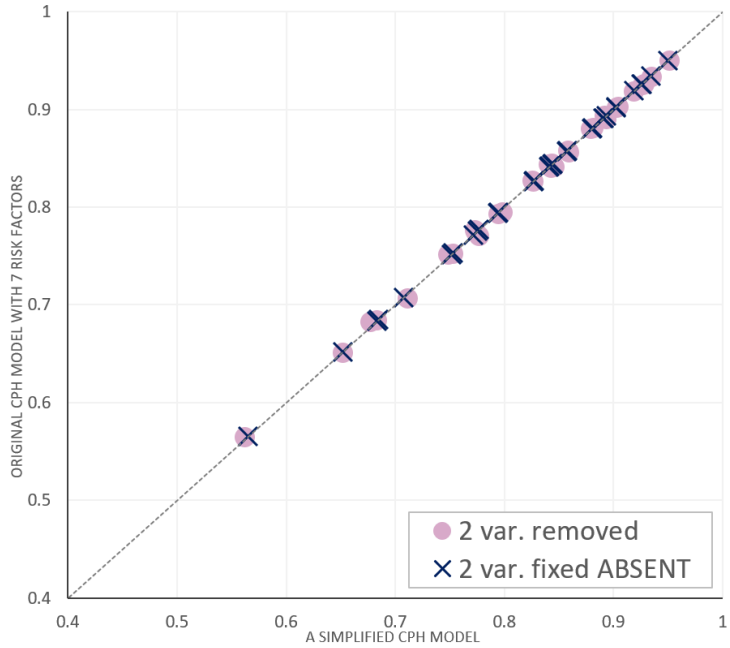
$$Pr(s \mid \xi) = \sum_{i=1}^{n} Pr(s \mid \xi, x_i) \cdot Pr(x_i) \ , \tag{3.10}$$

where x_i are states of, X_i, $Pr(s)$ is the survival probability, and ξ are all other risk factors.

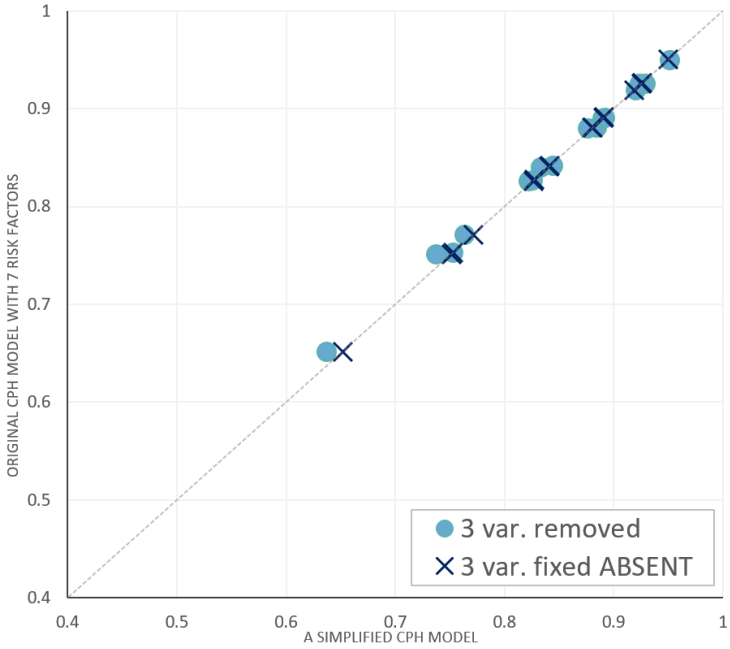
I performed an experiment on the use of marginalization and compared the results against the results from previous experiments. I created a BN-Cox model from all CPH parameters in



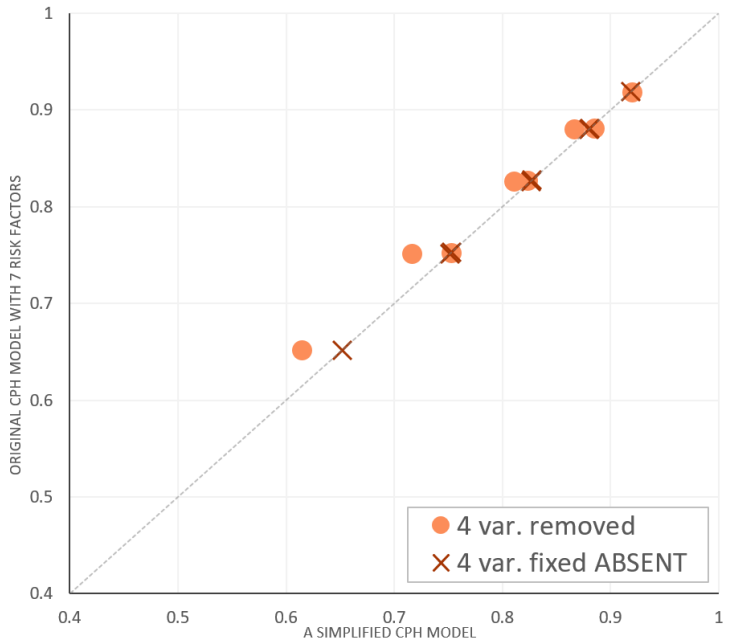
(a) One risk factors *absent* vs. *removed*



(b) Two risk factors *absent* vs. *removed*



(c) Three risk factors *absent* vs. *removed*



(d) Four risk factors *absent* vs. *removed*

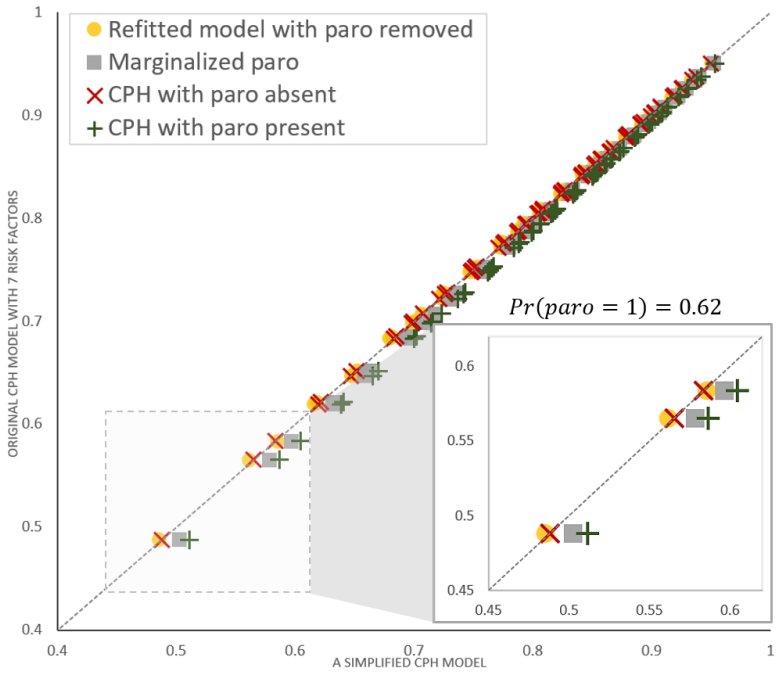
Figure 16: Effect of *absent* and *removed* risk factors in the simplified models against the original CPH model. The predicted probabilities from the simplified models are compared only for the cases when those selected risk factors in the original CPH model are *absent*.

Table 5, then marginalized the variables *paro* and *wexp* out. Hence, I collected the predicted probabilities from each marginalized model and compared the results to the original CPH model, the model with variable removed, and the fixing-state model, case-by-case (Figure 17).

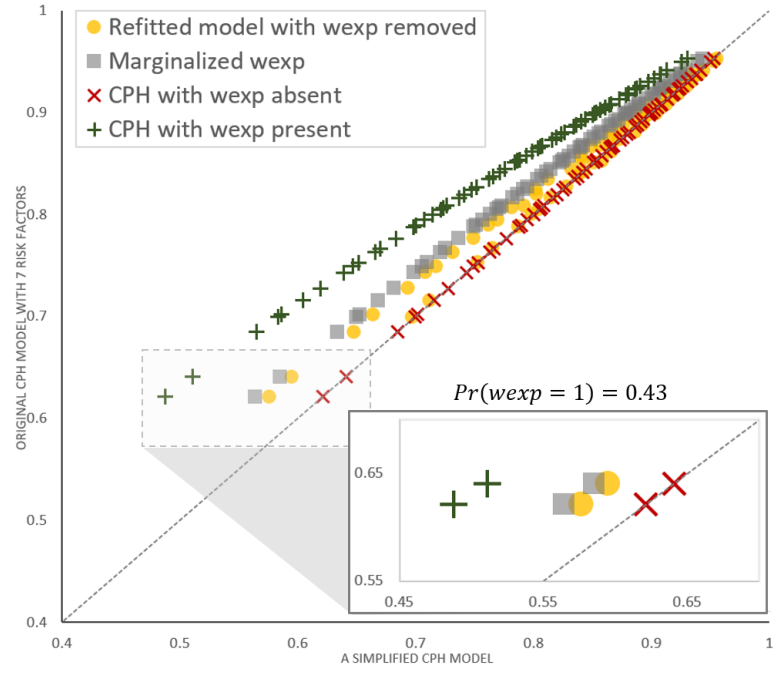
Figure 17a shows the result of the *paro*-marginalized model (the grey square markers) against the results from previous experiments, including the results from the no-*paro* refitted model (the yellow circle markers), the result from fixing-state model with *absent paro* (the red cross markers), the result from fixing-state model with *present paro* (the green plus markers), and the result from the original model when *paro*=0 (the diagonal grey-dotted line). As we expected, the marginalized model produced probabilities in-between the results from the *absent*- and *present*-fixed model, and also closer to the *present*-fixed model, since 68 percents of inmates in the Recidivism data set have been on parole before being released. We observed the same trends in other figures: The marginalized model produced the results by weighing out the effect of each state by its prior probability. We believe that marginalization is the correct way to remove the selected variable since it still preserves the effect of the risk variables.

In summary, I have studied two ways of making the BN-Cox model computationally efficient. Our main challenge to making BN-Cox more practical is an exponential growth of the conditional probability tables of the survival variable node. Two approaches were tested: (1) parent divorcing, and (2) removing least influential risk factors. The BN-Cox model turns out to be not decomposable and approximating of decomposition leads to high loss of accuracy. Hence, we suggest to simplify the network structure by removing the least influential risk factors.

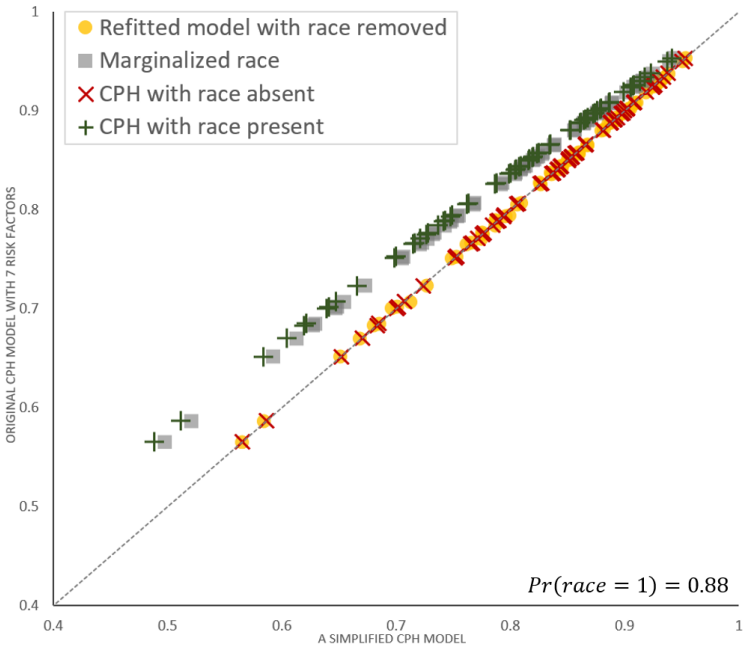
We can use any statistical variable selection method Fan and Li (2002) to simplify or reduce the number of risk factors in the CPH models when we have a data set to refit the simplified model. However, when data are not available, we can simplify the model by removing least influential risk factors based on both the value of β coefficients and the statistical significance by marginalization, as it leads to smallest error on the average.



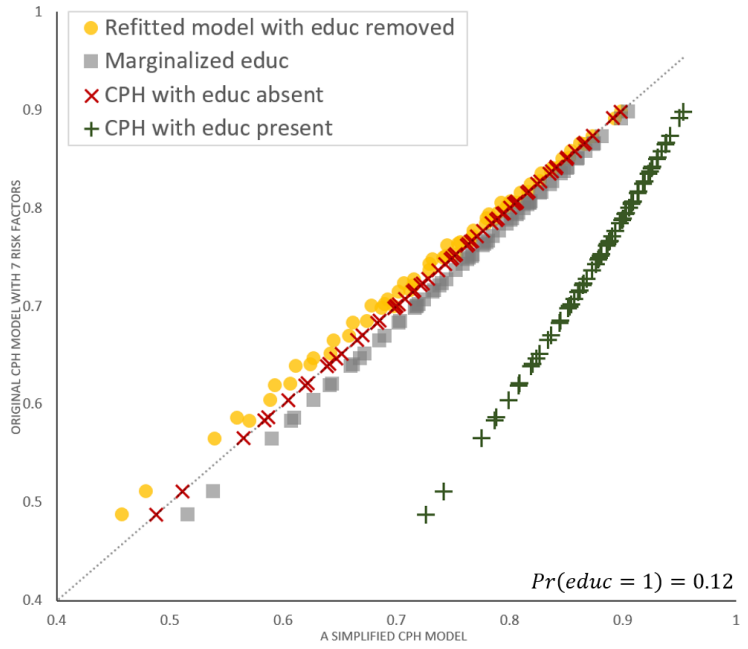
(a) Marginalized *paro*



(b) Marginalized *wexp*



(c) Marginalized *race*



(d) Marginalized *educ*

Figure 17: Effect of marginalized risk factors in the simplified models against the original CPH model, the refitted model, and the fixed-state models. The diagonal gray line shows the ideal probability as produced from the original CPH model.

4.0 Bayesian network vs CPH model: context sensitivity

This chapter examines the character of influence of risk factors to a given outcome in Bayesian network model vs. CPH models. I show another point of departure of the CPH model from data, notably influence of individual risk factors, as expressed by CPH hazard ratios, against dynamic and flexible entropy-based measure of influence.

I discuss static and dynamic character of influences in Section 4.1. Section 4.2 provides more details in entropy-based measurement of influence. Finally, Section 4.3 empirically demonstrates for simplifying the BN-Cox model for the sake of representational and computational efficiency

4.1 Static vs. dynamic influence

In CPH models, influence of each individual risk factor to the outcome is expressed by a number called *hazard ratio*. The hazard ratio is defined as a ratio of the hazard in the corresponding risk group to the hazard in the baseline group (i.e., a hypothetical group in which none of the risk factors are *present*). This ratio is, by one of the proportional hazard assumptions of the CPH model, constant over time. For example, Table 4 reports the hazard ratio for *pericardial effusion* as 1.35. This means that patients with pericardial effusion have a 35% higher risk of dying from PAH than patients at the baseline state (i.e., patient with no pericardial effusion). When performing prediction for estimating the outcome probability, this influence still do not change regardless of the context of other risk factors, i.e., the presence or absence of other risk factors. The hazard ratio is fixed as it can be considered as a *static* influence of the risk factor to the outcome.

Unlike CPH models, Bayesian networks do not explicitly define the influence of individual risk factor to the outcome. The structure of the network defines interaction between risk factors. As some of the risk factors are observed, the role of other risks, expressed by their potential to change their influence on the outcome variable, changes. I define this impact as

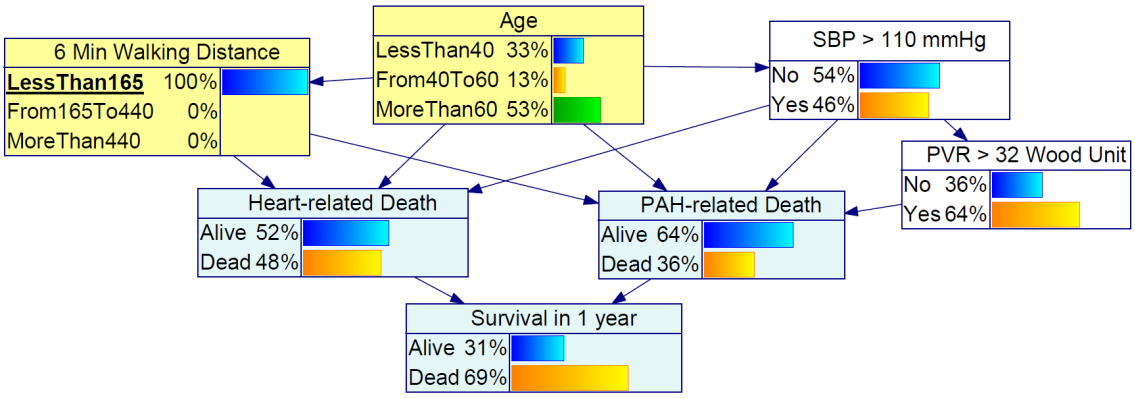


Figure 18: An example Bayesian network predicting survival of patients with partial observations: only *6 Minute Walking Distance* is observed.

dynamic influence.

4.2 Entropy-based measurement of influence

In Bayesian networks and information theory (Shannon, 1948), entropy measures the degree of uncertainty of a given random variable, X; defined as

$$H(X) = -\sum_{i=1}^{n} P(x_i) log_2 P(x_i) \ , \tag{4.1}$$

where $P(x_i)$ is a probability of an individual state, x_i in a random variable X.

In this section, we apply the concept of change in entropy to measure an influence of a risk factor to an outcome. Suppose we have a Bayesian network predicting survival of patients given their list of risk factors (Figure 18). Our outcome variable is S: *Survival in 1 year*. To estimate an influence of each state in a given risk factor, we first measure an entropy of S before observed any risk factors, i.e., $H(S)_o$. Then, we observed a risk factor (X), such as, *6 Minute Walking Distance (6MWD)* with a state x_i, and measure the entropy of S, i.e, $H(S|X = x_i)$. Change in entropy at the S node ($H(S|X = x_i) - H(S)_o$), therefore, defines influence of the observed state of risk factor to the outcome variable.

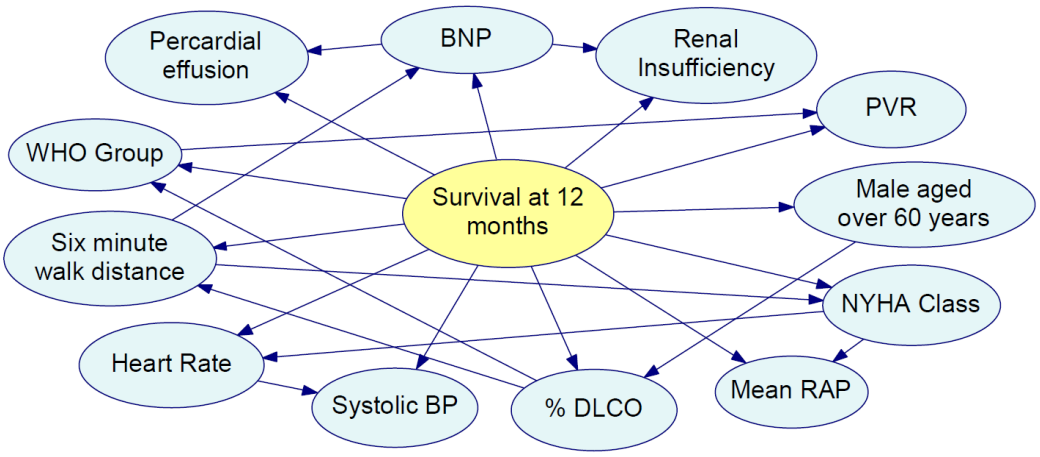


Figure 19: The TAN Bayesian network learned from the REVEAL registry data.The node *Survival at 12 months* is the predicted outcome variable, connected to every risk factor.

4.3 Failure of the CPH model to capture dynamic character of influence

As I mentioned above, one of the important assumptions of the CPH model is that the individual hazard ratios are constant over time and do not change with presence or absence of other risk factors. This assumption did not seem realistic, so I performed the following experiment to probe it.

4.3.1 Methods

For the purpose of this experiment, I use an existing Bayesian network model for PAH risk assessment. Figure 19 shows the Bayesian network which is one of the Bayesian network model for Pulmonary Arterial Hypertension Outcomes Risk Assessment (PHORA) project, The model was learned from a data set of 2,456 patient records from the REVEAL registry data by using a Tree Augmented Naïve (TAN) learning algorithm.

The list of variables was preserved from the REVEAL risk score calculator with the same discretization levels. It is clear that some of the variables in the table have been artificially created for the purpose of CPH modeling. For example, the three WHO variables are mutually exclusive states of a single variable. The same holds for the NYHA class, Six-

Table 6: A list of 19 binary risk factors from the REVEAL risk score calculator Benza et al. (2010) along with their counterparts in the Bayesian network. The baseline states are shown in **bold**.

Risk factors	**Random Variable**	**States**
APAH-CTD	WHO Group	APAH-CTD
FPAH		FPAH
APAH-PoPH		APAH-PoPH
		Other
Renal insufficiency	Renal insufficiency	Yes
		No
Male >60 years age	Male >60 years	Yes
		No
NYHA/WHO FC I	NYHA/WHO FC	I
		II
NYHA/WHO FC III		III
NYHA/WHO FC IV		IV
SBP <110 mmHg	Systolic BP	<110
		≥110
Heart Rate >92bpm	Heart rate	>92
		≤92
6MWD <165 m	6 Min Walking Distance	<165
		165-<440
6MWD ≥440 m		≥440
BNP <50 pg/ML	BNP	<50
		50-180
BNP >180 pg/ML		>180
Pericardial effusion	Pericardial effusion	Yes
		No
% DLCO ≤32%	% DLCO	≤32
		>32-<80
% DLCO ≥80%		≥80
Mean RAP > 20 mmHg	Mean RAP	>20
		≤20
PVR >32 WU	PVR	>32
		≤32

Min Walking Distance, BNP and % DLCO variables. The CPH model required them to be risk factors, modeled as states of binary variables. These states were combined back into single variables, as the laws of probability require. For all numerical variables, which we had to discretize in order to include them into the Bayesian network model, we applied the cut points used by the REVEAL risk score calculator. We also added a baseline state, wherever needed but not explicitly defined in the calculator. Table 6 shows the list of risk factors from the REVEAL risk score calculator along with their counterparts in the Bayesian network.

The TAN learning algorithm is one of the most popular learning methods for Bayesian network classification. TAN extends the Naïve Bayes structure by adding most important interdependencies between feature variables. At the same time, the algorithm constraints the maximum number of incoming arcs to two and, by this, keeps the conditional probability tables (CPTs) in individual nodes small. Small CPTs mean a small number of parameters, which can be learned reliably even when the learning data set is small. Effectively, when the learning data set is small, the quality of the parameters remains high and the entire TAN model typically matches well the joint probability distribution that generated the data. Hence, statistical properties of a data set generated from the TAN network will not depart too far from the statistical properties of the original data set.

Given a 30,000 record data set, I was able to simulate situations in which some of the risk factors have been observed (this amounted to selecting a subset of the data) and to learn a new CPH model from the resulting data. Our goal was to check whether the hazard ratios for those variables that have not been observed yet are indeed constant, i.e., the same in the selected subset of records.

4.3.2 Discussion

Figure 20 shows the result of this experiment. Figure 20a shows the hazard ratios (HRs) calculated for subsets in which a single risk factor (listed in the header of the table) has been observed. All columns differ from the first column, which contains the original CPH parameters that was learned from the generated data. Figure 20b shows differences between

CPH	HR	APAH_CTD	APAH_PoPH	FPAH	NYHA_I	NYHA_III	NYHA_IV	SIXMWD_165	SIXMWD_440	BNP_50	BNP_180	DLCO_32	DLCO_80	SYSBP	HR	MRAP	PVR	RI	MALE_60YR	PERI_EFFU
APAH_CTD	1.52				1.30	1.68	1.20	1.62	1.09	1.54	1.52	1.65	0.27	1.55	1.27	1.39	4.59	1.20	1.71	1.51
APAH_PoPH	2.42				5.87	2.58	1.67	1.89	2.90	3.62	2.12	2.63	3.40	2.51	2.15	2.72	3.22	1.96	1.57	2.33
FPAH	1.82				0.70	1.56	1.63	1.55	2.69	2.91	1.53	1.02	2.71	1.83	1.39	1.88	5.32	1.32	0.54	1.77
NYHA_I	0.35	0.40	0.51	0.09				1.09	0.28	0.36	0.37	0.44	0.19	0.37	0.52	2.67	1.62	0.20	0.36	0.41
NYHA_III	1.46	1.56	1.44	1.02				1.04	1.48	1.52	1.43	1.53	1.29	1.42	1.58	0.93	2.14	1.44	1.38	1.53
NYHA_IV	4.20	3.76	3.39	3.91				3.03	1.40	5.13	3.93	3.83	4.33	3.77	3.55	4.57	4.36	3.65	2.76	4.02
SIXMWD_165	1.46	1.53	1.23	1.83	11.80	1.36	1.48			1.91	1.51	1.43	1.31	1.52	1.45	1.52	1.28	1.48	1.34	1.43
SIXMWD_440	0.58	0.46	0.55	0.74	0.77	0.62	0.16			1.41	0.48	0.97	0.76	0.55	0.73	0.67	0.61	0.39	0.52	0.53
BNP_50	0.41	0.40	0.46	0.49	0.68	0.37	0.38	0.52	0.89			0.52	0.53	0.39	0.39	0.38	0.49	0.25	0.49	0.45
BNP_180	1.67	1.72	1.36	1.25	1.94	1.65	1.51	1.91	1.79			1.73	1.60	1.64	1.56	1.58	1.97	1.18	1.81	1.76
DLCO_32	1.39	1.38	1.56	0.98	1.08	1.42	1.25	1.23	3.12	2.41	1.32			1.44	1.41	1.18	1.33	1.61	1.44	1.39
DLCO_80	0.84	0.14	1.13	1.30	0.59	0.79	0.75	0.56	1.14	1.03	0.79			0.99	0.84	0.86	0.88	0.75	0.17	0.89
SYSBP	1.72	1.71	1.79	1.81	2.13	1.73	1.54	1.78	1.68	1.64	1.70	1.78	2.17		1.96	1.56	1.88	1.41	1.73	1.66
HR	1.33	1.16	1.35	1.23	2.55	1.46	0.93	1.21	1.92	1.40	1.28	1.24	1.61	1.47		1.30	1.61	1.41	1.11	1.34
MRAP	1.46	1.40	1.48	1.85	10.87	1.11	1.81	1.50	1.60	1.44	1.47	1.42	1.67	1.40	1.50		2.98	1.95	1.69	1.36
PVR	1.99	3.56	2.15	4.10	9.00	2.50	1.44	2.13	2.24	2.71	2.02	1.53	2.25	2.00	1.95	2.19		1.98	1.59	1.54
RI	1.60	1.43	1.51	1.32	0.54	1.66	1.51	1.52	1.24	1.38	1.54	1.73	1.43	1.42	1.74	2.17	0.98		1.42	1.56
MALE_60YR	1.83	2.01	1.24	0.79	1.46	1.91	1.45	1.65	1.75	2.10	1.80	1.98	0.42	1.83	1.58	2.41	1.40	1.68		1.65
PERI_EFFU	1.56	1.54	1.54	1.65	1.62	1.64	1.42	1.48	1.47	1.70	1.59	1.54	1.73	1.51	1.60	1.34	0.87	1.50	1.32	

(a) Hazard ratios of each observed group

CPH	HR	APAH_CTD	APAH_PoPH	FPAH	NYHA_I	NYHA_III	NYHA_IV	SIXMWD_165	SIXMWD_440	BNP_50	BNP_180	DLCO_32	DLCO_80	SYSBP	HR	MRAP	PVR	RI	MALE_60YR	PERI_EFFU
APAH_CTD	1.52				-15%	10%	-21%	6%	-29%	1%	0%	8%	-82%	2%	-17%	-9%	201%	-21%	12%	-1%
APAH_PoPH	2.42				142%	7%	-31%	-22%	20%	49%	-12%	9%	40%	4%	-11%	12%	33%	-19%	-35%	-4%
FPAH	1.82				-61%	-14%	-10%	-15%	48%	60%	-16%	-44%	49%	1%	-23%	3%	193%	-27%	-70%	-3%
NYHA_I	0.35	17%	49%	-74%				216%	-19%	3%	6%	28%	-44%	7%	50%	673%	369%	-42%	5%	18%
NYHA_III	1.46	6%	-2%	-30%				-29%	1%	4%	-2%	5%	-12%	-3%	8%	-37%	46%	-2%	-6%	4%
NYHA_IV	4.20	-10%	-19%	-7%				-28%	-67%	22%	-6%	-9%	3%	-10%	-15%	9%	4%	-13%	-34%	-4%
SIXMWD_165	1.46	4%	-16%	25%	706%	-7%	1%			31%	3%	-2%	-11%	4%	-1%	4%	-13%	1%	-9%	-3%
SIXMWD_440	0.58	-22%	-5%	27%	31%	7%	-72%			142%	-17%	66%	31%	-5%	25%	15%	5%	-33%	-12%	-10%
BNP_50	0.41	-2%	12%	19%	67%	-9%	-6%	28%	117%			28%	29%	-4%	-5%	-6%	20%	-38%	21%	9%
BNP_180	1.67	3%	-19%	-25%	17%	-1%	-10%	15%	7%			4%	-4%	-1%	-6%	-5%	18%	-29%	9%	5%
DLCO_32	1.39	-1%	12%	-30%	-23%	2%	-10%	-11%	125%	74%	-5%			4%	1%	-15%	-4%	15%	4%	0%
DLCO_80	0.84	-83%	34%	55%	-30%	-5%	-11%	-33%	36%	23%	-6%			17%	0%	2%	5%	-11%	-80%	6%
SYSBP	1.72	-1%	4%	5%	24%	1%	-10%	4%	-2%	-5%	-1%	4%	26%		14%	-9%	9%	-18%	1%	-3%
HR	1.33	-13%	1%	-7%	92%	10%	-30%	-9%	45%	5%	-3%	-7%	21%	11%		-2%	21%	6%	-17%	1%
MRAP	1.46	-4%	2%	27%	646%	-24%	24%	3%	10%	-1%	1%	-2%	15%	-4%	3%		105%	34%	16%	-7%
PVR	1.99	79%	8%	106%	353%	26%	-27%	7%	13%	36%	2%	-23%	13%	1%	-2%	10%		-1%	-20%	-23%
RI	1.60	-11%	-6%	-18%	-66%	4%	-6%	-5%	-23%	-14%	-4%	8%	-11%	-11%	8%	35%	-39%		-11%	-3%
MALE_60YR	1.83	10%	-32%	-57%	-20%	4%	-21%	-10%	-4%	14%	-2%	8%	-77%	0%	-14%	31%	-24%	-8%		-10%
PERI_EFFU	1.56	-2%	-2%	5%	4%	5%	-9%	-5%	-6%	9%	1%	-2%	11%	-3%	2%	-14%	-44%	-4%	-15%	

(b) Percent relative change of the hazard ratio from the baseline

Figure 20: Effect of observing one of the risk factors on the hazard ratios of the remaining variables

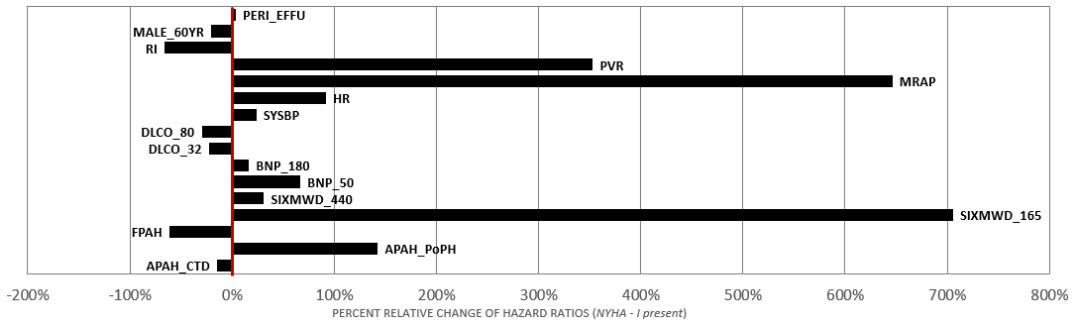


Figure 21: Percent relative change of hazard ratios when we observed *NYHA-I*

the hazard ratios calculated for each of the cases relative to the original parameters and expressed as a percentage of change. Colors give a visual indication of where the largest differences are. Some of the hazard ratios in the table have changed as much as 700%.

Figure 21 shows NYHA-I column of Figure 20b in graphical format. We can see that these risk factors, e.g., *SIXMWD_165*, *MRAP*, become very important once we observe that the patient belongs to *NYHA Functional Class I*. HRs are static and are not capturing this context-induced change.

Modeling with Bayesian networks does not require us to make such assumptions. In fact, varying degree of influence of risk factors is a natural consequence of varying context. As some of the risk factors are observed, the role of other risks, expressed by their potential to impact of the survival variable, changes.

Figure 22 shows a scatterplot of hazard ratios and entropy for the *NYHA Functional Class I* case. The plot shows the baseline situation, i.e., when no risk factors are observed (triangle marks) and a change in context, when *NYHA-I* is observed (circles). The two measures are correlated with each other at the baseline. However, the entropy changes with context, while the hazard ratios stay the same by definition.

Bayesian networks offer more flexibility and result in more intuitive models. As shown in Figure 22, the assumptions of the CPH model may be unrealistic in practice. Bayesian networks model naturally varying magnitude of influence of risk factors as other factors are observed.

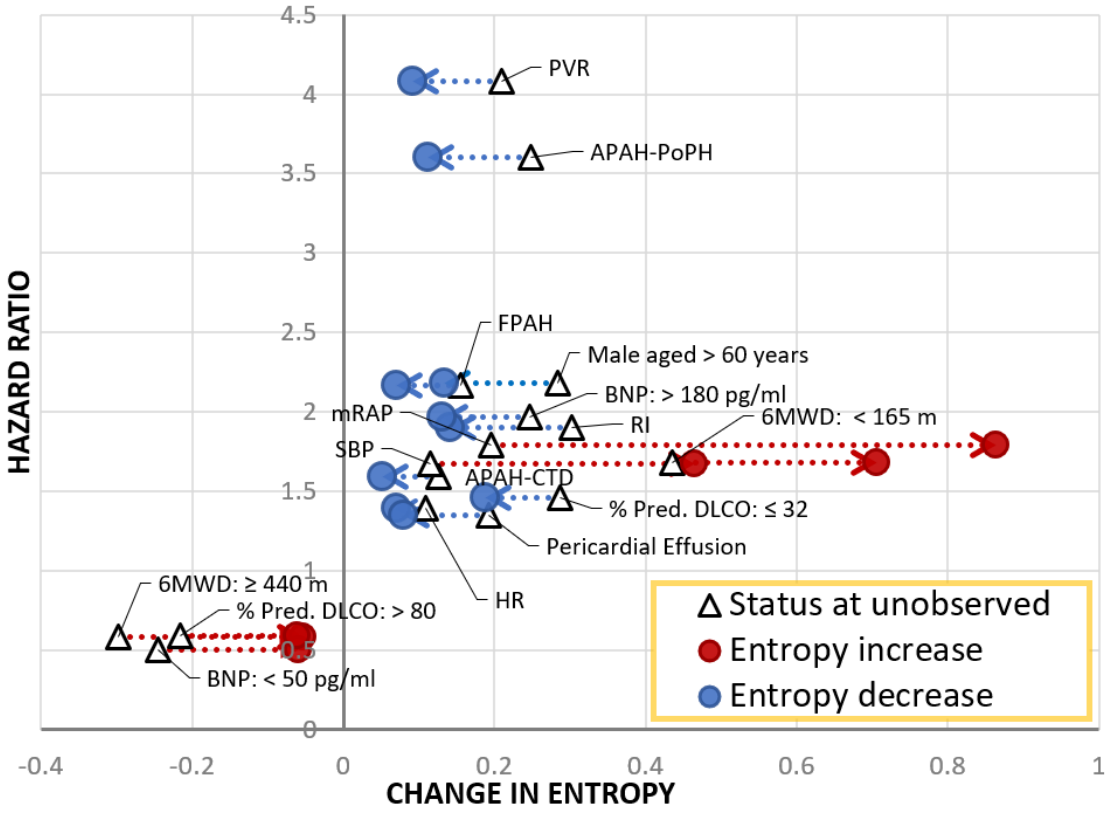


Figure 22: An example of the movement of the entropy when we observed *NYHA-I*. The entropy change or the influence of the risk factors is clearly context-dependent

5.0 Enhancing learning of Bayesian network parameters by means of priors

When building a Bayesian network form a small data set, it is common that some combinations of conditions in conditional probability tables are represented by few or no data records. As a result, the quality of parameters of the resulting Bayesian network is poor, which is usually manifested by uniform distributions (Oniśko et al., 2001). Uniform distribution are essentially based on uninformed uniform priors. The process of learning parameters can be improved by having better priors than uniform distribution. In this chapter, I discuss and provide empirical evaluation on approaches to enhance learning of Bayesian networks by different sources of priors, including prior from experts knowledge in Section 5.2, and simplified probabilistic models such as a Tree-Augmented Naïve Bayes in Section 5.3. The objective of these approaches is to improve the quality of learned parameters and, hence, model accuracy, especially when we deal with small data sets.

5.1 Data sets used in experiments reported in this chapter

For the purpose of all experiments in this chapter, I used the data sets listed in Table 7, all selected from the UCI Machine Learning Repository data sets (Dua and Karra Taniskidou, 2019). I used the following selection criteria for the sets:

- The data set must include a discrete class variable for the purpose of model evaluation.
- The data set should have a wide range in the number of records.
- The selected data sets have a wide range in the number of variables, e.g., 8-30, for the purpose of evaluation.
- The majority of variables (i.e., at least 1/2 of variables) should be discrete variables to minimize the need of discretization. The remaining numerical variables are excluded from model learning when there are enough discrete variables (at least 6 variables) to create a Bayesian network model.

Table 7: A list of data sets

Data set	**# Class**	**# Records**	**% Missing**	**# Attribute** (Discrete/Cont./Constant)
Adult	2	32561	7.37	14 (8/ 6/ 0)
Breast Cancer	2	288	3.82	9 (9/ 0/ 0)
Credit Approval	2	690	5.36	15 (9/ 6/ 0)
Flag	6	194	0.00	28 (18/10/ 0)
German Credit	2	1000	0.00	20 (13/ 7/ 0)
Lymphography	4	148	0.00	18 (18/ 0/ 0)
Mushroom	4	8124	30.53	22 (21/ 0/ 1)
Nursery	5	12960	0.00	8 (8 / 0/ 0)
Artificial REVEAL	2	2500	0.00	14 (14/ 0/ 0)

- The data set must have at least 2/3 records with no missing values to prevent learning a poor quality of a Bayesian network. Those records that contain missing values will be removed for Bayesian network structure learning.

The remaining data set, *Artificial REVEAL*, is an artificial data set of 2,500 records generated from a TAN REVEAL 2.0 network. The TAN REVEAL 2.0 network is one of Bayesian networks developed for the PHORA project which was learned from the REVEAL registry data set using the REVEAL risk score calculator 2.0 (Benza et al., 2019) cut points. The artificial REVEAL data set matches the above criteria. It also represents the practical problem for Bayesian networks in the PAH risk assessment.

5.2 Priors obtained from experts

This section discuss a method of using priors from experts for enhancing parameters in Bayesian networks. I provide a background on obtaining priors from experts and its alternative (Section 5.2.1 and 5.2.2.) Finally, Section 5.2.3 describe an experiment that tests the proposed methods

5.2.1 Elicitation of probabilities from experts

When building a Bayesian network, the most common sources of priors are statistical data, literature, human experts, or a combination of these (Druzdzel and van der Gaag, 2000). Given the structure of a Bayesian network, the numerical probabilities in the conditional probability table (CPT) of each variable are needed. Elicitation of probabilities requires a good design of knowledge engineering process with a combination of tools.The simplest tools focus on single probabilities and are, therefore, extremely laborious. There are methods that ease the elicitation burden. For example, van der Gaag et al. (1999) developed an elicitation method including transcribed texts for explaining conditional probability to be assessed along with a scale of verbal probability expression mapping to their numerical probability (e.g., *probable = 0.85*, *improbable = 0.15*). Although there are many proposed elicitation methods, this process still requires a lot of time and effort (Lucas et al., 2004).

5.2.2 Canonical gates as an aid to obtain priors

Another technique to facilitate the process of eliciting probability from experts is to use canonical gates to reduce the number of parameters of conditional probability distribution. The conditional probability distribution are stored in the conditional probability table (CPT). The CPT of a node with n binary parents will need by 2^n parameters, which poses substantial difficulties for knowledge engineering. For a sufficiently large 10, obtaining numerical parameters from an expert is becomes practically impossible. Zagorecki and Druzdzel (2013) found that typically over half of probability distributions in practical Bayesian networks can be reasonably approximated by canonical gates. Models based on canonical gates require fewer parameters ($2n$ instead of 2^n in binary case). This increase the quality of parameter learning (Oniśko et al., 2001) and reduces time and efforts in parameter elicitation from experts.

5.2.3 An experiment testing priors from experts

An ideal experiment testing this approach would require elicitation of parameters from experts for a handful of networks. This would prove highly labor intensive and possibly beyond the scope of time expected for completing a doctoral dissertation. I am, therefore, proposing a simpler experiment that simulates this situation. This can be achieved by using the Bayesian Search algorithm (Cooper and Herskovits, 1992) to create a structure of a Bayesian network and the EM algorithm for parameter learning. To simulate that the qualification of this network may be coming from an expert, I make the numerical probabilities less precise, which one might expect from a human expert. To that effect, I apply a generic stationary rounding algorithm (Heinrich et al., 2005) to each numerical parameter of the network. I describe the details of this procedure in Part I in Experiment 1.

Experiment 1. ***Testing priors from experts in enhancing parameter learning***

Part I: Creating an expert-simulated Bayesian network

The experiment consisted of the following steps:

1. From the preprocessed data set D_i with the total number of non-missing records n, learn a Bayesian network, N_{BS} using the Bayesian Search algorithm (with 200 iterations) for structure learning and the EM algorithm for parameter learning. n must be at least 5,000 records to ensure good quality structures of a Bayesian network.
2. From the same data set D_i, learn a Bayesian network N_{TAN} using the TAN algorithm.
3. Use N_{TAN} to create a 10-time larger data set D_L.
4. Relearn parameters of N_{BS} using the EM algorithm using data set D_L with randomized initial parameters. As a result, we have a Bayesian network with parameters learned from the larger datasets, N^p_{BS}.
5. Use a generic stationary rounding algorithm to round all probabilities of the Bayesian network N^p_{BS}: stationary parameter $q = 0.5$, accuracy $n = 5$, and a global multiplier $v = n = 5$. As a result, we have a Bayesian network with less precise parameters, N_{expert}.

Steps 2-3 help preventing uniform distribution in a Bayesian network resulting from Step 1, while the rounding algorithm in Step 5 makes probabilities less precise, which one

might expect from a human expert. Then, I used the resulting simulated expert Bayesian networks, N_{expert}, in Part II to investigate the effect of priors.

Part II: Applying an expert-based Bayesian network as priors for parameter learning
The experiment for testing priors from experts consisted of the following steps:

1. From the preprocessed data set D_i with the total number of non-missing records n, randomly select subsets of data sets with a small number of records (300, 500, and 1000 records) as training sets: $Tr(D_{i300})$, $Tr(D_{i500})$ and $Tr(D_{i1000})$. The remaining data of each subset are used for testing sets: $Tt(D_{i300})$, $Tt(D_{i500})$ and $Tt(D_{i1000})$ respectively.
2. Relearn parameters of N_{expert} from Part I using the EM algorithm using each training data set ($Tr(D_{i300})$, $Tr(D_{i500})$ and $Tr(D_{i1000})$) with uniform initial parameter, i.e., disregarding existing parameters and learn new parameters from given data sets. As a result, we have three Bayesian network learned from different small data sets: $N_{expert300}$, $N_{expert500}$ and $N_{expert1000}$.
3. Validate $N_{expert300}$, $N_{expert500}$ and $N_{expert1000}$ on their corresponding testing set and record their accuracy.
4. Relearn parameters of N_{expert} from Part I using the EM algorithm using each training data set ($Tr(D_{i300})$, $Tr(D_{i500})$ and $Tr(D_{i1000})$) with the original parameters as priors from experts. As a result, we have another set of three Bayesian networks: $N^p_{expert300}$, $N^p_{expert500}$ and $N^p_{expert1000}$.
5. Validate $N^p_{expert300}$, $N^p_{expert500}$ and $N^p_{expert1000}$ on their corresponding testing set and record their accuracy.
6. Compare accuracies obtained from Step 3 against Step 5.

In this experiment, I used three large data sets (with more than 5,000 records) from Table 7: Adult, Mushroom and Nursery. Table 8 shows the result from Experiment 1. For the Adult and Nursery data sets, all Bayesian network models show accuracy improvement between 0.4% and 8% after enhancing with priors from experts, while Bayesian networks learning from the Mushroom data sets show almost no improvement. However, the accuracy of the Mushroom Bayesian network models is very high (99%) and it is quite a challenge to further improve it.

Table 8: Accuracy improvement of Bayesian networks with priors from experts

		ACC		Difference on ACC	
Model	**#Records**	**learned from data**	**After enhanced parameter learning**	**Absolute**	**Relative**
Adult	300	71.0	78.7	7.7	10.8%
	500	78.2	80.9	2.7	3.5%
	1000	79.9	81.5	1.6	2.0%
Mushroom	300	98.8	99.8	1.0	1.0%
	500	99.6	99.7	0.1	0.1%
	1000	99.7	99.7	0.0	0.0%
Nursery	300	82.8	90.9	8.1	9.8%
	500	87.4	91.5	4.1	4.7%
	1000	92.7	93.1	0.4	0.4%

5.3 Simplified probabilistic model as the sources of priors

Another way to enhance Bayesian network parameters in to use a simplified probabilistic model, e.g., Tree-Augmented Naïve Bayes model, as the source of priors. The TAN algorithm constraints the maximum number of incoming arcs to two and, by this, keeps the conditional probability tables (CPTs) in individual nodes small. Small CPTs mean a small number of parameters, which can be learned reliably even when the learning data set is small. Effectively, when the learning data set is small, the quality of the parameters remains high and the entire TAN model may match reasonably well the joint probability distribution that generated the data, even though TAN models does not mimic the causal structure of interactions among the model variables. In this section, I proposed the way of using such simplified models to obtain priors for parameter learning in a Bayesian network.

5.3.1 Methodology

In this experiment, I use all data sets listed in Table 7. For the UCI data sets, I created initial Bayesian network models using the Bayesian Search learning algorithm: one network per one data set. The Bayesian Search algorithm does not handle missing values and con-

tinuous variables. Hence, I preprocessed each data set in the same way by removing all records with missing values and excluding continuous variables. With the criteria mentioned in the previous section, Bayesian network models still have good quality structures. For the sources of priors, I used Bayesian network models learning from a Tree-Augmented Naïve Bayes (TAN) learning algorithm. The parameters captured in the TAN model will serve to generate a larger data set. This larger data set will serve to enhance the quality of the parameters in the initial Bayesian network model.

Experiment 2. ***Using a simplified probabilistic model to generate priors for parameter learning***

For the UCI data sets, the experiment consisted of the following steps:

1. From the preprocessed data set D_i with the total number of non-missing records n, learn a Bayesian network, N_{BS} using the Bayesian Search algorithm (with 200 iterations) for structure learning and the EM algorithm for parameter learning.
2. Validate N_{BS} with D_i using 10-fold cross validation. Record ACC(N_{BS}).
3. From the preprocessed data set D_i in Step 1, learn a Bayesian network N_{TAN} using the TAN algorithm.
4. Validate N_{TAN} with D_i using 10-fold cross validation. Record ACC(N_{TAN}).
5. Use N_{TAN} to create a 10-time larger data set D_L.
6. Relearn parameters of N_{BS} using the EM algorithm using data set D_L with randomized initial parameters. As a result, we have a Bayesian network with parameters learned from the larger dataset, N^p_{BS}.
7. Validate N^p_{BS} with D_i using 10-fold cross validation with different confidence level: 1, 10 and 100. Record ACC(N^p_{BS}). Compare ACC(N_{BS}) and ACC(N^p_{BS}).

I applied a similar approach to the *Artificial REVEAL* data set. In this case, we obtained six Bayesian network structure (labeled as E01:REVEAL to E06:REVEAL) from medical experts. The structure of these networks represent causal relationship between variables in the data set. I used the EM algorithm for parameter learning and used the TAN REVEAL 2.0 model created from the original REVEAL data set as a source of priors. I generated a

larger data set,i.e., 25,000 records, from the TAN model, and relearned parameters of each expert-based Bayesian network from the generated data set.

For model comparison, I used the classification accuracy (ACC) to measure model performance of the original Bayesian network models and the Bayesian network models enhanced with priors. I used 10-fold cross validation method implemented in GeNIe for model validation. In this case, I reported the result of 10-fold cross-validation with different level of confidence (Conf.): 1, 10, 100. The confidence level represents an equivalent sample size (ESS), i.e., the number of records that the parameters in the network are based on. Low confidence, i.e., 1, means that even a small amount of data can easily change the probability distribution in the network. This allows me to evaluate the optimal confidence for the experiment.

5.3.2 Result

Table 9 reports the model accuracy (ACC) of the original model and the accuracy of the model after parameter enhancement by priors. There are no significant differences in accuracy between confidence levels of 1, 10 and 100 for 10-fold cross validation with the EM algorithm. However, confidence equal to 100 seems to be the best of the three in terms of the percentage of improvement in accuracy.

Figure 23 shows the percentage improvement in accuracy for each Bayesian network after enhancing with priors. I only plotted the result of validation with confidence 100. Table 10 reports the remaining result. Majority of the data sets show slight improvement (between 0% and 5%.) Two data sets (E02: REVEAL and E06: REVEAL), however, show an improvement in accuracy of 22.5% and 16.2% respectively.

5.3.3 Discussion

The approach to enhance the Bayesian network accuracy by means of priors from the TAN model is by far most effective, especially when we are dealing with a complex Bayesian network and a small data set. As Table 10 shows, accuracy of Bayesian networks with large numbers of parameters benefits from the methods, while accuracy of Bayesian networks

Table 9: Performance of Bayesian network with priors from a simplified probabilistic model

Model	**Original ACC**	**TAN (%) ACC (%)**	**With Prior Conf.=1 (%)**	**With Prior Conf.=10 (%)**	**With Prior Conf.=100 (%)**
D01:Adult	81.6	82.1	81.7	81.7	81.7
D02:Breast Cancer	75.1	75.1	75.5	75.8	75.8
D03:Credit Approval	73.7	85.2	74.3	74.6	75.1
D04:Flag	69.1	62.4	69.6	70.6	71.1
D05:German Credit	73.1	72.0	73.1	73.1	73.6
D06:Lymphography	83.8	85.1	83.1	83.1	85.8
D07:Mushroom	98.5	99.8	98.7	98.7	98.7
D08:Nursery	94.3	93.4	94.4	94.4	94.6
E01:REVEAL	85.1	89.3	87.9	88.4	88.5
E02:REVEAL	71.1	89.3	93.5	93.7	93.6
E03:REVEAL	89.4	89.3	89.4	89.5	89.5
E04:REVEAL	87.3	89.3	90.3	90.5	90.5
E05:REVEAL	86.9	89.3	89.5	89.7	89.8
E06:REVEAL	76.7	89.3	92.5	92.9	92.9

Figure 23: Percentage improvement of accuracy in enhanced Bayesian network

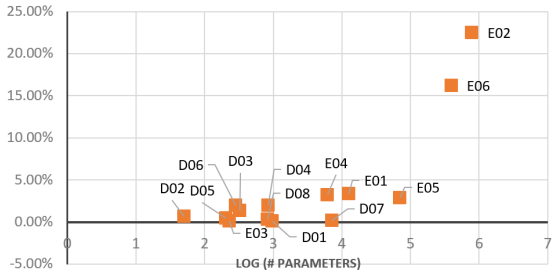
Table 10: Percentage change in accuracy of Bayesian network models with parameter learning enhanced with priors from a simplified probabilistic model

Model	**#Traing**	**#Parameter**	**With Prior Conf.=1 (%)**	**With Prior Conf.=10 (%)**	**With Prior Conf.=100 (%)**
D01	30162	78	0.1	0.1	0.1
D02	277	50	0.4	0.7	0.7
D03	653	325	0.6	0.9	1.4
D04	194	1074	0.5	1.5	2.0
D05	1000	203	0.0	0.0	0.5
D06	148	282	-0.7	-0.7	2.0
D07	5644	7236	0.2	0.2	0.2
D08	12960	834	0.0	0.0	0.2
E01	2500	12582	2.8	3.3	3.4
E02	2500	786756	22.4	22.6	22.5
E03	2500	228	0.0	0.1	0.1
E04	2500	6203	3.0	3.2	12.6
E05	2500	70207	2.6	2.8	2.9
E06	2500	393360	15.8	16.2	16.2

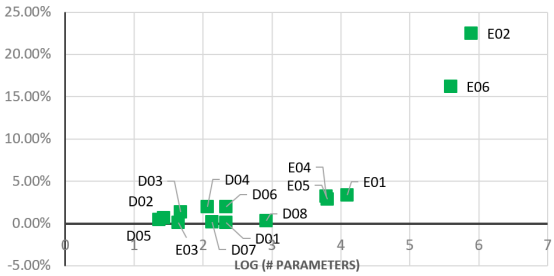
with small number of parameters improves slightly. When the data sets are large, parameter learning also benefits from the methods, although just slightly.

I tested different factors that could possibly allow us to a-priori predict the accuracy improvement of the network, including number of parameters, number of independent parameters, maximum indegree of the network (maximum number of parents of a node) and maximum number of column in a CPT. Figure 24 shows a list of scatterplots showing relationships between those factors (x-axis) against the improvement of accuracy after enhancing parameters with priors for both, Bayesian networks (Figure 24a, 24c, 24e and 24g) and the class node's Markov Blanket (Figure 24b, 24d, 24f and 24h). The Markov blanket of a random variable, X_i, consists of variables that are parents, children, and parents of its children, such that, when observed, make X_i independent of the remainder variables in the network (Pearl, 1988). In other words, Markov blanket of a *class* node is a simpler version of a Bayesian network.

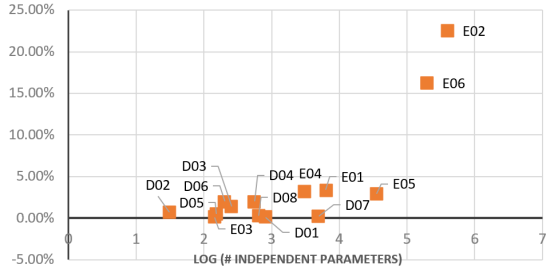
As we expected, complex network having large number of parameters/independent pa-



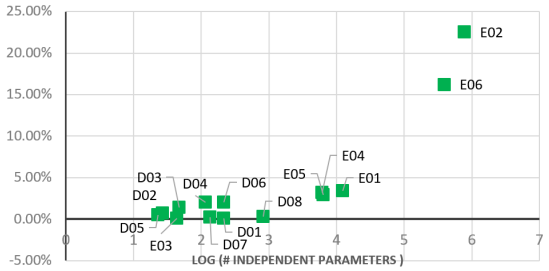
(a) Log_{10}(#Parameters): Bayesian network



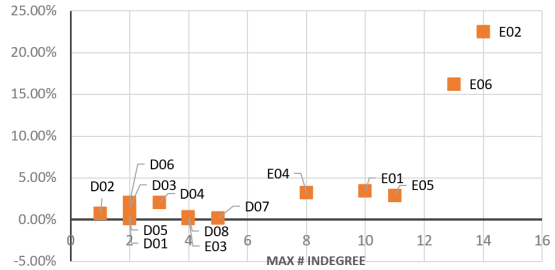
(b) Log_{10}(#Parameters): Markov blanket



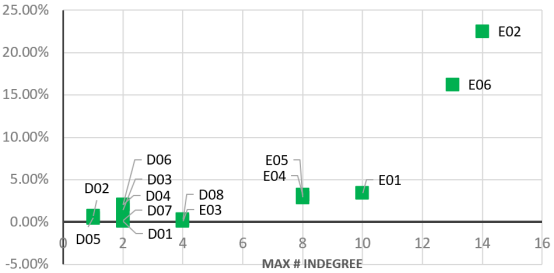
(c) Log_{10}(#Indp. parameters): Bayesian network



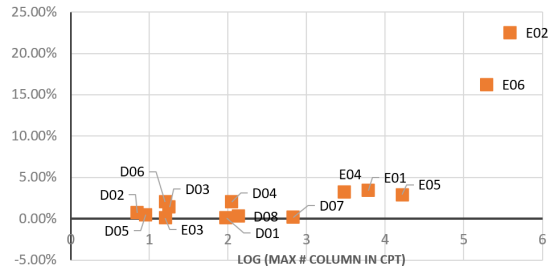
(d) Log_{10}(#Indp. parameters): Markov blanket



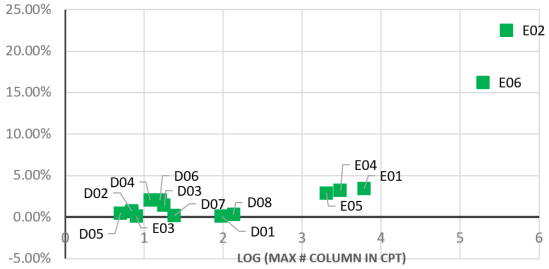
(e) Max #Indegree: Bayesian network



(f) Max #Indegree: Markov blanket



(g) Log_{10}(Max#Col. CPT): Bayesian network



(h) Log_{10}(Max#Col. CPT): Markov blanket

Figure 24: Effect of each network parameter to accuracy improvement after parameter enhancement from priors

rameters, such as E02 and E06, benefit from the proposed methods most, while simple Bayesian network show almost no improvement in accuracy after enhancement. I observed the similar trend with maximum number of indegree and maximum number column of CPT, which also represents the degree of complexity in a network. There are not many differences in the results from Bayesian networks and Markov blankets. Our explanation fro this lack of difference is that in this experiment, I excluded all records with missing values for model building and validation, and variable in the Markov blanket were typically observed and screened the class nodes for the rest of the variables.

5.3.4 Potential of overfitting

Because the TAN networks used for priors were trained on the same data set, it is possible that the resulting Bayesian networks overfitted to the training data sets. I conducted an experiment to investigate the degree of overfitting. I selected the top three Bayesian network models with the most improvement in accuracy after enhancing with TAN networks.

Experiment 3. ***Testing overfitting of parameter learning***

The experiment consisted of the following steps:

1. From the selected data set D_i with total number of complete records n, randomly assign each record into a training set $Tr(D_i)$ and a testing set $Tt(D_i)$. The training to testing ratio is 80 to 20.
2. From a Bayesian network N_i created in Experiment 2, use the EM algorithm for parameter learning with uniform initial parameters, i.e., disregard the existing parameters and learn new parameters from a training set $Tr(D_i)$.
3. Validate N_i with a testing set $Tt(D_i)$. Record ACC(N_{BS}).
4. From the training set $Tr(D_i)$, learn a Bayesian network N_{TAN} using the TAN algorithm.
5. Validate N_{TAN} on $Tt(D_i)$. Record ACC(N_{TAN}).
6. Use N_{TAN} to create a 10-time larger training data set of $Tr(D_{ix10}$.
7. Relearn parameters of N_i using the EM algorithm using data set $Tr(D_{ix10}$. As a result, we have a Bayesian network learned from the larger dataset, N_i^p.

Table 11: Accuracy improvement of Bayesian networks after parameter enhancement: potential overfitting

	Training		All records		Difference on Improved ACC	
Model	**ACC**	**Improved ACC**	**ACC**	**Improved ACC**	**Absolute**	**Relative**
E01	85.2	88.2	85.1	88.5	-0.30	-0.34%
E02	70.6	86.6	71.1	93.6	-7.00	-8.08%
E06	74.3	76.4	76.7	92.9	-4.70	-5.33%

8. Validate N_i^p on $Tt(D_i)$. Record ACC(N_i^p) and compare it to the accuracy of Bayesian networks in Experiment 2.

Table 11 shows the result of this experiment. For the sets of Bayesian networks learning from a training subset (80%) of data, I reported the improvement in accuracy validated on the testing set (20%) of data sets along with their accuracy improvement of Bayesian network models from Experiment 2 by mean of 10-fold cross validation with confidence 100. The last two columns report the absolute and relative improvement that may stem from overfitting. The result is conservative, as some of that improvement is due to a larger training set (100% vs. 80% of records). Even if some overfitting is taking place, priors still enhance parameter learning significantly.

6.0 Discussion and future work

One of the most prevalent methods for risk assessment is the CPH model The weaknesses of this approach are: (1) the underlying model can be only learned from data and is not readily amenable to refinement based on expert knowledge, and (2) the CPH model rests on several assumptions simplifying the interactions between the risk factors and the predicted outcome. While the CPH-based risk assessment models has been successfully used for decades, Bayesian networks offer more modeling flexibility and possibly superior performance.

The contributions of this dissertation demonstrate our effort to replace the CPH model underlying risk assessment by using Bayesian networks. I proposed a Bayesian network interpretation of the CPH (BN-Cox) model, which use the CPH models as data sources in the process of parameter estimation for Bayesian networks. I successfully replaced the use of the CPH model in the REVEAL risk score calculator (Benza et al., 2010) with an BN-Cox-based risk score calculator, and hence, offered precisely the same accuracy. I studied two approaches to mitigate an exponential growth of conditional probability table in BN-Cox model: (1) decomposition of the underlying Bayesian network or parent divorcing, and (2) simplifying the network structure by removing least influential risk factors. The BN-Cox model is not decomposable and approximating of decomposition leads to high loss of accuracy. Hence, simplifying the network structure by removing the least influential risk factors by any statistical variable selection methods was recommenced, when we have a data set to refit the simplified model. However, when data are not available, we can simplify the model by removing or marginalizing least influential risk factors based on both the value of β coefficients and the statistical significance.

I demonstrated the unrealistic assumptions of the CPH model in practice. When performing prediction for estimating the outcome probability, the strength of influence of risk factors to an outcome variable in the CPH model do not change regardless of the context of other risk factors. I empirically demonstrated the influence of risk factors in the CPH-based model. CPH model do not model correctly varying magnitude of influence of risk factors as

other factors are observed.

I discussed methods for enhancing the quality of Bayesian network parameters, as learned from small data sets, by means of different priors: priors from expert knowledge and priors from simplified probabilistic models such as Tree-Augmented Naïve Bayes. I provided on empirical evaluation of the proposed methods and demonstrated that they improve quality of parameters and accuracy for Bayesian network in risk assessment on several data sets. I investigated different factors of the network related to the improvement of accuracy. Complex Bayesian networks, i.e., those with large numbers of parameters, max indegree and CPTs, benefits from the proposed methods most, while simple Bayesian networks show almost little or no improvement in accuracy. It seems that enhancing parameter learning with the TAN networks generated some overfitting but still led to significant improvement in accuracy.

One direction of future work would be to extend the experiments for parameter enhancement methods in Chapter 5 to be more comprehensive. For example, (1) using a real expert-based Bayesian networks in Section 5.2.3, (2) providing an experiment on parameter enhancement based on canonical gates, and (3) propose a better parameter enhancement from the TAN network that minimize overfitting.

Bibliography

Allison, P. D. (2010). *Survival Analysis Using SAS: A Practical Guide, Second Edition*. SAS Institute Inc., Cary, NA.

Benza, R. L., Gomberg-Maitland, M., Elliott, C. G., Farber, H. W., Foreman, A. J., Frost, A. E., McGoon, M. D., Pasta, D. J., Selej, M., Burger, C. D., and Frantz, R. P. (2019). Predicting survival in patients with pulmonary arterial hypertension: The REVEAL risk score calculator 2.0 and comparison with ESC/ERS-based risk assessment strategies. *Chest*.

Benza, R. L., Gomberg-Maitland, M., Miller, D. P., Frost, A., Frantz, R. P., Foreman, A. J., Badesch, D. B., and McGoon, M. D. (2012). The REVEAL registry risk score calculator in patients newly diagnosed with pulmonary arterial hypertension. *Chest*, 141(2):354–362.

Benza, R. L., Miller, D. P., Gomberg-Maitland, M., Frantz, R. P., Foreman, A. J., Coffey, C. S., Frost, A., Barst, R. J., Badesch, D. B., Elliott, C. G., Liou, T. G., and McGoon, M. D. (2010). Predicting survival in pulmonary arterial hypertension: Insights from the Registry to Evaluate Early and Long-Term Pulmonary Arterial Hypertension Disease Management (REVEAL). *Circulation*, 122(2):164–172.

Cooper, G. F. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347.

Covello, V. T. and Merkhoher, M. W. (2013). *Risk assessment methods: approaches for assessing health and environmental risks*. Springer Science & Business Media.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Díez, F. J. and Druzdzel, M. J. (2006). Canonical probabilistic models for knowledge engineering. Technical report, UNED, Madrid, Spain.

Díez, F. J. and Galan, S. F. (2003). Efficient computation for the noisy MAX. *International Journal of Intelligent Systems*, 18(2):165–177.

Druzdzel, M. J. and van der Gaag, L. C. (2000). Building probabilistic networks: "where do the numbers come from?". *IEEE Transactions on Knowledge and Data Engineering*, 12(4):481–486.

Dua, D. and Karra Taniskidou, E. (2019). UCI machine learning repository.

Fan, J. and Li, R. (2002). Variable selection for Cox's proportional hazards model and frailty model. *Annals of Statistics*, 30(1):74–99.

Fox, J. (2002). *An R and S-Plus Companion to Applied Regression*. Sage Publication Inc., CA.

Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, 29(2-3):131–163.

Heinrich, L., Pukelsheim, F., and Schwingenschlögl, U. (2005). On stationary multiplier methods for the rounding of probabilities and the limiting law of the Sainte-Laguë divergence. *Statistics & Decisions*, 23(2/2005):117–129.

Husmeier, D., Dybowski, R., and Roberts, S. (2005). *Probabilistic Modeling in Bioinformatics and Medical Informatics*. Springer.

Kanwar, M. K., Lohmueller, L. C., Kormos, R. L., Teuteberg, J. J., Rogers, J. G., Lindenfeld, J., Bailey, S. H., McIlvennan, C. K., Benza, R., Murali, S., et al. (2018). A Bayesian model to predict survival after left ventricular assist device implantation. *JACC: Heart Failure*, 6(9):771–779.

Kaplan, E. L. and Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):457–481.

Kirklin, J. K., Pagani, F. D., Kormos, R. L., Stevenson, L. W., Blume, E. D., Myers, S. L., Miller, M. A., Baldwin, J. T., Young, J. B., and Naftel, D. C. (2017). Eighth annual INTERMACS report: special focus on framing the impact of adverse events. *The Journal of Heart and Lung Transplantation*, 36(10):1080–1086.

Klein, J. P. and Moeschberger, M. L. (2003). *Survival Analysis: Censored and Truncated Data, Second Edition*. Springer-Verlag New York, Inc, New York, NY.

Kraisangka, J. and Druzdzel, M. J. (2014). Discrete Bayesian network interpretation of the Cox's proportional hazards model. In van der Gaag, L. C. and Feelders, A. J., editors, *Probabilistic Graphical Models*, volume 8754 of *Lecture Notes in Computer Science*, pages 238–253. Springer International Publishing.

Kraisangka, J. and Druzdzel, M. J. (2016). Making large Cox's proportional hazard models tractable in Bayesian networks. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, pages 252–263.

Kraisangka, J. and Druzdzel, M. J. (2018). A Bayesian network interpretation of the Cox's proportional hazard model. *International Journal of Approximate Reasoning*, 103:195–211.

Kraisangka, J., Druzdzel, M. J., and Benza, R. L. (2016). A risk calculator for the pulmonary arterial hypertension based on a Bayesian network. In *Proceedings of the 13th UAI Bayesian Modeling Applications Workshop*, pages 1–6.

Lauritzen, S. L. (1995). The EM algorithm for graphical association models with missing data. *Computational Statistics & Data Analysis*, 19(2):191–201.

Levy, W. C., Mozaffarian, D., Linker, D. T., Sutradhar, S. C., Anker, S. D., Croppan, A. B., Anand, I., Maggionl, A., Burton, P., Sullivan, M. D., Pitt, B., Poole-Wilson, P. A., Mann, D. L., and Packer, M. (2006). The Seattle Heart Failure Model: Prediction of survival in heart failure. *Circulation*, 113(11):1424–1433.

Lewin-Koh, N. (2018). *Hexagon Binning: an Overview*.

Lucas, P. J., van der Gaag, L. C., and Abu-Hanna, A. (2004). Bayesian networks in biomedicine and health-care. *Artificial intelligence in medicine*, 30(3):201–214.

Nowak, K. and Druzdzel, M. J. (2014). Learning parameters in canonical models using weighted least squares. In van der Gaag, L. C. and Feelders, A. J., editors, *Probabilistic Graphical Models, Springer Lecture Notes in Computer Science, Vol. 8754*, pages 366–381, Heidelberg. Springer International Publishing.

Oniśko, A., Druzdzel, M. J., and Wasyluk, H. (2001). Learning Bayesian network parameters from small data sets: Application of Noisy-OR gates. *International Journal of Approximate Reasoning*, 27(2):165–182.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Rossi, P. H., Berk, R. A., and Lenihan, K. J. (1980). *Money, Work, and Crime – Experimental Evidence*. Academic Press, Inc., San Diego, CA.

Savicky, P. and Vomlel, J. (2007). Exploiting tensor rank-one decomposition in probabilistic inference. *Kybernetika (Special issue dedicated to the memory of Albert Perez)*, 43(5):747–764.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423.

van der Gaag, L. C., Renooij, S., Witteman, C. L., Aleman, B. M., and Taal, B. G. (1999). How to elicit many probabilities. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 647–654. Morgan Kaufmann Publishers Inc.

van Gerven, M. A., Taal, B. G., and Lucas, P. J. (2008). Dynamic Bayesian networks as prognostic models for clinical patient management. *Journal of Biomedical Informatics*, 41(4):515–529.

Vomlel, J. and Tichavsky, P. (2014). Probabilistic inference with noisy-threshold models based on a CP tensor decomposition. *International Journal of Approximate Reasoning*, 55(4):1072–1092.

Zagorecki, A. and Druzdzel, M. J. (2013). Knowledge engineering for Bayesian networks: How common are noisy-max distributions in practice? *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(1):186–195.

Zagorecki, A., Voortman, M., and Druzdzel, M. J. (2006). Decomposing local probability distributions in Bayesian networks for improved inference and parameter learning. In Sutcliffe, G. and Goebel, R., editors, *Recent Advances in Artificial Intelligence: Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS–2006)*, pages 860–865, Menlo Park, CA. AAAI Press.