An Empirical Study of Bayesian Network Parameter Learning with Monotonic Influence Constraints

Yun Zhou[a,b,*], Norman Fenton[a], Cheng Zhu[b]

[a] *Risk and Information Management (RIM) Research Group, Queen Mary University of London*
[b] *Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology*

Abstract

Learning the conditional probability table (CPT) parameters of Bayesian networks (BNs) is a key challenge in real-world decision support applications, especially when there are limited data available. A conventional way to address this challenge is to introduce domain knowledge/expert judgments that are encoded as qualitative parameter constraints. In this paper we focus on a class of constraints which is naturally encoded in the edges of BNs with monotonic influences. Experimental results indicate that such monotonic influence constraints are widespread in practical BNs (all BNs used in the study contain such monotonic influences). To exploit expert knowledge about such constraints we have developed an improved constrained optimization algorithm, which achieves good parameter learning performance using these constraints, especially when data are limited. Specifically, this algorithm outperforms the previous state-of-the-art and is also robust to errors in labelling the monotonic influences. The method is applied to a real world medical decision support BN where we had access to expert-provided constraints and real hospital data. The results suggest that incorporating expert judgments about monotonic influence constraints can lead to more accurate BNs for decision support and risk analysis.

Key words: BN parameter learning; Monotonic influences; Exterior con-

[*] Corresponding author

Email addresses: zhouy.nudt@gmail.com (Yun Zhou), n.fenton@qmul.ac.uk (Norman Fenton), zhucheng@nudt.edu.cn (Cheng Zhu)

Preprint submitted to Decision Support Systems *May 9, 2016* straints; Experiments on publicly available BNs; Real medical study.

1. Introduction

Bayesian networks (BNs) have become increasingly popular in the AI field during the last two decades because of their ability to model probabilistic dependent relationships among variables in many real-world problems. A BN model consists of two components: a network structure and a set of conditional probability tables (CPTs) whose entries are considered as parameters.

In real-world decision support problems that we wish to model as BNs, there are typically limited or no relevant data. In such situations attempts to learn BN structures purely from data are unlikely to result in useful models. For example, even 500 data points (which in many real-world situations is a very large sample) is nowhere near enough to learn the structure of a very small BN such as the well-known Asia BN that has just 8 nodes and 8 edges in total. Using the pure data based structure learning algorithm (Eaton and Murphy, 2007) in this example results in more than half of the learnt edges being different from the ground truth. The scarce data problem is typical of many real-world decision support problems in which we have nevertheless used BN models effectively, by exploiting expert domain knowledge. Specifically, the decision support problems addressed include:

- determine whether or not to provide a specific type of intervention for a given psychiatric patient (Constantinou et al., 2016).
- determine whether or not a limb should be amputated given a patient's specific pathology (Yet et al., 2014).
- determine whether a given prisoner with a background of violence can be safely released into the community (Constantinou et al., 2015b).
- determine which of two alternative medical tests optimises the balance between accuracy, safety and cost (Fenton and Neil, 2010).

- determine how and when to place bets on football matches to 'beat the bookmakers' (Constantinou et al., 2013).

In all of these problems limited data were available (both in terms of size of data and complete absence of data for some key variables), but we had access to relevant domain experts who were able to provide the BN structure (including causal relationships involving unobserved variables) and insights into the conditional probability table (CPT) parameters where there was little or no data. However, although there has been some progress in attempts to make more systematic the methods helping experts define BN structures (see, e.g. Constantinou et al. (2016) and Fenton et al. (2013) the process for fully defining the CPTs (i.e. eliciting the BN parameters) from experts has been largely ad hoc. The objective of this paper is to demonstrate a more systematic and rigorous approach to combining expert knowledge and data to achieve more accurate parameter elicitation in such models. Hence, to clarify the scope, the paper is focused on the following common scenario:

> *A BN structure has been hand-crafted by domain experts to model a real-world decision support problem. A small amount of data relevant to the model is available. The challenge is to build the model parameters by combining the limited data with domain knowledge about the parameters.*

In addition to the examples described above an increasing number of decision support problems (medical, financial and safety) fit with this scenario (Fenton and Neil, 2012), and so there is a genuine demand for improved solutions. It is also important to note that, because we are restricting our discussion to BNs whose structures have been hand-crafted by experts, the scope is limited to relatively 'small' BNs (generally expert defined BNs with fewer than 100 nodes), although our experiments do include some larger BNs.

The simplest parameter learning approach is maximum likelihood estimation (MLE). However, this method usually fails to find good estimates for parameters with few data points (in some complex BNs, there is an explosion of variable state configurations, we might not have enough training data in some specific variable state configurations even in cases where big-data is available). To address this researchers developed the maximum a posteriori probability (MAP) approach by introducing a Dirichlet parameter prior, which we discuss in Section 2. However, as we also discuss in Section 2, experts tend to feel more comfortable providing qualitative or semi-numerical judgments with less cognitive effort. Such judgments are expressed as constraints between parameters of interest, and are more easily elicited from experts than corresponding point-wise estimates.

In this paper we focus on an important class of such constraints elicited from monotonic influences (also known as qualitative influences (Wellman, 1990) or qualitative monotonicities (Altendorf et al., 2005)), which are naturally encoded in the edges/structures of BNs. A monotonic influence is one where the increase (or decrease) of one variable will monotonically change the value of another variable. This kind of influence can be directly elicited from the BN structures, and can be easily converted into associated parameter constraints, which we refer to as monotonic influence constraints. These constraints are exterior parameter constraints (relations between parameters from different conditional distributions). For a simple example, in “Smoke → Cancer”, it is widely accepted that people who smoke have a higher risk of getting cancer than those who do not. Thus:

$P(Cancer = true|Smoke = true) \geq P(Cancer = true|Smoke = flase)$

is an example of a monotonic influence constraint.

When the training data is limited, incorporating such exterior constraints from experts could help the BN parameter learning. In this paper, we investigate the extent to which such monotonic influences and their generated exterior constraints are present in a set of real-world BNs, and provide a simple improved constrained optimization algorithm for parameter estimation with these constraints.

The paper is organized as follows. In Section 2 we discuss related work in BN parameter learning with limited data. In Section 3 we introduce the BN parameter learning notation to be used throughout this paper. In Section 4 we describe the monotonic influences and the improved parameter learning method. In Section 5 we report on the experiments of 12 different real-world BNs. In Section 6 we present the results of applying the method to a real world medical decision support BN. Our conclusions are in Section 7.

2. Related Works

There are several methods for handling parameterization with limited or no relevant data, described in a rich literature of books, articles and software packages, which are briefly summarized in (Druzdel and Van Der Gaag, 2000; Neapolitan, 2004; O'Hagan et al., 2006; Fenton and Neil, 2012). Of these, expert knowledge/judgments are widely used in real-world BN construction (de Campos et al., 2008; Baumgartner et al., 2008; Constantinou et al., 2013), especially in medical decision support applications (Lucas et al., 2004; Hutchinson et al., 2009; Flores et al., 2011; Yet et al., 2013, 2014; Constantinou et al., 2015a).

However, expert elicitation is expensive, time-consuming and sometimes error-prone (Neil et al., 2000), because the number of parameters increase exponentially with the number of nodes in the BN. Therefore, the challenge has mainly been addressed using methods that minimize the number of elicited parameters. The Noisy-OR (Diez, 1993) and Noisy-MAX (Pradhan et al., 1994) are examples of methods to reduce the number of elicited parameters, based on the independence of causal influences (ICI) assumption (Zagorecki and Druzdzel, 2013). Extensions of these models include the Ranked Node (Fenton et al., 2007) and NIN-AND tree (Xiang and Jia, 2007; Xiang and Truong, 2014) models.

To address the problem that some parameters have zero observations in limited training data, a Dirichlet parameter prior is introduced for them. Experts are required to provide Dirichlet hyperparameters. In the BDeu prior, experts are only needed to provide the equivalent sample size parameter (Heckerman et al., 1995). Guidance in choosing the value of equivalent sample size is well studied (Silander et al., 2007). However, elicited hyperparameters of ICI models and Dirichlet distributions are both numerical, which means they are quantitative knowledge. Previous work has shown that eliciting qualitative or semi-numerical judgments is easier than collecting numerical values of CPTs (Helsper et al., 2005). Parameter constraint (Druzdzel and van der Gaag, 1995) is an important class of such qualitative judgments. For example, the statement "the probability of people getting cancer is very low" is such a parameter constraint.

Several models have been proposed to integrate parameter constraints and improve the learning accuracy. The most popular is the constrained convex optimization (CO) formulation (Niculescu et al., 2006; de Campos and Ji, 2008; de Campos et al., 2008; Liao and Ji, 2009; de Campos et al., 2009). These algorithms seek the global optimal estimation (maximal log likelihood) with respect to the parameter constraints. The parameters also can be estimated by the Monte Carlo method (Chang et al., 2008), where only the samples that consist of the constraints are kept. Recently, auxiliary BN models (Zhou et al., 2013, 2014a, 2015) have been developed for solving this problem. In this approach, the target parameters, data observations and elicited constraints are all modelled as nodes in the auxiliary BNs. Thus, the parameters are estimated via the inference in the auxiliary BNs. However, constraints discussed in these models are not elicited from qualitative monotonic influences, and are usually expensive to elicit.

An alternative approach to reducing the burden of expert elicitation is to find monotonic influences in some edges of BNs, and use them to generate exterior parameter constraints. BNs that are fully specified by monotonic influences are referred to as Qualitative Probabilistic Networks (QPNs) (Wellman, 1990). An efficient sign-propagation algorithm is achieved by restricting the maximal number of node-sign changes during the inference (Druzdzel and Henrion, 1993; Renooij and Van der Gaag, 2008). The inference results answer the question of how observations of some variables change the probability distributions of other variables. The combination of QPNs and BNs is referred to as Semi-Qualitative Probabilistic Networks (SQPNs) (Renooij and van der Gaag, 2002), which means parts of the variables are represented by joint probability tables rather than qualitative influences. Inference and learning in SQPNs is discussed in later work (de Campos and Cozman, 2005).

As in previous work (Altendorf et al., 2005; Feelders and van der Gaag, 2006; van der Gaag et al., 2006; Feelders, 2007), in this paper, we only use signs of qualitative probabilistic networks and their generated monotonic influence constraints to constrain the probabilities in the standard BN parameter learning. Thus, experts are only required to identify which edges in the BN have such qualitative monotonicity property. Cano et al. (2011) and Masegosa and Moral (2013) proposed an interactive BN structure learning approach that iteratively queries the domain expert about the reliability of learnt edges. This interactive paradigm can be easily applied to help elicit edges' monotonic influences from experts and suggests that the theoretical method presented in this paper can be applied in practice (we do not use the interactive paradigm here because in our experiments we assume that any edges with monotonic influences are known and we simulate errors made by experts).

More discussions about parameter learning with exterior constraints generated from monotonic influences can be found in (van der Gaag et al., 2009; Yang and Natarajan, 2013; Zhou et al., 2014b). Although parameter learning with these constraints has been well studied, there is no empirical analysis of the extent to which such monotonic influences exist in real-world BNs. This paper addresses this research gap by investigating the qualitative monotonicity for each edge in a set of real-world BNs. Moreover, the learning performance of state-of-the-art CO algorithm and our simple improved version (which we refer to as COFP) in these BNs is also reported.

3. Bayesian Networks Parameter Learning

A BN consists of a directed acyclic graph (DAG) $G = (U, E)$ (whose nodes $U = \{X_1, X_2, X_3, \ldots, X_n\}$ correspond to a set of random variables, and whose arcs E represent the direct dependencies between these variables), together with a set of probability distributions associated with each variable (Pearl, 1988).

For discrete variables[1] the probability distribution is described by a conditional probability table (CPT) that contains the probability of each value of the variable given each instantiation of its parent values in G. We write this as $P(X_i|\Pi_i)$ where Π_i denotes the set of parents of the variable X_i in DAG G. Thus, the BN defines a simplified joint probability distribution over U given by:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|\Pi_i) \tag{1}$$

Let θ denote a set of numerical parameters of the categorical random variables in some set U. Let r_i denote the cardinality of the space of X_i, and $|\Pi_i|$ represent the cardinality of the space of parent configurations of X_i. Let $P(X_i|\Pi_i = p)$ denote the discrete probability distribution of X_i given the p-th state configuration of its parents ($\Pi_i = p$). The k-th probability value of $P(X_i|\Pi_i = p)$ can be represented as θ_{ipk}, where $\theta_{ipk} \in \theta$, $1 \leq i \leq n$, $1 \leq p \leq |\Pi_i|$ and $1 \leq k \leq r_i$. Assuming $D = \{D_1, D_2, \ldots, D_N\}$ is a dataset of fully observable cases for a BN, then D_l is the l-th complete case of D, which is a vector of values of all variables in U.

The classical maximum likelihood estimation (MLE) (Barber, 2012) is to find the set of parameters that maximize the log likelihood $\ell(\theta|D) = \sum_l \log P(D_l|\theta)$. Let N_{ipk} be the number of data records in sample D for which X_i takes its k-th value and its parents Π_i take the p-th state configuration. Then $\ell(\theta|D)$ can be rewritten as $\ell(\theta|D) = \sum_{ipk} N_{ipk} \log \theta_{ipk}$. MLE seeks to estimate θ by maximizing $\ell(\theta|D)$. In particular, we can get the estimation of each parameter as follows:

$$\theta^*_{ipk} = \frac{N_{ipk}}{N_{ip}} \tag{2}$$

where $N_{ip} = \sum_{k=1}^{r_i} N_{ipk}$.

However, it is common (even for a large dataset) that certain parent-child state combinations seldom appear, and MLE fails in this situation. Hence,

[1] For continuous variables we normally refer to a conditional probability distribution.

another classical parameter learning algorithm (maximum a posteriori, MAP) is used to mediate this problem by introducing the Dirichlet prior:

$$\theta^* = \arg\max_{\theta} P(D|\theta)P(\theta) \tag{3}$$

Therefore, the MAP estimation of each parameter is:

$$\theta^*_{ipk} = \frac{N_{ipk} + \alpha_{ipk}}{N_{ip} + \alpha_{ip}} \tag{4}$$

Intuitively, one can think of the hyperparameter α_{ipk} in the Dirichlet prior as an experts' guess of the virtual data counts of the parameter θ_{ipk}. When there is no related expert judgments, people usually use a uniform/flat prior $\alpha_{ipk} = 1$ or BDeu prior $\alpha_{ipk} = \frac{1}{r_i|\Pi_i|}$ (likelihood equivalent uniform Bayesian Dirichlet) (Heckerman et al., 1995).

4. Parameter Learning with Monotonic Influence Constraints

This section provides a full formalism of how to translate a set of qualitative judgments into probability constraints. Here, we follow Wellman's approach (Wellman, 1990), where qualitative judgments involve monotonic influences between nodes. We will firstly discuss the exterior constraints and then give the formal definition of monotonic influences and their converted exterior constraints. After that, we will discuss how to solve the parameter estimation problem with such constraints. Finally, we will discuss the computational time complexity of the proposed learning algorithm.

4.1. The Exterior Constraint

Parameter constraints can be divided into two types according to the constrained parameters' parent state configurations: 1) interior constraint and 2) exterior constraint. The interior constraint restricts the node parameters within a CPT column (parameters that share the same parent state configuration). For example, an interior constraint could be "the probability of a patient getting cancer is smaller than 1%" in a medical BN. In Zhou et al. (2014a) we showed that significant improvements to CPT learning can be achieved from a relatively small number of expert provided interior constraints. However, in many situations it is possible (and actually more efficient) to elicit constraints between parameters with different parent state configurations. These constraints are referred to as exterior constraints, and defined as follows:

Definition 4.1. Exterior Constraints For any variable X_i in a BN, if the two associated parameters θ_{ipk} and θ_{iqk} in X_i have different parent state configurations $\Pi_i = p$ or $\Pi_i = q$ $(p \neq q)$, we call $\theta_{ipk} \geq \alpha + \beta\theta_{iqk}$ or $\theta_{ipk} < \alpha + \beta\theta_{iqk}$ (where $\alpha, \beta \in \mathbb{R}$ and $\beta \neq 0$) an *exterior constraint.*

This kind of constraint can be generated from monotonic influences which can greatly reduce the burden of expert judgment elicitation. Next, we will discuss the definition of monotonic influences.

4.2. The Monotonic Influences and Generated Exterior Constraints

Definition 4.2. Monotonic Influences For any dependent relationship $X_j \to X_i$ in a BN with ordered categorical variables, if an increase in X_j leads to an increase in X_i no matter the values of other variables in $\Pi_i \setminus \{X_j\}$, we call this a *positive* monotonic influence $X_j \overset{+}{\to} X_i$. Conversely, if an increase in X_j leads to a decrease in X_i no matter the values of other variables in $\Pi_i \setminus \{X_j\}$, we call this a *negative* monotonic influence $X_j \overset{-}{\to} X_i$.

A zero influence ($X_j \overset{0}{\to} X_i$) is defined analogously whereby an increase in X_j will not change the value of X_i. This influence is left implicit in the network's graphical representation. Finally, if there is no positive or negative monotonic influence between X_j and X_i, we call this *ambiguous* monotonic influence $X_j \overset{?}{\to} X_i$.

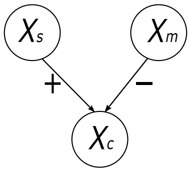

Figure 1: A three-node BN with two monotonic influences.

As we will show in Section 5, positive and negative monotonic influences occur widely in real-world BN applications. Based on the above definitions (and using smoking X_s, cancer X_c and medical treatment X_m as example variables shown in Figure 1), two monotonic influences in the BN example can be formulated as exterior constraints as follows:

$$\begin{aligned} X_s \overset{+}{\rightarrow} X_c &: F(x_c|\ X_s = false,\ x_m) \geq F(x_c|\ X_s = true,\ x_m) \\ X_m \overset{-}{\rightarrow} X_c &: F(x_c|\ x_s,\ X_m = false) \leq F(x_c|\ x_s,\ X_m = true) \end{aligned} \tag{5}$$

Here there variables X_s, X_m and X_c are ordered binary categorical variables, their values x_s, x_m and x_c are from the set $\{false, true\}$. Moreover, $F(x_c) = P(X_c \leq x_c)$. Thus, for the example above, in $X_s \overset{+}{\rightarrow} X_c$, observing higher values for X_s makes higher values for X_c more likely, regardless of any other values of X_m. The arc signs ($\overset{+}{\rightarrow}$ and $\overset{-}{\rightarrow}$) specify the types of the monotonic influence. The negative influence represents the opposite relationship compared with the positive influence.

Although such monotonic influences has been well discussed in previous works (Wellman, 1990; Altendorf et al., 2005; Feelders and van der Gaag, 2006), there is no empirical analysis on using such generated exterior constraints in parameter leaning. Next, we will introduce a flat parameter prior (K2 prior) into the parameter learning with such exterior constraints, and derive the related equations for the constrained optimization problem.

4.3. The Constrained Optimization Method

Parameter learning with monotonic influence constraints can be formulated as a constrained optimization problem, and solved by the gradient descent approach. Therefore, the parameter estimation is converted to find the most probable parameters that maximize the log likelihood given training data and monotonic influence constraints. Any violation of constraints is penalized by reducing the objective log likelihood.

Without loss of generality, for a monotonic influence of X_j on X_i, we can generate its monotonic influence constraints (denoted by $ec_{i,p,q}^{k_c}$):

$$\textstyle\sum_{k=1}^{k_c} \theta_{ipk} \geq \sum_{k=1}^{k_c} \theta_{iqk}$$

where $1 \leq i \leq n,\ 1 \leq p \leq |\Pi_i|,\ 1 \leq q \leq |\Pi_i|,\ 1 \leq k \leq k_c$. p and q are two parent state configurations, their corresponding sub-indices[2] on X_j are denoted as $sub(p,j)$ and $sub(q,j)$. The k_c is the state index for which the cumulative distribution function is evaluated, and satisfies the condition $1 \leq k_c < r_i$.

If the monotonic influence is positive $(X_j \overset{+}{\rightarrow} X_i)$, we have $sub(p,j) < sub(q,j)$. If it is negative$(X_j \overset{-}{\rightarrow} X_i)$, the sub-indices of X_j satisfy the condition that $sub(p,j) > sub(q,j)$. Moreover, the sub-indices of X_i's parents in $\Pi_i \setminus \{X_j\}$ satisfy the condition $sub(p,\tilde{j}) = sub(q,\tilde{j})$ (*caeteris paribus* condition Wellman (1990)).

Let $C_i = \{ec^{k_c}_{i,p,q}\}$ represent all the elicited exterior constraints in X_i, which means:

$$C_i = \{\textstyle\sum_{k=1}^{k_c} \theta_{ipk} \geq \sum_{k=1}^{k_c} \theta_{iqk} | 1 \leq p \leq |\Pi_i|, 1 \leq q \leq |\Pi_i|, 1 \leq k \leq k_c < r_i\}$$

Therefore, the constrained maximization problem can be written as follows:

$$\begin{array}{c} \arg\max_\theta(\ell(\theta|D) - \frac{w}{2} \cdot \sum_{i=1}^{n} \sum_{ec^{k_c}_{i,p,q} \in C_i} penalty(ec^{k_c}_{i,p,q})) \\ s.t.\ \sum_{k=1}^{r_i} \theta_{ipk} - 1 = 0 \end{array} \tag{6}$$

where the w is the penalty weight, and chosen empirically. The penalty function is normally set as the squared difference of two parameters (Altendorf et al., 2005):

$$penalty(ec^{k_c}_{i,p,q}) = I_{\sum_{k=1}^{k_c}(\theta_{iqk} - \theta_{ipk}) \geq 0} \cdot (\textstyle\sum_{k=1}^{k_c}(\theta_{iqk} - \theta_{ipk}))^2$$

where $I_{(\cdot)}$ is the indicator function whose value equal to 1 if the condition $(\cdot)$ is satisfied, otherwise its value equal to 0.

Here, the condition $\sum_{k=1}^{r_i} \theta_{ipk} = 1$ ensures that the sum of all the estimated parameters in a probability distribution is equal to one. To eliminate this con-

[2] The conversion between parent state configurations and sub-indices is needed in the implementation of the algorithm. This is supported by the *ind2subv* function, which is available at `http://research.microsoft.com/en-us/um/people/minka/software/lightspeed/`.

dition, we introduce a new parameter μ_{ipk} so that

$$\theta_{ipk} = \frac{e^{\mu_{ipk}}}{\sum_{k=1}^{r_i} e^{\mu_{ipk}}} \tag{7}$$

Thus, the estimated parameters will automatically respect the condition $\sum_{k=1}^{r_i} \theta_{ipk} = 1$. Meanwhile, the local maximum w.r.t μ_{ipk} is also the local maximum w.r.t θ_{ipk}, and vice versa.

The solution of equation (6) moves towards the direction of reducing constraint violations and increasing data log likelihood. To ensure the returned solution is global optimum, the objective function must be convex, which limits the usage of constraints. Meanwhile, because the starting points are randomly generated in gradient descent, this may cause unacceptably poor parameter estimation results when learning with zero or limited data counts N_{ipk}. Thus, we simply improve the first term of equation (6) by introducing a flat Dirichlet prior θ':

$$J(\theta) = \ell(\theta|D, \theta') - \frac{w}{2} \cdot \sum_{i=1}^{n} \sum_{ec_{i,p,q}^{k_c} \in C_i} penalty(ec_{i,p,q}^{k_c}) \tag{8}$$

The derivative of $J(\theta)$ w.r.t μ can be expressed as:

$$\frac{\partial}{\partial \mu_{ipk}} J(\theta) = \frac{\partial}{\partial \mu_{ipk}} \ell(\theta|D, \theta') - \frac{w}{2} \cdot \sum_{i=1}^{n} \sum_{ec_{i,p,q}^{k_c} \in C_i} \frac{\partial}{\partial \mu_{ipk}} penalty(ec_{i,p,q}^{k_c}) \tag{9}$$

The derivative of the first term is defined by the partials:

$$\frac{\partial}{\partial \mu_{ipk}} \ell(\theta|D, \theta') = N_{ipk} + 1 - \frac{e^{\mu_{ipk}}}{\sum_{k=1}^{r_i} e^{\mu_{ipk}}} \cdot (N_{ip} + r_i) \tag{10}$$

The derivative of the second term is only valid when the constraint is violated. For each exterior constraint $ec_{i,p,q}^{k_c}$, the violation margin can be represented as:

$$\epsilon = \frac{\sum_{k=1}^{k_c} e^{\mu_{iqk}}}{\sum_{k=1}^{r_i} e^{\mu_{iqk}}} - \frac{\sum_{k=1}^{k_c} e^{\mu_{ipk}}}{\sum_{k=1}^{r_i} e^{\mu_{ipk}}} > 0 \tag{11}$$

Thus, the derivative of the penalty is defined by:

$$\begin{aligned}
&\frac{\partial}{\partial \mu_{ipk}} penalty(ec_{i,p,q}^{k_c}) = (-2) \cdot I_{\epsilon \geq 0} \cdot \epsilon \cdot e^{\mu_{ipk}} \cdot (\frac{I_{k \leq k_c} \cdot \sum_{k=1}^{r_i} e^{\mu_{ipk}} - \sum_{k=1}^{k_c} e^{\mu_{ipk}}}{(\sum_k e^{\mu_{ipk}})^2}) \\
&\frac{\partial}{\partial \mu_{iqk}} penalty(ec_{i,p,q}^{k_c}) = 2 \cdot I_{\epsilon \geq 0} \cdot \epsilon \cdot e^{\mu_{iqk}} \cdot (\frac{I_{k \leq k_c} \cdot \sum_{k=1}^{r_i} e^{\mu_{iqk}} - \sum_{k=1}^{k_c} e^{\mu_{iqk}}}{(\sum_k e^{\mu_{iqk}})^2})
\end{aligned} \tag{12}$$

Given the equation (6) and decomposability property of the log likelihood, the large optimization problem can be decomposed into n smaller sub-problems:

$$\theta_i = \arg\max_{\theta_i} (\ell(\theta_i|D) - \frac{w}{2} \cdot \sum_{ec_{i,p,q}^{k_c} \in C_i} penalty(ec_{i,p,q}^{k_c})) \tag{13}$$

Based on the gradients (equations (9–12)) discussed above, this sub-problem can be solved using the Karush-Kuhn-Tucker theorem. And we use Sequential Quadratic Programming (SQP) to compute its solutions (Nocedal and Wright, 2006). The parameter learning algorithm is shown in Algorithm 1. We refer to it as the Constrained Optimization algorithm with a Flat parameter Prior (COFP). The COFP algorithm consists of two parts. For unconstrained parameters, we perform the standard estimation. For constrained parameters, we make the solutions move towards the direction of increasing objective function value.

The $A = \{A_{j,i} | 1 \leq j \leq n, 1 \leq i \leq n\}$ in Algorithm 1 is the monotonic influence label matrix, where $A_{j,i} = 1$ and $A_{j,i} = -1$ represent a positive ($X_j \overset{+}{\rightarrow} X_i$) and a negative ($X_j \overset{-}{\rightarrow} X_i$) monotonic influence respectively, and $A_{j,i} = 0$ means there is no monotonic influence between X_j and X_i ($X_j \overset{?}{\rightarrow} X_i$).

4.4. Time Complexity Analysis

The constrained optimization step usually takes a fixed amount time to find the optimal parameter estimate. Therefore, we treat this optimization step as the elementary operation $O(1)$. The bottleneck in terms of efficiency of the COFP algorithm lies in the total number of exterior constraints generated from the monotonic influences.

Assuming there are n nodes in a BN and each node has maximum r states, the worst-case time complexity $T(n)$ of the COFP algorithm happens in the BN

```
INPUT   : Bayesian network $G$, Data $D$, Monotonic influence label
          matrix $A$.
OUTPUT: Estimated parameters $\theta = \{\theta_{ipk}\}$.
for each BN variable $i$ = 1 to $n$ do
    $C_i = \{\}$;
    for each parent variable $j$ = 1 to $|\Pi_i|$ do
        if there is no monotonic influences $A_{j,i} = 0$ then
            for parent state configuration $p = 1$ to $|\Pi_i|$ do
                for state index $k = 1$ to $r_i$ do
                    $\theta_{ipk} = \frac{N_{ipk}+1}{N_{ip}+r_i}$;
                end
            end
        else
            if $A_{j,i} = 1$ then
                for parent state configuration pair $p$ and $q$ satisfying
                $sub(p, j) < sub(q, j)$ and $sub(p, \tilde{j}) = sub(q, \tilde{j})$ do
                    $C_i = \{C_i,\ ec^{k_c}_{i,p,q}\}$;
                end
            else
                for parent state configuration pair $p$ and $q$ satisfying
                $sub(p, j) > sub(q, j)$ and $sub(p, \tilde{j}) = sub(q, \tilde{j})$ do
                    $C_i = \{C_i,\ ec^{k_c}_{i,p,q}\}$;
                end
            end
        end
    end
    $\theta_i = \arg\max_{\theta_i}(\ell(\theta_i|D) - \frac{w}{2} \cdot \sum_{ec^{k_c}_{i,p,q} \in C_i} penalty(ec^{k_c}_{i,p,q}))$;
end
return  $\theta = \{\theta_i\}$
```

Algorithm 1: COFP BN parameter learning algorithm with monotonic influence constraints and a flat parameter prior

structure that contains one child node and its $n-1$ parent nodes, with all $n-1$ edges fully specified by monotonic influences. Therefore, the total number of monotonic influence constraints is equal to the product of total number of parent configurations and the number of child node states. Hence, the worst-case time complexity is exponential with respect to the total number of nodes:

$$T(n) = O(\frac{r^{(n-1)}!}{2!(r^{(n-1)}-2)!}r) = O(\frac{1}{2}(r^{(n^2-2n+2)} - r^n)) \tag{14}$$

Despite this complexity, the COFP algorithm is able to produce the results relatively efficiently for all the real-world models examined in Section 5 with the biggest model running in 23.81 seconds[3].

5. Experiments

The experiments have two goals. First to demonstrate the widespread existence of monotonic influences in BNs in the publicly available repository that we describe in Section 5.1 and second to show the advantages of using the generated exterior constraints in parameter learning (with the COFP algorithm).

In the experiments in this section we are relying on publicly available BNs. Unlike the real-world case study that follows in Section 6, this means there are a number of necessary experimental conditions, which we acknowledge are only a simulation of reality (and hence limit the general applicability of the results). Specifically:

- In practice we would have a known BN structure (this is the key scenario we are assuming, as explained in the Introduction) together with a small amount of real-world data relevant to the BN variables. However, here we have to simulate such data – not just because we do not actually have it, but also because we wish to evaluate our method under different sample sizes. To simulate a realistic set of data we have to assume that the actual CPTs provided with the models represent the 'true' model parameters (of course, in practice the 'true' parameters are unknown because this is what we are trying to learn). To do this we use the *forwards sampling* function

[3] Relevant experiments are performed on an Intel core i7 CPU running at 2.5 GHz and 16 GB RAM.

which is built into the BNT package[4] to generate data samples (of sizes 50, 100 and 500) randomly from the distributions of the 'true' CPTs.

- In practice we would have one or more domain experts on hand from whom to elicit the monotonic influence labels. Because we do not have access to such experts, we are using the 'true' labels from the BN. However, to simulate reality we include experiments in which some erroneous labels are randomly generated.

For comparison we consider the following parameter algorithms:

- Conventional parameter BN learning algorithms (MLE and MAP).
- Constrained optimization (CO) algorithm (considering exterior constraints) (Altendorf et al., 2005; Liao and Ji, 2009).
- Our improved CO algorithm by incorporating a flat parameter prior (COFP).

In all cases the resulting learnt CPTs are evaluated against the true CPTs by using the K-L divergence metric[5] (Kullback and Leibler, 1951), which is recommended to measure the distance between distributions. The smaller the K-L divergence is, the closer the estimated CPT values are to the true CPT values. If estimated CPT values are zero, they are replaced by a tiny real value (1×10^{-7}) to guarantee they can be computed by the K-L divergence. Moreover, each experiment setting is repeated 10 times, and the results are presented with their mean and standard deviation.

5.1. The BNs Used in The Experiments

The publicly available BN repository[6] contains 20 complete BNs, most of which have been developed in total or in part by domain experts. Hence, they

[4] BNT is a Matlab toolbox called "Bayes Net Toolbox" that can be found at `https://code.google.com/p/bnt/`.

[5] Here the K-L divergence is locally measured for each CPT column and averaged over the whole model. This is to ensure that the fit of each distribution is equally weighted in the overall metric.

[6] `http://www.bnlearn.com/bnrepository/`

satisfy the scope of our work. These BNs are encoded in the Bayesian Interchange Format (.bif). We used a Perl program called *bif2bnt*[7] to convert these BNs into the standard BNT format. This resulted in 12 of the BNs being successfully converted and used in the experiments; the rest were clearly not well-defined, but there is no reason to believe that the 12 that were successfully extracted are not representative of the kind of real-world BN models that satisfy the scope of our work.

Table 1: Details and monotonic influence analysis of 12 publicly available Bayesian networks.

Name	Nodes	Edges	Parameters	$\xrightarrow{+}$ Edges	$\xrightarrow{-}$ Edges
Alarm	37	46	509	18	2
Andes	223	338	1157	336	1
Asia	8	8	18	8	0
Cancer	5	4	10	3	1
Earthquake	5	4	10	4	0
Hailfinder	56	66	2656	23	0
Hepar2	70	123	1453	34	12
Insurance	27	52	984	9	2
Sachs	11	17	178	2	0
Survey	6	6	21	2	1
Weather	4	4	9	3	1
Win95pts	76	112	574	26	3

Table 1 provides a summary of these BNs. They range from typically small expert-built BNs to those which are as large as any that could be reasonably produced by experts. Each edge of these BNs is investigated for qualitative monotonicity, and the details are also described in Table 1. As we can see, monotonic influences are widespread in all these BNs; in half of them the vast majority of edges have monotonic influences.

To illustrate in detail the findings we use the example of the well-known Alarm BN, which is an acronym for “A Logical Alarm Reduction Mechanism”. This BN is a medical diagnostic application used for patient monitoring that contains 37 variables in total: 8 diagnoses, 16 findings and 13 intermediate variables. The BN has 46 edges and 509 parameters, the maximum edge in-degree is 4.

Figure 2 shows the full structure of the BN, where the signs on the edges

[7] `http://www.digitas.harvard.edu/~ken/bif2bnt/`

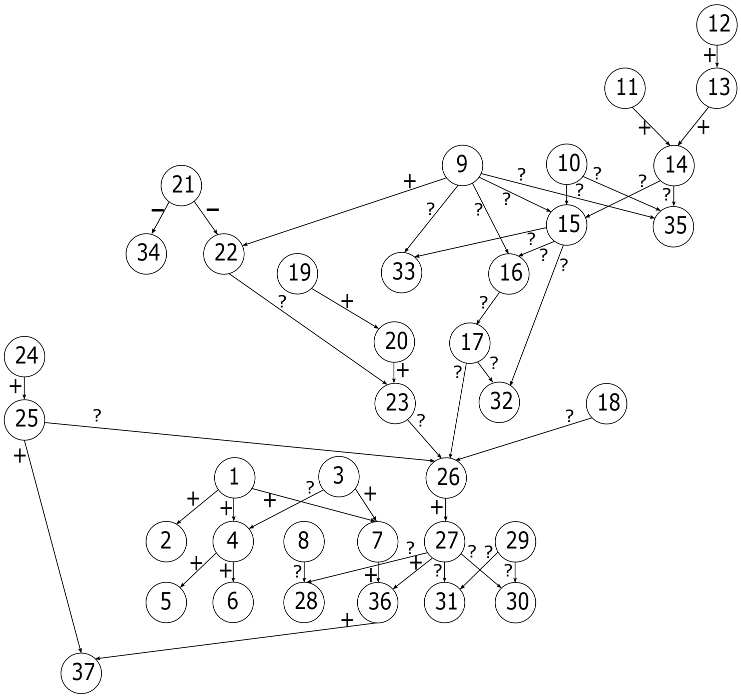

Figure 2: The monotonic influence labels in the Alarm BN.

indicate whether the associated monotonic influences are positive or negative. There are 18 positive and 2 negative monotonic influences, which means that 43.5% of the edges encode such monotonic influences.

5.2. Results with Different Data Sparsity

In this experiment, we consider three training data sizes: 50, 100 and 500. Table 2 summarises the average K-L divergence between learnt BNs and true BNs for three different training sample sizes. The best results are presented in bold. Statistically significant improvements of the best results over competitors are indicated with asterisks * ($p \leq 0.05$).

For sample size 50 (Table 2(a)), MAP, CO and COFP all achieve good performance compared with the conventional MLE, which suffers from the absence of data in several state configurations in such limited data. Moreover, COFP significantly outperforms MAP and CO in most experiment settings. Specifically, compared with MAP and CO results, COFP achieves 16.3% and 70.0%

Table 2: Learning results of MLE, MAP, CO and COFP on 12 publicly available BNs.

(a)

50 Data Samples	MLE	MAP	CO	COFP
Alarm	2.83±0.18*	0.76±0.03*	2.48±0.17*	**0.67**±0.03
Andes	1.50±0.03*	0.25±0.01*	0.17±0.01*	**0.16**±0.01
Asia	0.98±0.22*	0.44±0.03*	**0.23**±0.08	0.30±0.04*
Cancer	0.93±0.49*	0.11±0.03	0.11±0.04	**0.08**±0.03
Earthquake	1.58±0.73*	0.16±0.03	0.30±0.06*	**0.14**±0.05
Hailfinder	3.39±0.04*	0.57±0.01*	3.24±0.03*	**0.49**±0.01
Hepar2	3.48±0.09*	0.36±0.01*	3.23±0.08*	**0.35**±0.01
Insurance	2.49±0.11*	1.39±0.01*	2.07±0.10*	**1.29**±0.02
Sachs	2.21±0.15*	0.91±0.03*	1.97±0.13*	**0.84**±0.02
Survey	0.47±0.11*	0.05±0.01*	0.15±0.05*	**0.03**±0.01
Weather	**0.03**±0.03	0.07±0.02*	**0.03**±0.03	0.04±0.01
Win95pts	3.88±0.09*	0.89±0.01*	3.22±0.12*	**0.83**±0.01
Average	1.98±0.19	0.50±0.02	1.43±0.08	**0.43**±0.02

(b)

100 Data Samples	MLE	MAP	CO	COFP
Alarm	2.24±0.12*	0.65±0.02*	2.03±0.10*	**0.58**±0.03
Andes	1.06±0.02*	0.18±0.00*	0.11±0.03	**0.10**±0.00
Asia	0.57±0.28*	0.34±0.06*	**0.09**±0.08	0.19±0.07*
Cancer	0.61±0.60*	0.08±0.03	0.12±0.09	**0.07**±0.04
Earthquake	1.16±0.46*	0.14±0.04	0.35±0.29*	**0.11**±0.06
Hailfinder	2.86±0.03*	0.46±0.01*	2.76±0.02*	**0.40**±0.01
Hepar2	3.13±0.10*	0.33±0.01	2.97±0.08*	**0.32**±0.01
Insurance	1.85±0.11*	1.17±0.02*	1.59±0.09*	**1.07**±0.02
Sachs	1.67±0.16*	0.76±0.03*	1.50±0.14*	**0.69**±0.02
Survey	0.35±0.15*	0.04±0.01*	0.11±0.04*	**0.03**±0.01
Weather	**0.02**±0.02	0.03±0.01*	**0.02**±0.02	**0.02**±0.01
Win95pts	3.61±0.07*	0.82±0.02*	2.99±0.12*	**0.74**±0.02
Average	1.59±0.18	0.42±0.02	1.22±0.09	**0.36**±0.03

(c)

500 Data Samples	MLE	MAP	CO	COFP
Alarm	1.39±0.13*	0.43±0.02*	1.29±0.14*	**0.39**±0.02
Andes	0.37±0.03*	0.07±0.00*	0.05±0.01*	**0.02**±0.00
Asia	0.25±0.15*	0.21±0.03*	**0.02**±0.01	0.05±0.01*
Cancer	0.05±0.03*	**0.01**±0.01	0.03±0.02*	**0.01**±0.01
Earthquake	0.59±0.19*	0.08±0.03*	0.10±0.08	**0.04**±0.05
Hailfinder	1.52±0.03*	0.24±0.00*	1.50±0.03*	**0.22**±0.00
Hepar2	2.43±0.10*	**0.26**±0.01	2.36±0.09*	**0.26**±0.01
Insurance	0.88±0.04*	0.65±0.01*	0.77±0.04*	**0.58**±0.01
Sachs	0.95±0.16*	0.47±0.04*	0.90±0.16*	**0.44**±0.04
Survey	0.04±0.01*	0.02±0.01	0.02±0.01*	**0.01**±0.00
Weather	**0.00**±0.00	0.01±0.00*	**0.00**±0.00	**0.00**±0.00
Win95pts	2.97±0.06*	0.64±0.01*	2.53±0.13*	**0.50**±0.01
Average	0.95±0.08	0.26±0.01	0.80±0.06	**0.21**±0.01

average reductions of K-L divergence respectively.

For sample size 100 (Table 2(b)), the performance of MLE, MAP, CO and COFP are all improved compared with their results in 50 data samples. Specifically, in the small network (Weather BN), the basic MLE also achieves the best learning result, which means 100 training example is already enough to train a good model. Again, COFP beats the competitors in every setting except the Asia BN. Compared with MAP and CO results, COFP achieves 14.3% and 70.5% average reductions of K-L divergence respectively.

For sample size 500 (Table 2(c)), the performance of MLE, MAP, CO and COFP are further improved. Compared with other learning methods, COFP still achieves the best overall learning performance. Specifically, of the total 12 experiments, 10 experiments show improvement in COFP over both MAP and CO. Moreover, the COFP achieves 19.2% and 73.8% average reductions of K-L divergence compared with MAP and CO.

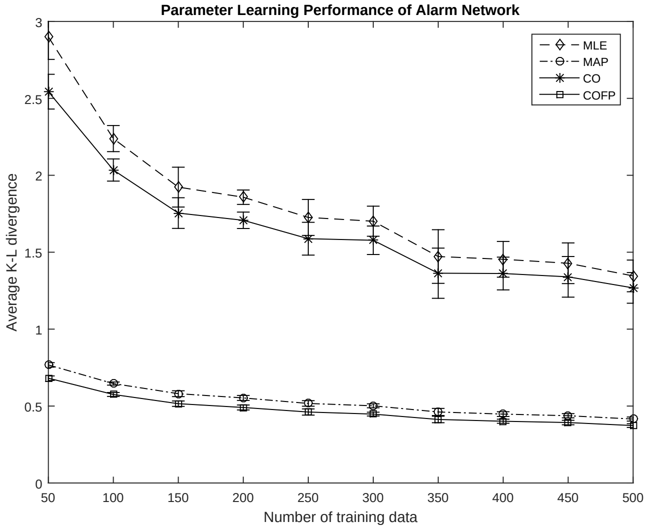

Figure 3: The learning performance of MLE, MAP, CO and COFP in the Alarm BN under different data sizes.

As a detailed example, Figure 3 highlights the learning results of the Alarm BN under different data sizes ranging from 50 to 500 samples. It is clear that the average K-L divergence of all four algorithms show the decreasing trends with increasing sample sizes. Moreover, the CO results always outperform the MLE results, which demonstrate the usefulness of using elicited exterior constraints from monotonic influences. As expected, with the increase of data sizes, the gap between the performances of CO and MLE decreases.

More importantly, COFP greatly outperforms MLE and CO, and it also outperforms MAP with 10.8% average reduction of K-L divergence. These findings show the superiority and effectiveness of applying COFP in the Alarm BN parameter estimation with extremely limited data.

5.3. Time Complexity Analysis

As discussed in Section 4.4, the computational complexity is mainly determined by the total number of exterior constraints. Table 3 describes the average computational time of each learning task for different learning algorithms.

Table 3: Running time (seconds) for MLE, MAP, CO and COFP in 12 publicly available BN parameter learning problems.

Name	Data	MLE	MAP	CO	COFP
Alarm	50	0.02	0.01	2.75	0.98
	100	0.02	0.01	2.54	1.17
	500	0.02	0.01	2.61	1.40
Andes	50	0.05	0.05	18.38	9.67
	100	0.05	0.07	21.81	15.60
	500	0.06	0.06	22.71	12.11
Asia	50	0.00	0.00	0.23	0.19
	100	0.00	0.00	0.23	0.20
	500	0.00	0.00	0.22	0.21
Cancer	50	0.00	0.00	0.12	0.09
	100	0.00	0.00	0.13	0.11
	500	0.00	0.00	0.12	0.11
Earthquake	50	0.00	0.00	0.13	0.10
	100	0.00	0.00	0.15	0.11
	500	0.00	0.00	0.16	0.10
Hailfinder	50	0.01	0.01	2741.99	20.05
	100	0.01	0.01	1694.87	21.78
	500	0.02	0.02	2579.77	23.81
Hepar2	50	0.02	0.01	2.91	1.66
	100	0.02	0.02	3.01	1.99
	500	0.02	0.02	2.73	2.19
Insurance	50	0.01	0.01	8.19	4.36
	100	0.01	0.01	12.23	5.39
	500	0.01	0.01	19.32	8.13
Sachs	50	0.00	0.00	0.98	0.32
	100	0.00	0.00	0.90	0.31
	500	0.00	0.00	0.59	0.36
Survey	50	0.00	0.00	0.16	0.11
	100	0.00	0.00	0.14	0.10
	500	0.00	0.00	0.13	0.14
Weather	50	0.00	0.00	0.11	0.10
	100	0.00	0.00	0.10	0.10
	500	0.00	0.00	0.14	0.12
Win95pts	50	0.02	0.02	6.92	4.80
	100	0.02	0.02	7.55	5.38
	500	0.03	0.02	7.73	6.78
Average	N/A	0.01	0.01	198.97	4.17

As expected from the complexity analysis, there is clearly a much greater computational overhead in using CO and COFP compared to MLE and MAP (which is inevitable given the iteration steps in the constraint optimization).

Crucially, however, COFP performs much more efficiently than CO (it outperforms CO in 34 of the 36 experiments).

5.4. Results with Error Labels

As shown in the above experiments, incorporating exterior constraints generated from monotonic influences can significantly improve the learning performance. However, in real-world applications it is inevitable that, when eliciting such constraints from experts, the influence labels will sometimes be wrong. Hence, it is important to investigate the sensitivity of the results to such errors. To this end, we generate "error labels" for a randomly selected small subset of the previously elicited monotonic influences (where an error label is a positive influence labelled negative or vice versa). We consider two sets of experiments for each BN: one in which there is exactly one edge with an error label; and one in which 5% of the edges have error labels (also note that, for BNs with less than 20 edges with monotonic influences, these are the same). We feel that anything more than 5% does not realistically represent expert judgment error.

Table 4: Learning results of MLE, MAP, CO and MPL-EC with error monotonic influence labels and 100 training data samples in 12 publicly available BN parameter learning problems.

Name	1 Error Label		5% Error Labels[†]		COFP Better or Equal
	CO	COFP	CO	COFP	Than MAP?
Alarm	2.09±0.10	**0.59**±0.02	2.09±0.10	**0.59**±0.02	Yes
Andes	0.11±0.01	**0.10**±0.00	0.19±0.03	0.13±0.00	Yes
Asia	**0.30**±0.10	0.41±0.11	**0.30**±0.10	0.41±0.11	Yes
Cancer	0.22±0.07	**0.16**±0.03	0.22±0.07	**0.16**±0.03	No
Earthquake	0.86±0.21	**0.46**±0.06	0.86±0.21	**0.46**±0.06	No
Hailfinder	2.75±0.04	**0.42**±0.01	2.77±0.05	0.43±0.01	Yes
Hepar2	2.94±0.07	**0.32**±0.01	2.94±0.07	**0.32**±0.01	Yes
Insurance	1.69±0.12	**1.12**±0.02	1.69±0.12	**1.12**±0.02	Yes
Sachs	1.63±0.15	**0.75**±0.02	1.63±0.15	**0.75**±0.02	Yes
Survey	0.16±0.08	**0.03**±0.01	0.16±0.08	**0.03**±0.01	Yes
Weather	**0.07**±0.02	**0.07**±0.01	**0.07**±0.02	**0.07**±0.01	No
Win95pts	3.08±0.11	**0.75**±0.02	3.10±0.11	0.83±0.02	Yes
Average	1.33±0.09	**0.43**±0.03	1.33±0.09	0.44±0.03	N/A

[†] The final number of error labels is rounded up to the nearest integer.

As expected, the results in (Table 4) show the performance of CO and COFP are both worse than their previous results that learnt with correct influence labels (Table 2(b)). For example, the average K-L divergence of COFP learnt with one error label and 5% error labels are 0.43 and 0.44, which are much higher than the previous result (0.36) learnt with correct influence labels.

However, as is shown in the last column of Table 4, even with the errors introduced, COFP outperforms MAP in most cases. The exceptions are the Cancer, Earthquake and Weather BNs, which have less than 5 edges with monotonic influence constraints (so a single error represents 25% of the edges, and it is unsurprising in such cases that the performance of COFP is badly affected).

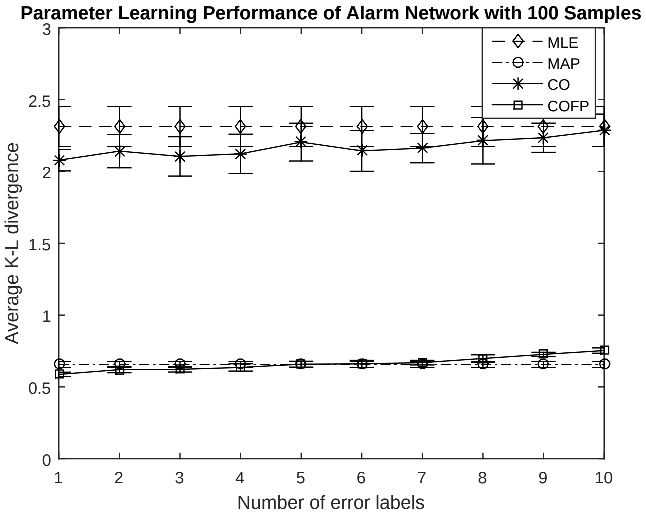

Figure 4: The learning performance of CO and COFP in the Alarm BN with increasing number of error labels.

For large BNs, the performance of COFP is remarkably robust to errors, as can be seen in Figure 4, which shows how the introduced errors affect the learning results of CO and COFP in the Alarm BN with 100 data samples. Obviously errors are irrelevant for the learning performance of MLE and MAP so their estimation results are fixed over varying numbers of error labels. As we can see, there are increasing trends of K-L divergence in CO and COFP with increasing number of error monotonic influences. The slight fluctuation comes from randomly chosen error labels in each experiment repetition. However, COFP outperforms MAP with up to 5 errors (which is 25% of the relevant edges).

6. A Real Medical Case Study

The previous experiments demonstrated the effectiveness of our COFP algorithm on repository BN models under simulated conditions of scarce data and generated exterior constraints. In this section we demonstrate its effectiveness to learn parameters of a real BN developed for a medical problem, where the "true" parameters and monotonic influences are unknown.

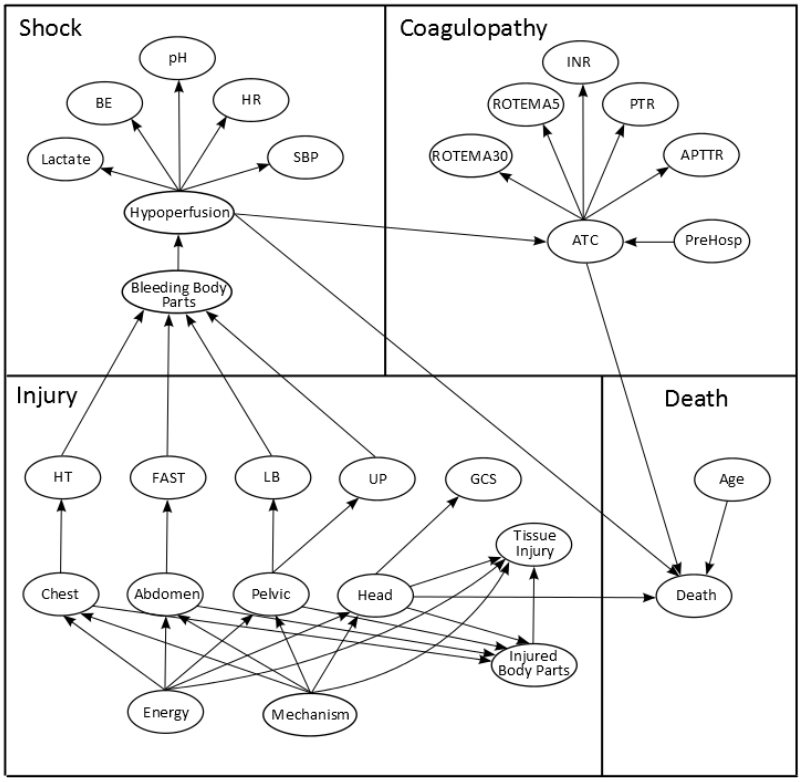

Figure 5: The Trauma Care Bayesian network, which contains four main parts: "Injury", "Shock", "Coagulopathy" and "Death".

The BN was developed by trauma care specialists, and relates to procedures in hospital. The full details of the BN (whose graph is shown in Figure 5) and datasets are proprietary to the hospitals involved. This BN contains 18 discrete variables (of which 3 are hidden) and 11 Gaussian variables[8] that are grouped into 4 parts:

- the degree of overall tissue injury,
- the degree of hypoperfusion resulting from blood loss for the patient,
- the risk of developing acute traumatic coagulopathy, and
- the risk of death for the patient.

Here, a well learnt BN is important because rapid and accurate identification of hidden risk factors and conditions modelled by the network are important to support a doctors' decision making about treatments which reduce mortality rate (Karaolis et al., 2010).

In this experiment, we elicited monotonic influences from medical experts and were also give access to a hospital dataset. Hence we are able, in a real-world setting, to evaluate the MPL-TC method.

The monotonic influences and their descriptions are shown in Table 6, which constrain the variables: 'Death', 'ATC', 'Age', 'Hypoperfusion', and 'Head' (their details can be found in Table 5).

Table 5: Details of constrained variables.

Variable	Description	States
Death	The risk of patient's death in 48 hours.	No Yes
ATC	Acute traumatic coagulopathy.	No Yes
Age	Patient's age.	Y: $Age \leq 45$ M: $45 < Age < 65$ O: $Age \geq 65$
Hypoperfusion	The degree of decreased blood flow through an organ.	Uncompensated None Compensated
Head	Severe head injury of patient.	No Yes

The real dataset was collected from an inner city hospital in Germany, and contains 105 instances. We perform cross-validation in this dataset, using half

[8] The details of these variables can be found in `http://www.traumamodels.com/`.

Table 6: The elicited expert judgments for monotonic influences in the Trauma Care BN.

Monotonic influences	Description
ATC $\xrightarrow{+}$ Death	ATC occurs will result in the death of patient with very *high* probability.
Age $\xrightarrow{+}$ Death	Old patient has *higher* risk of death than young patient.
Hypoperfusion $\xrightarrow{-}$ Death	Uncompensated hypoperfusion will very *likely* result in the death of patient.
Head $\xrightarrow{+}$ Death	Severe head injury will *likely* result in the death of patient.

Table 7: Prediction performance (AUC) for the Trauma Care BN. The query variable is "Death".

Algorithm	MLE	MAP	CO	COFP
AUC	0.829*	0.872	0.859*	**0.938**

the instances to train the model, and half to evaluate the model. To evaluate the model we instantiate the evidence variables in the target domain test set, select one of the variables of interest (*Death*), and query this variable. AUC values are calculated for the query variable. To get an effective decision support model, we need to pick up the trained model that has the highest AUC value. The results are presented in Table 7, with the best result in bold, and statistically significant improvements of the best result over competitors indicated with asterisks * ($p \leq 0.05$).

As shown in Table 7, all learning algorithms mentioned in this paper have been compared. Due to data scarcity, MAP outperforms MLE. After incorporating constraints generated from monotonic influences, the performance of CO is better than MLE, which demonstrates the correctness of expert judgments in Table 6. Moreover, the COFP achieves the best result, which shows the potential benefit of using COFP for real-world decision support problems, especially when the training data are extremely limited.

7. Conclusions and Future Work

When data are limited, purely data driven BN learning is inaccurate. In this paper our focus is on those scenarios in which we have a BN whose structure is expert-defined, but whose parameters we seek to learn from a combination of scarce data and expert judgments. By incorporating monotonic influence constraints discussed in this paper parameter learning performance is significantly improved.

The broad goal of this paper was to understand the monotonic influence constraints in a range of BNs, and to determine the extent to which knowledge of such constraints improved learning performance. We analysed such properties in each edge of every readable BN in the publicly available BN repository. Surprisingly, monotonic influences were widespread in all the BNs (typically over 40% of all edges in most of the 12 BNs used in the study). We described an improved parameter learning algorithm (COFP) that incorporates constraints generated from these monotonic influences, and compared its performance to MLE, MAP and the previous state-of-the-art algorithm CO using a range of different sample size settings.

We demonstrated that over the full set of models in the experiment COFP consistently outperforms CO. We also demonstrated that, while COFP is obviously far more computationally demanding than MLE and MAP it is actually at least as efficient as CO in most BNs. We also showed that, COFP is robust with respect to a small number of error labels, especially in large BNs. In Alarm BN, it requires more than 25% errors before COFP is outperformed by MAP.

The experiments in Section 5 were only a simulation of the real-world problem of learning parameters for a fixed BN structure given scarce data together with expert judgments. However, we believe the set of BNs was representative of those defined within the scope of our research, and the simulation method, which included simulating expert errors, was a reasonable match to real-world scenarios. In our real medical case study, we had access to a BN structure developed by trauma care experts (for coagulopathy risk), together with expert elicited monotonic influences and a hospital dataset; the COFP algorithm achieved the best learning results.

While this paper has provided a contribution to improving the accuracy of BN parameter learning by incorporating monotonic influences, there are a number of areas in which the work could be extended in future research. First, in real-world decision support applications, the dataset might contain 'missing values'. In such cases the simplest way continue to use our algorithm is to employ imputation techniques that fill the missing values of the dataset with the most likely value; however, this may introduce large amounts of bias especially when the data is scarce. To address this we could apply the EM algorithm (Murphy, 2012), where the exterior constraints should be incorporated in the M-step. Because of the multiple iterations in the EM algorithm, this would, however reduce the efficiency of the algorithm. A second area of future research would be to investigate the existence of the extended representations of monotonic influences in the BNs that we studied, which are named *context-specific influences* (Renooij et al., 2002). This representation can model knowledge about monotonic influences ($X_j \to X_i$) that hold only for specific values of $\Pi_i \setminus \{X_j\}$. Therefore, ambiguous monotonic influences could be further exploited, and might be used to improve the parameter learning accuracy in some BNs.

8. Acknowledgments

The authors would like to thank the editor and two anonymous reviewers and for their valuable feedback. This work is supported by the European Research Council (ERC-2013-AdG339182-BAYES-KNOWLEDGE) and the China Scholarship Council (CSC)/Queen Mary Joint PhD scholarships. YZ and CZ are supported by and National Natural Science Foundation of China (61273322, 71471174).

9. References

Altendorf, E.E., Restificar, A.C., Dietterich, T.G., 2005. Learning from sparse data by exploiting monotonicity constraints, in: Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, pp. 18–26.

Barber, D., 2012. Bayesian Reasoning and Machine Learning. Cambridge University Press.

Baumgartner, K., Ferrari, S., Palermo, G., 2008. Constructing Bayesian networks for criminal profiling from limited data. Knowledge-Based Systems 21, 563–572.

de Campos, C.P., Cozman, F.G., 2005. Belief updating and learning in semi-qualitative probabilistic networks, in: Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, pp. 153–160.

de Campos, C.P., Ji, Q., 2008. Improving Bayesian network parameter learning using constraints, in: Proceedings of the 19th International Conference on Pattern Recognition, pp. 1–4.

de Campos, C.P., Tong, Y., Ji, Q., 2008. Constrained maximum likelihood learning of Bayesian networks for facial action recognition, in: Proceedings of the 10th European Conference on Computer Vision. Springer, pp. 168–181.

de Campos, C.P., Zeng, Z., Ji, Q., 2009. Structure learning of Bayesian networks using constraints, in: Proceedings of the 26th International Conference on Machine Learning, ACM. pp. 113–120.

Cano, A., Masegosa, A.R., Moral, S., 2011. A method for integrating expert knowledge when learning Bayesian networks from data. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 41, 1382–1394.

Chang, R., Stetter, M., Brauer, W., 2008. Quantitative inference by qualitative semantic knowledge mining with Bayesian model averaging. Knowledge and Data Engineering, IEEE Transactions on 20, 1587–1600.

Constantinou, A.C., Fenton, N., Marsh, W., Radlinski, L., 2016. From complex questionnaire and interviewing data to intelligent Bayesian network models for medical decision support. Artificial intelligence in medicine .

Constantinou, A.C., Fenton, N.E., Neil, M., 2013. Profiting from an inefficient association football gambling market: Prediction, risk and uncertainty using Bayesian networks. Knowledge-Based Systems 50, 60 –86.

Constantinou, A.C., Freestone, M., Marsh, W., Coid, J., 2015a. Causal inference for violence risk management and decision support in forensic psychiatry. Decision Support Systems 80, 42–55.

Constantinou, A.C., Freestone, M., Marsh, W., Fenton, N., Coid, J., 2015b. Risk assessment and risk management of violent reoffending among prisoners. Expert Systems with Applications 42, 7511–7529.

Diez, F.J., 1993. Parameter adjustment in Bayes networks. the generalized noisy OR-gate, in: Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann. pp. 99–105.

Druzdel, M., Van Der Gaag, L.C., 2000. Building probabilistic networks:“where do the numbers come from?”. Knowledge and Data Engineering, IEEE Transactions on 12, 481–486.

Druzdzel, M.J., van der Gaag, L.C., 1995. Elicitation of probabilities for Belief networks: Combining qualitative and quantitative information, in: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, pp. 141–148.

Druzdzel, M.J., Henrion, M., 1993. Efficient reasoning in qualitative probabilistic networks, in: Proceedings of the 11th National Conference on Artificial Intelligence, Washington, pp. 548–553.

Eaton, D., Murphy, K., 2007. Bayesian structure learning using dynamic programming and mcmc, in: Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence, AUAI Press. pp. 101–108.

Feelders, A., 2007. A new parameter learning method for Bayesian networks with qualitative influences, in: Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence, pp. 117–124.

Feelders, A., van der Gaag, L., 2006. Learning Bayesian network parameters under order constraints. International Journal of Approximate Reasoning 42, 37–53.

Fenton, N., Neil, M., 2010. Comparing risks of alternative medical diagnosis using bayesian arguments. Journal of Biomedical Informatics 43, 485–495.

Fenton, N., Neil, M., 2012. Risk Assessment and Decision Analysis with Bayesian Networks. CRC Press, New York.

Fenton, N., Neil, M., Caballero, J.G., 2007. Using ranked nodes to model qualitative judgments in Bayesian networks. Knowledge and Data Engineering, IEEE Transactions on 19, 1420–1432.

Fenton, N., Neil, M., Lagnado, D.A., 2013. A general structure for legal arguments about evidence using Bayesian networks. Cognitive Science 37, 61–102.

Flores, M.J., Nicholson, A.E., Brunskill, A., Korb, K.B., Mascaro, S., 2011. Incorporating expert knowledge when learning Bayesian network structure: A medical case study. Artificial Intelligence in Medicine 53, 181 –204.

van der Gaag, L.C., Renooij, S., Geenen, P.L., 2006. Lattices for studying monotonicity of Bayesian networks, in: Proceedings of the 3rd European Workshop on Probabilistic Graphical Models, pp. 99–106.

van der Gaag, L.C., Tabachneck-Schijf, H.J., Geenen, P.L., 2009. Verifying monotonicity of Bayesian networks with domain experts. International Journal of Approximate Reasoning 50, 429 – 436.

Heckerman, D., Geiger, D., Chickering, D.M., 1995. Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning 20, 197–243.

Helsper, E.M., van der Gaag, L.C., Feelders, A.J., Loeffen, W.L., Geenen, P.L., Elbers, A.R., 2005. Bringing order into Bayesian-network construction, in: Proceedings of the 3rd International Conference on Knowledge Capture, ACM. pp. 121–128.

Hutchinson, R.A., Niculescu, R.S., Keller, T.A., Rustandi, I., Mitchell, T.M., 2009. Modeling fMRI data generated by overlapping cognitive processes with unknown onsets using hidden process models. NeuroImage 46, 87 – 104.

Karaolis, M., Moutiris, J., Hadjipanayi, D., Pattichis, C., 2010. Assessment of the risk factors of coronary heart events based on data mining with decision trees. Information Technology in Biomedicine, IEEE Transactions on 14, 559–566.

Kullback, S., Leibler, R.A., 1951. On information and sufficiency. The Annals of Mathematical Statistics , 79–86.

Liao, W., Ji, Q., 2009. Learning Bayesian network parameters under incomplete data with domain knowledge. Pattern Recognition 42, 3046–3056.

Lucas, P.J., van der Gaag, L.C., Abu-Hanna, A., 2004. Bayesian networks in biomedicine and health-care. Artificial Intelligence in Medicine 30, 201–214.

Masegosa, A.R., Moral, S., 2013. An interactive approach for Bayesian network learning using domain/expert knowledge. International Journal of Approximate Reasoning 54, 1168–1181.

Murphy, K.P., 2012. Machine Learning: A Probabilistic Perspective. The MIT Press, Cambridge.

Neapolitan, R.E., 2004. Learning Bayesian networks. Pearson Prentice Hall.

Neil, M., Fenton, N., Nielson, L., 2000. Building large-scale Bayesian networks. The Knowledge Engineering Review 15, 257–284.

Niculescu, R.S., Mitchell, T., Rao, B., 2006. Bayesian network learning with parameter constraints. The Journal of Machine Learning Research 7, 1357–1383.

Nocedal, J., Wright, S.J., 2006. Least-Squares Problems. Springer.

O'Hagan, A., Buck, C.E., Daneshkhah, A., Eiser, J.R., Garthwaite, P.H., Jenkinson, D.J., Oakley, J.E., Rakow, T., 2006. Uncertain judgements: eliciting experts' probabilities. Wiley. com.

Pearl, J., 1988. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann.

Pradhan, M., Provan, G., Middleton, B., Henrion, M., 1994. Knowledge engineering for large Belief networks, in: Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc.. pp. 484–490.

Renooij, S., van der Gaag, L.C., 2002. From qualitative to quantitative probabilistic networks, in: Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc.. pp. 422–429.

Renooij, S., Van der Gaag, L.C., 2008. Enhanced qualitative probabilistic networks for resolving trade-offs. Artificial Intelligence 172, 1470–1494.

Renooij, S., Van der Gaag, L.C., Parsons, S., 2002. Context-specific sign-propagation in qualitative probabilistic networks. Artificial Intelligence 140, 207–230.

Silander, T., Kontkanen, P., Myllymaki, P., 2007. On sensitivity of the MAP Bayesian network structure to the equivalent sample size parameter, in: Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence, AUAI Press. pp. 360–367.

Wellman, M.P., 1990. Fundamental concepts of qualitative probabilistic networks. Artificial Intelligence 44, 257–303.

Xiang, Y., Jia, N., 2007. Modeling causal reinforcement and undermining for efficient CPT elicitation. Knowledge and Data Engineering, IEEE Transactions on 19, 1708–1718.

Xiang, Y., Truong, M., 2014. Acquisition of causal models for local distributions in Bayesian networks. Cybernetics, IEEE Transactions on 44, 1591–1604.

Yang, S., Natarajan, S., 2013. Knowledge intensive learning: Combining qualitative constraints with causal independence for parameter learning in probabilistic models, in: Machine Learning and Knowledge Discovery in Databases. Springer, pp. 580–595.

Yet, B., Bastani, K., Raharjo, H., Lifvergren, S., Marsh, W., Bergman, B., 2013. Decision support system for Warfarin therapy management using Bayesian networks. Decision Support Systems 55, 488–498.

Yet, B., Perkins, Z., Fenton, N., Tai, N., Marsh, W., 2014. Not just data: A method for improving prediction with knowledge. Journal of Biomedical Informatics 48, 28 – 37.

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

Zhou, Y., Fenton, N., Hospedales, T., Neil, M., 2015. Probabilistic graphical models parameter learning with transferred prior and constraints, in: Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence, AUAI Press. pp. 972–981.

Zhou, Y., Fenton, N., Neil, M., 2014a. Bayesian network approach to multinomial parameter learning using data and expert judgments. International Journal of Approximate Reasoning 55, 1252 – 1268.

Zhou, Y., Fenton, N., Neil, M., 2014b. An extended MPL-C model for Bayesian network parameter learning with exterior constraints, in: van der Gaag, L., Feelders, A. (Eds.), Probabilistic Graphical Models. Springer International Publishing. volume 8754 of *Lecture Notes in Computer Science*, pp. 581–596.

Zhou, Y., Fenton, N., Neil, M., Zhu, C., 2013. Incorporating expert judgement into Bayesian network machine learning, in: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp. 3249–3250.