An Extended MPL-C Model for Bayesian Network Parameter Learning with Exterior Constraints

Yun Zhou[1,2,*], Norman Fenton[1], and Martin Neil[1]

[1] Risk and Information Management (RIM) Research Group,
Queen Mary University of London, United Kingdom
[2] Science and Technology on Information Systems Engineering Laboratory,
National University of Defense Technology, PR China
{yun.zhou,n.fenton,m.neil}@qmul.ac.uk

Abstract. Lack of relevant data is a major challenge for learning Bayesian networks (BNs) in real-world applications. Knowledge engineering techniques attempt to address this by incorporating domain knowledge from experts. The paper focuses on learning node probability tables using both expert judgment and limited data. To reduce the massive burden of eliciting individual probability table entries (parameters) it is often easier to elicit *constraints* on the parameters from experts. Constraints can be interior (between entries of the same probability table column) or exterior (between entries of different columns). In this paper we introduce the first auxiliary BN method (called MPL-EC) to tackle parameter learning with exterior constraints. The MPL-EC itself is a BN, whose nodes encode the data observations, exterior constraints and parameters in the original BN. Also, MPL-EC addresses (i) how to estimate target parameters with both data and constraints, and (ii) how to fuse the weights from different causal relationships in a robust way. Experimental results demonstrate the superiority of MPL-EC at various sparsity levels compared to conventional parameter learning algorithms and other state-of-the-art parameter learning algorithms with constraints. Moreover, we demonstrate the successful application to learn a real-world software defects BN with sparse data.

Keywords: BN parameter learning, Monotonic causality, Exterior constraints, MPL-EC model.

1 Introduction

Bayesian networks have proven valuable in modeling uncertainty and supporting decision making in practice [1]. However, in many applications there is extremely

[*] The authors would like to thank the three anonymous reviewers for their valuable comments and suggestions. This work was supported by European Research Council (grant no. ERC-2013-AdG339182-BAYES-KNOWLEDGE). The first author was supported by China Scholarship Council (CSC)/Queen Mary Joint PhD scholarships and National Natural Science Foundation of China (grant no. 61273322).

L.C. van der Gaag and A.J. Feelders (Eds.): PGM 2014, LNAI 8754, pp. 581–596, 2014.

limited data available to learn either the BN structure or probability tables. In such situations we have to use qualitative knowledge from domain experts in addition to any quantitative data available [2]. There are numerous recent real-world applications in which BN models incorporate significant expert judgment – for example, in medical diagnostics [3,4], traffic incident detection [5] and facial action recognition [6]. However, eliciting expert judgment remains a major challenge.

Directly asking experts to provide quantitative parameter values is time consuming and error-prone because the number of parameters increase exponentially with the number of nodes in the BN. For example, for a node X with 3 states that has 5 parents (each with 2-states), the probability table for X has 32 columns and 3 rows, i.e., 96 probability values to be elicited. Since the columns sum to 1, each column requires only 2 probability values to be elicited, so we consider these as 'parameters' and there are 64 in total. Recent study [7] shows exploring qualitative relationships and their generated constraints would greatly reduce the elicitation burden. However, in applying this method, central challenges include *how to estimate* parameters with both data and constraints [8], *how to optimally perform* expert judgments elicitation [9], and *how to fuse* different weights from different causal relationships and different parent state configurations. These are crucial to ensure that parameter learning is accurate and effective. Despite the finding of qualitative relationships published more than twenty years ago, only limited work [8,10,6,11] has been done on addressing these challenges.

In this paper we assume the BN structure is already defined and only investigate elicited *constraints* on parameters to help learn a target BN with sparse data. The paper extends earlier work [12] in which we introduced an auxiliary BN method (multinomial parameter learning with constraints, which is also referred as MPL-C) for learning parameters given expert constraints and limited data. In that work we considered only parameters constraints restricted to a single probability table column; for example:

"$P(\text{cancer} = \text{true}|\text{smoker} = \text{true}) > 0.01$)" or

"$P(\text{cancer} = \text{true}|\text{smoker} = \text{true}) > P(\text{cancer} = \text{false}|\text{smoker} = \text{true})$"

In this paper we extend this to exterior parameter constraints (across columns) like:

"$P(\text{cancer} = \text{true}|\text{smoker} = \text{true}) > P(\text{cancer} = \text{true}|\text{smoker} = \text{false})$"

This kind of exterior parameter constraints are encoded in monotonic causality between two BN variables [13,14,15]. Parameter learning with this constraints normally is solved via establishing a constrained optimization problem [6,11], and is restricted to assumptions of binary nodes and convex constraints.

Our contribution in this paper is to extend the original MPL-C model (now refered to as MPL-EC) to support parameter learning with both data observations and exterior constraints. In MPL-EC the original parameter estimation problem converts to a BN inference problem. In this way, our model supports either convex or non-convex exterior constraints. Because the MPL-EC is a hybrid BN (contains continuous, as well as discrete, nodes) the inference is achieved via a dynamic discretization junction tree (DDJT) algorithm [16]. Some other works

[17,18,19] also support inference in hybrid BNs with deterministic conditional distributions. In this paper, we mainly focus on building the hybrid BN model to support the parameter learning with exterior constraints. Hence, we will not compare the DDJT with other inference algorithms. In our model, different exterior constraints have different strengths (added as the margin in each inequality [13]), which has a generative equation that encodes the weights from different causal relationships and the weights from different parent state configurations. This is itself an important output for modeling the constraints in a more precise way. To evaluate the algorithm, we conduct experiments on three standard networks, i.e., Weather, Cancer and Asia BNs, comparing against three baselines and prior learning with constraints methods. Finally, we apply our method to parameter learning in a real-world software defects BN.

2 Bayesian Networks Parameter Learning

2.1 Preliminaries

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

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

Let r_i denote the cardinality of the space of X_i, and q_i represent the cardinality of the space of parent configurations of X_i. The k-th probability value of a conditional probability distribution $P(X_i|pa(X_i) = j)$ can be represented as $\theta_{ijk} = P(X_i = k|pa(X_i) = j)$, where $\theta_{ijk} \in \theta$, $1 \le i \le n$, $1 \le j \le q_i$ and $1 \le k \le r_i$. 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 each variable. The classical maximum likelihood estimation (MLE) is to find the set of parameters that maximize the data loglikelihood $l(\theta|D) = \log \prod_l P(D_l|\theta)$. Let N_{ijk} be the number of data records in sample D for which X_i takes its k-th value and its parent $pa(X_i)$ takes its j-th value. Then $l(\theta|D)$ can be rewritten as $l(\theta|D) = \sum_{ijk} N_{ijk} \log \theta_{ijk}$. The MLE seeks to estimate θ by maximizing $l(\theta|D)$. In particular, we can get the estimation of each parameter as follows:

$$\theta^*_{ijk} = \frac{N_{ijk}}{N_{ij}} \qquad (2)$$

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

However, for several cases in the unified model, a certain parent-child state combination would seldom appear, and the MLE learning fails in this situation. Hence, another classical parameter learning algorithm (maximum a posteriori, MAP) can be used to mediate this problem via introducing *Dirichlet* prior: $\theta^* = \arg\max_\theta P(D|\theta)P(\theta)$. Therefore, we can derive the following equation for MAP:

$$\theta^*_{ijk} = \frac{N_{ijk} + \alpha_{ijk} - 1}{N_{ij} + \alpha_{ij} - 1} \tag{3}$$

Intuitively, one can think of the hyperparameter α_{ijk} in *Dirichlet* prior as an experts' guess of the virtual data counts for the parameter θ_{ijk}. When there is no related expert judgments, people usually use uniform prior or BDeu prior [2] in the MAP.

2.2 Constrained Optimization Approach

Although the *Dirichlet* prior is widely used, it is usually difficult to elicit the numerical hyperparameters from experts. Since the ultimate goal of MAP is to infer a posterior distribution, people directly introduce expert provided constraints to regularize the posterior estimation. As discussed above, some related work solves this problem via constrained optimization (CO). In CO, the expert judgments are encoded as convex constraints. For example, based on the previous definition, a convex constraint can be defined as $f(\theta_{ijk}) \leq \mu_{ijk}$, where $f : \Omega_{\theta_{ijk}} \rightarrow R$ is a convex function over θ_{ijk}, and $\mu_{ijk} \in [0, 1]$. Regarding parameter constraints, the scores are computed by a constrained optimization approach (i.e., gradient descent). In detail, for $\forall_{i,j,k}\, \theta_{ijk}$, we maximize the score function $l(\theta|D)$ subject to $g(\theta_{ijk}) = 0$ and $f(\theta_{ijk}) \leq \mu_{ijk}$, where the constraint $g(\theta_{ijk}) = -1 + \sum_{k=1}^{r_i} \theta_{ijk}$ ensures the sum of all the estimated parameters in a probability distribution is equal to one. To model the strength of the constraints, [6] introduced a confidence level λ_{ijk} for the penalty term in the objective function, i.e., let $f(\theta_{ijk}) = \theta_{ijk}$, and the penalty term is defined as $penalty(\theta_{ijk}) = [\mu_{ijk} - \theta_{ijk}]^-$, where $[x]^- = \max(0, -x)$. Therefore, the constrained maximization problem can be rewritten as follows:

$$\begin{array}{c} \arg\max_\theta\ l(\theta|D) - \frac{w}{2} \sum_{ijk} \lambda_{ijk} \cdot penalty(\theta_{ijk})^2 \\ s.t.\ \forall_{i,j,k}\, g(\theta_{ijk}) = 0 \end{array} \tag{4}$$

where w is the penalty weight, which is chosen empirically. Obviously, the penalty varies with the confidence level for each constraint λ_{ijk}. To ensure the solutions move towards the direction of reducing constraint violations (the maximal score), the score 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_{ijk} in the score function.

2.3 Multinomial Parameter Learning with Constraints

Because the basic parameter learning method can be modeled with an auxiliary BN model, the constraints can be easily incorporated as the shared child of the nodes representing the constrained parameters. This auxiliary BN is a *hybrid* model (see Figure 1) containing a mixture of discrete and (non-normally distributed) continuous nodes. Therefore, the parameter estimation problem converts to a BN inference problem, where the data statistics and constraints are observed, and the target parameters are updated by a dynamic discretization inference algorithm [16].

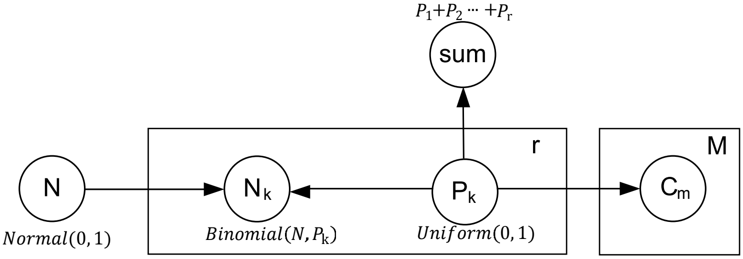


Fig. 1. The multinomial parameter learning model with constraints (MPL-C) and its associated distributions. C_m is a constraint node, which encodes constraints within a NPT column, i.e., $C_1 : P_1 > 0.5$ and $C_2 : P_2 > P_1$.

In Figure 1, for simplification, we use P_k ($k = 1$ to r) to represent the r parameters of a single column instead of θ_{ijk}. Similarly, the N (instead of N_{ij}) represents the data counts of a parent state configuration, and the N_k (instead of N_{ijk}) represent the data counts for its k-th state under this parent state configuration. Given the above model and its related observations, inference refers to the process of computing the discretized posterior marginal of unknown nodes P_k (these are the nodes without evidence). These nodes encode uniform priors, which prevents the problem of random initial values in constrained optimization. After inference, the mean value of P_k will be assigned as the parameter estimation (i.e., the corresponding NPT cell value). Full details can be found in [12].

3 The New Method

In this section we first describe (Section 3.1) the type of monotonic causality and its associated exterior constraints. In Section 3.2 we describe the extended version of the auxiliary BN model to incorporate new forms of exterior constraints provided from expert judgments in order to supplement the MPL-C. Because there is a state combination explosion problem in the extended BN model, we describe a novel alternative BN model which keeps the properties of the original extended BN but with fewer state combinations. In Section 3.3, a simple example is presented to show how to build and apply the MPL-EC model for parameter learning.

3.1 Parameter Constraints

There are two types of node parameter constraints that we consider: interior and exterior. An interior constraint, which is also called inter-relationship constraint, constrains two parameters that share the same node index i, and parent state configuration j (i.e., this is a constraint between values in the same column of a node probability table). An example of such a constraint is $\theta_{ijk} \geq \theta_{ijk'}$, where $k \neq k'$. Interior constraints, which can only be elicited from expert judgment, were studied extensively in our previous work [12]. We showed in [12] that significant improvements to table learning could be achieved from relatively small number of expert provided interior constraints. However, in many situations it is possible (and actually more efficient) to elicit constraints between parameters in different probability table columns. These are the exterior constraints.

Formally, an exterior constraint (also called inter-relationship constraint) is where two parameters in a relative relationship constraint share the same node index i, and state index k. Typically an exterior constraint will have the form: $\theta_{ijk} \geq \theta_{ij'k}$ where $j \neq j'$. This kind of constraint is encoded in monotonic causality which can greatly reduce the burden of expert judgment elicitation. Before we examine exterior constraints in detail, we need some definitions and notations:

The positive/negative monotonic causality: For the simplest single monotonic causal connection: X causes Y ($X \rightarrow Y$), the causality can either be positive or negative. Positive monotonic causality is represented by $X \overset{+}{\rightarrow} Y$ (increasing value of X leads to increases in Y). Negative monotonic causality is represented by $X \overset{-}{\rightarrow} Y$ (increasing value of X leads to decrease in Y); for example, if X is a particular medical treatment and Y is patient mortality.

Let $cdf(\cdot)$ denote the cumulative distribution function. The formal equation of these two kinds of monotonic causality can be formulated as exterior constraints as follows:

$X \overset{+}{\rightarrow} Y$: $cdf(P(Y|pa(Y) = j)) \geq cdf(P(Y|pa(Y) = j'))$
$X \overset{-}{\rightarrow} Y$: $cdf(P(Y|pa(Y) = j)) \leq cdf(P(Y|pa(Y) = j'))$

Here both X and Y are ordered categorical variables, j' and j are integers satisfying the inequality relationships $0 < j' < j < |pa(Y)|$, where the $|pa(Y)|$ represents the total number of state configurations in $pa(Y)$. In $X \rightarrow Y$, $pa(Y) = X$. As we can see, the negative causality represents the opposite causal relationship compared with positive causality. The model of introducing a single positive monotonic causality has been well discussed in previous work [7,13,20]. However, real-world BNs usually contain nodes whose parents provide a mixture of positive and negative causality, as synergistic interactions [11]. Previous work [11] has addressed this synergy problem at some point, where all the causalities should either be positive or negative (homogeneous synergies). Therefore, this work does not allow the synergy relationship have different types of monotonic causalities, which is referred as heterogeneous synergies. Recently, researchers [21,22] introduced a novel canonical gate (refered to as NIN-AND tree) to model different causal interactions: reinforcing and undermining. However, this work does not support learning with monotonic constraints and their margins. Actually, the synergies of different causalities are different when the causal weights (the confidences of the causal connections) are considered. Previous studies rarely discussed this problem, and no relevant model has tackled this issue.

In this paper, we introduce a generative form of the exterior constraint equation, which support homogeneous/heterogeneous synergies with different weights. Assume we have a BN with variables $U = \{Y, X_1, X_2, \ldots, X_n\}$ and the simple inverted naive structure, which means the variable Y is the shared child of $X_1, X_2, \ldots, X_n$. Then our generative exterior constraint is:

$$\begin{cases} cdf(P(Y|pa(Y) = j)) - cdf(P(Y|pa(Y) = j')) \geq M_{jj'} & if\, M_{jj'} > 0 \\ cdf(P(Y|pa(Y) = j)) - cdf(P(Y|pa(Y) = j')) \leq M_{jj'} & if\, M_{jj'} < 0 \end{cases} \tag{5}$$

where $M_{jj'} = \sum_{i=1}^{n} M_{jj'}^{i} = \sum_{i=1}^{n} w_i \cdot cl_i \cdot \varepsilon_{jj'}^{i}$ and $0 < j' < j < |pa(Y)|$. The $M_{jj'}$ represents the overall margin of the synergies, which is the summation of each single margin $M_{jj'}^{i}$. $M_{jj'}^{i}$ contains three terms: $w_i \geq 1$ represent the global weight (the subjective confidence) of the causal relationship $X_i \rightarrow Y$, its default value $w_i = 1$ indicates there is no subjective confidence on the causality; cl is the causality label ($cl_i = 1$ indicates the positive causality $X_i \overset{+}{\rightarrow} Y$; and $cl_i = -1$ represents the negative causality $X_i \overset{-}{\rightarrow} Y$); $\varepsilon_{jj'}^{i}$ is the term that describes the confidence of the inequality introduced by state configuration gap in a causality. That is to say, the $\varepsilon_{jj'}^{i}$ is a small positive value proportional to the state configuration distance in X_i under two indices j and j' in $pa(Y) = \{X_1, X_2, \ldots, X_n\}$.

To calculate $\varepsilon_{jj'}^{i}$, we need to find the subindices ($ind2sub_i(j)$ and $ind2sub_i(j')$) of X_i from the single indices in $pa(Y)$. Thus we have: $\varepsilon_{jj'}^{i} = \frac{ind2sub_i(j) - ind2sub_i(j')}{\lambda \cdot |X_i|}$. Here the $\lambda > 1$ is the trade-off parameter that controls the effect of the confidence introduced by state configuration gap. Because size $|pa(Y)| = \prod_{i=1}^{n} |X_i|$ increases exponentially with an increase of parent nodes, it would be very expensive to find all combinations of two indices in $|pa(Y)|$. Therefore, in this paper, we only discuss a very simple way to get the combinations. For state configuration size $|pa(Y)|$, we generate two indices pairs iteratively ("$|pa(Y)|, 1$", "$(|pa(Y)| - 1), 2$", ...) until no more pairs can be found.

As shown in equation 5, the type ($\geq$ or $\leq$) of the exterior constraint is decided by the value of the margin. The margin is equal to zero ($M = 0$) only in the situation where the effects of different causalities are intermediate in the shared child node. Thus, there is no associated exterior constraints.

Next, we present a simple example of our model: we assume the target variable Y is binary, and it has two binary parents X_1 and X_2 with "T" and "F" states. Assume the first causality is positive $X_1 \overset{+}{\rightarrow} Y$, and the second causality is negative $X_2 \overset{-}{\rightarrow} Y$. Therefore, the exterior constraints induced by these two monotonic causalities can be represented as:

$cdf(P(Y|pa(Y)=4)) - cdf(P(Y|pa(Y)=1)) \geq w_1 \cdot \varepsilon^1_{41} - w_2 \cdot \varepsilon^2_{41}$
$cdf(P(Y|pa(Y)=3)) - cdf(P(Y|pa(Y)=2)) \geq w_1 \cdot \varepsilon^1_{32} - w_2 \cdot \varepsilon^2_{32}$

In addition, there is no subjective judgments on their weights, i.e. $w_1 = w_2 = 1$. Thus the margin of the first equation equal to zero ($w_1 \cdot \varepsilon^1_{41} = w_2 \cdot \varepsilon^2_{41}$), and this equation is discarded. Also, because Y is binary, this means $y = \{y_T, y_F\}$. Therefore, we can have the following exterior constraints based on the above equation:

$P(y_T|x_{1T}, x_{2F}) - P(y_T|x_{1F}, x_{2T}) \geq \frac{1}{\lambda}$
$P(y_T|x_{1T}, x_{2F}) + P(y_F|x_{1T}, x_{2F}) - P(y_T|x_{1F}, x_{2T}) - P(y_F|x_{1F}, x_{2T}) = 0$

Note the equality only happens when y reaches the full range (the biggest) value in $cdf(P(y))$.

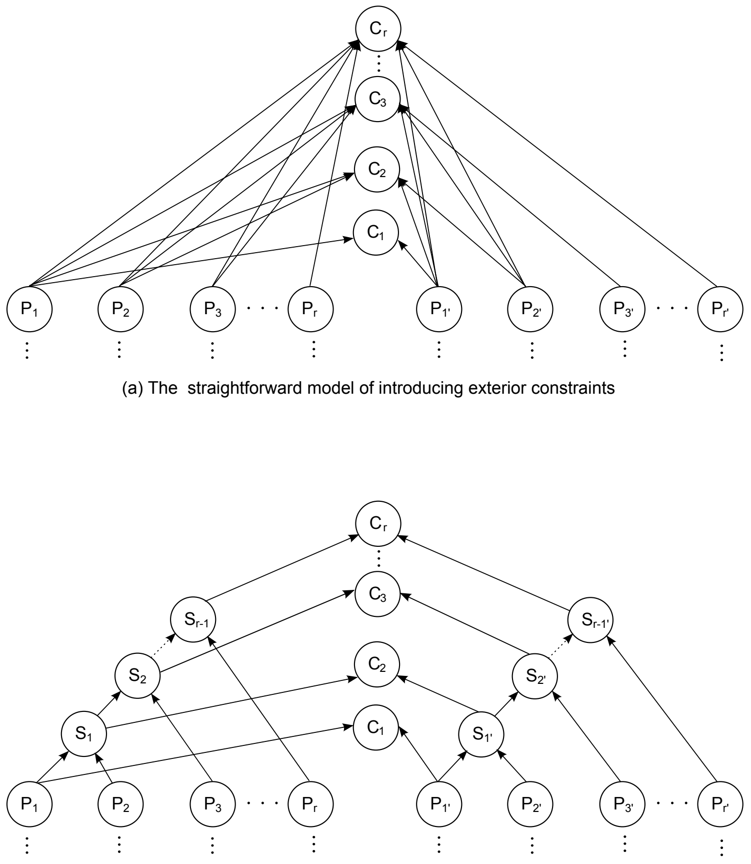
3.2 The Extended MPL-C Model

In this subsection, we present the extended MPL-C model (MPL-EC) to encode the constraints in equation 5. For any monotonic causality, we need to introduce a set of shared children nodes to model the introduced constraints $C_1, C_2, \ldots, C_r$ (see Figure 2). The size of the constraints set is equal to the number of states (ranges from 1 to r) in variable Y. In order to simplify the notation, we use P_k and $P_{k'}$ ($k/k' = 1$ to r/r') to represent parameters in Y under different state configurations of X_i. Therefore, for a single positive monotonic causality $X_i \overset{+}{\rightarrow} Y$, we have the following arithmetic constraints encoded in the MPL-EC model to constrain the parameters under two state configurations of X_i (j and j'):

$$\begin{cases} C_1: P_1 - P_{1'} \geq w_i \cdot cl_i \cdot \varepsilon^i_{jj'} \\ C_2: P_1 + P_2 - P_{1'} - P_{2'} \geq w_i \cdot cl_i \cdot \varepsilon^i_{jj'} \\ C_3: P_1 + P_2 + P_3 - P_{1'} - P_{2'} - P_{3'} \geq w_i \cdot cl_i \cdot \varepsilon^i_{jj'} \\ \vdots \\ C_r: \sum_{k=1}^{r} P_k - \sum_{k'=1}^{r} P_{k'} = 0 \end{cases} \tag{6}$$

In the last exterior constraint equation C_r, two sides of the relative relationship are equal to each other. As we can see there are additional $(1+n) \cdot n$ edges when we introduce n constraint nodes. To reduce the model complexity it must be replaced by an equivalent model whose structure has a restricted number of parents.

Previous work has proposed a binary factorization algorithm [23] to improve the efficiency of the DDJT algorithm. This idea can also be applied here to produce an alternative model of the straightforward MPL-EC. The new model is called binary summation model, which introduces an additional $2 \cdot (n-1)$ auxiliary nodes, which only encode the simple sum arithmetic equations to model the summations of its parents. This model has the same number of edges as the straightforward model, but the maximal number of parents is fixed as two in this model. This avoids the parent state combination explosion problem. The detail of its structure can be found in Figure 2(b).



(a) The straightforward model of introducing exterior constraints

(b) The binary summation model of introducing exterior constraints

Fig. 2. The straightforward MPL-EC model and its alternative binary summation model. Due to the space limitation, the MPL-EC model presented here only display the part for modeling introduced constraints, the left part for modeling multinomial parameter learning is not displayed, which is the same as MPL-C in Section 2.3.

3.3 A Simple Example

In this subsection, we use a simple example to demonstrate the exterior constraints and its generated MPL-EC model. This example encodes the simplest single positive causal connection: $X \overset{+}{\rightarrow} Y$, where the two nodes involved are both binary with "T" and "F" states. Therefore, we have two parameter columns under two parent state instantiations to estimate in Y, which are $P(Y|x_T)$ and $P(Y|x_F)$. Its MPL-EC model is shown in Figure 3.

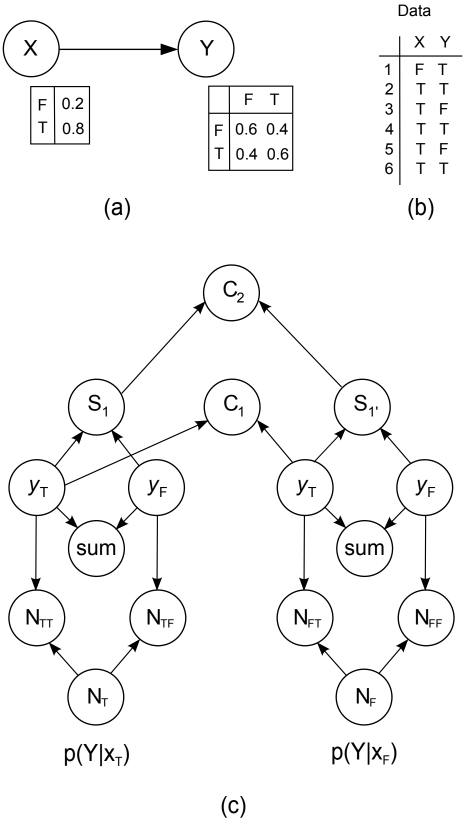


Fig. 3. The original BN and its training data in the simple example. (a) The DAG and its associated NPTs. (b) The 6 data records for the two variables in the BN. (c) The MPL-EC model for estimating the parameters in Y. The constraint nodes are modeled as binary (True/False) nodes with expressions that specify the constraint relationships between its parents. The auxiliary BN model is implemented in AgenaRisk [24], which supports hybrid BNs containing conditionally deterministic expressions. For example, the software statement for C_1 is: $if(P(y_T|x_T) - P(y_T|x_F) \geq w \cdot cl \cdot \varepsilon$, "$True$", "$False$").

The detail of the exterior constraints encoded in the constraint nodes of MPL-EC is:

$$\begin{cases} S_1 : \ P(y_T|x_T) + P(y_F|x_T) \\ S_{1'} : \ P(y_T|x_F) + P(y_F|x_F) \\ C_1 : \ P(y_T|x_T) - P(y_T|x_F) \geq w \cdot cl \cdot \varepsilon \\ C_2 : \ S_1 - S_{1'} \geq w \cdot cl \cdot \varepsilon \end{cases} \tag{7}$$

where $cl = 1$, $\varepsilon = \frac{1}{2\lambda}$. according to above definition, and w represents the subjective confidence whose value can be chosen empirically from the domain knowledge.

Based on the statistics on the dataset (Figure 3(b)) and previous definition, we have: $N_T = 5$ ($N_{TF} = 2$, $N_{TT} = 3$) under the condition of $X = x_T$, and $N_F = 1$ ($N_{FF} = 0$, $N_{FT} = 1$) in the state initiation of $X = x_F$. Therefore, the MLE results of Y are $P(y_T|x_T) = 0.6$ and $P(y_T|x_F) = 1$. As we can see, the estimation of $P(y_T|x_F)$ is far away from the ground truth (0.6 and 0.4) due to the sparse data records under the $X = x_F$ condition.

With the above data observations, we now can set the evidence for certain nodes including constraint nodes (all are set as "True" observations), number of trials, total numbers, and the summation of all the estimated parameters. Based on these evidences, the inference in the MPL-EC is to compute the discretized posterior marginals of each of the unknown nodes $y_{F/T}$ (these are the nodes without evidence) via DDJT algorithm [16]. This algorithm alternates between two steps: 1) performing dynamic discretization, which searches and splits the regions with the highest relative entropy error determined by a bounded K-L divergence with the current approximated estimates of the marginals; 2) performing junction tree inference, which updates the posterior of the marginals. At convergence, the mean value of $y_{F/T}$ will be assigned as the final corresponding NPT cell values. After inference with the model in Figure 3(c), we have $P(y_T|x_T) = 0.67$ and $P(y_T|x_F) = 0.50$, which are much reasonable than the MLE results.

4 Experiments

The goal of the experiments is to demonstrate the benefits of our method and show the advantages of using elicited signs of causalities (either from ground truth or from expert judgment) and their generated exterior constraints to improve the parameter learning performance. We test the method against the conventional learning techniques (MLE and MAP) as well as against the competing method that incorporates exterior constraints (i.e., the constraint optimization method). Sections 4.1 and 4.2 describe the details of the experiments. The first (Section 4.1) uses the well-known Weather, Cancer and Asia BN (their signs are elicited from the ground truth), while the second (Section 4.2) uses a software defects BN, and its signs of causalities are elicited from a real expert.

In all cases, we assume that the structure of the model is known and that the 'true' NPTs that we are trying to learn are those that are provided as standard with the models. Obviously, for the purpose of the experiment we are not given these 'true' NPTs but instead are given a number of sample observations which are randomly generated based on the true NPTs. The experiments consider a range of sample sizes. In all case the resulting learnt NPTs are evaluated against the true NPTs by using the K-L divergence measure [25], which is recommended to measure the distance between distributions. The smaller the K-L divergence is, the closer the estimated NPT is to the true NPT. If frequency estimated values are zero in MLE, Laplace smoothing is applied to guarantee they can be computed. The global weights of all causal relationships are set as default value $w_i = 1$ in all experiment settings, and the trade-off value λ is set as 10.

4.1 Different Standard BNs Experiments

In the first set of experiments we use three standard models [26,27,28] that have been widely used for evaluating different learning algorithms. Based on these BNs and elicited signs, we compare the performance of different parameter learning algorithms: MLE, MAP, CO and MPL-EC.

Table 1 shows the structure of each BN and its associated parameter learning results. The BN structures are presented in the middle column of the table and annotated with positive/negative signs on their edges. The learning results in each setting are presented in the last column for each row. In each sub figure, the x-coordinate denotes the data sample size from 10 to 100, and the y-coordinate denotes the average K-L divergence for each parameter. For each data sample size, the experiments are repeated 5 times, and the results are presented with their mean and standard deviation.

As shown in the last column of Table 1, for all parameter learning methods, the K-L divergence decreases as expected when the sample size increases. Specifically, methods of learning with constraints, i.e., CO and MPL-EC always outperform the conventional MLE algorithm, especially in the sparse data situations. However, the CO failed to outperform MAP in all data settings of Cancer and Asia BNs, while the MPL-EC method always achieves the best performance in all cases for the three different BNs.

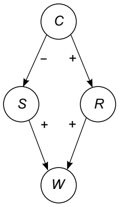
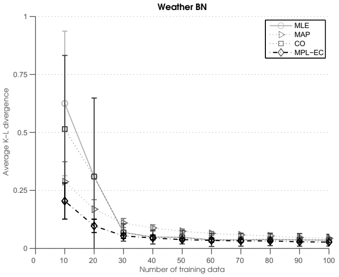
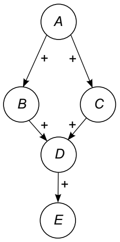
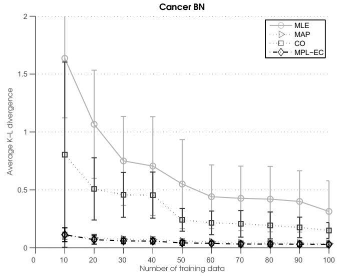
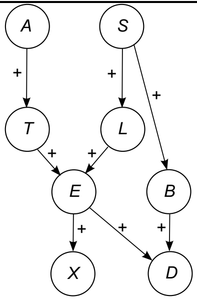
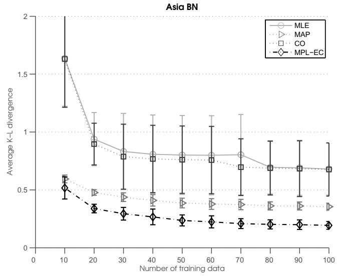
4.2 Software Defects BN Experiment

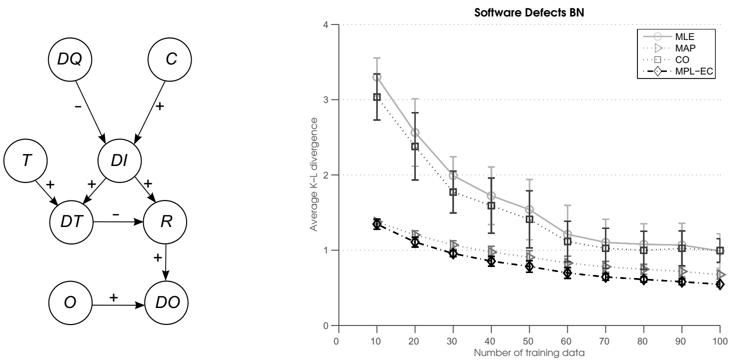
In this section, we consider a very well documented BN model that has been used by numerous technology companies worldwide [29] to address a real-world problem: the software defects prediction problem. The idea is to be able to predict the quality of software in terms of defects found in operation based on observations that may be possible during the software development (such as component complexity and defects found in testing). This BN contains eight nodes: “design process quality (DQ)”, “component complexity (C)”, “defects inserted (DI)”, “testing quality (T)”, “defects found in testing (DT)”, “residual defects (R)”, “operation usage (O)” and “defects found in operation (DO)”. All of them are discrete, which have 3 ordered states: “Low”, “Medium”, and “High”.

Figure 4(a) represents the structure of the BN, the signs on the edges indicate whether the associate monotonic causalities are positive or negative. These causalities are elicited from real expert judgments, i.e., as design process quality (DQ) goes from “Low” to “High”, the defects inserted (DI) go from “High” to “Low”, this encodes a negative monotonic causality.

Figure 4(b) shows the learning results, where the MPL-EC outperforms all other algorithms in every scenario. Compared with the state-of-art CO algorithm, our MPL-EC significantly improves the parameter learning performance, i.e., the MPL-EC outperforms the CO in all training sample sizes, with an overall 47.06% K-L divergence reduction.

Table 1. Learning results for MLE, MAP, CO and MPL-EC in Weather, Cancer and Asia BN learning problems. Four lines are presented in each sub figure, where the solid line with circle marker represents the learning results of baseline MLE algorithm, the dotted line with right-pointing triangle marker represents the learning results of MAP algorithm, the dotted line with square marker denotes the results of the CO algorithm, and the bold dash-dot line with diamond marker shows the learning results of the MPL-EC method.

Name	Directed acyclic graph (DAG)	Learning performance
Weather		
Cancer		
Asia		



(a) (b)

Fig. 4. The Learning results for MLE, MAP, CO and MPL-EC in software defects BN learning problem: (a) The DAG and real elicited exterior constraints; (b) The details of the learning results for different training data sample sizes

5 Conclusions

When data is sparse, purely data driven BN learning is inaccurate. Our framework tackles this problem by leveraging a set of exterior constraints elicited from experts. Our model is an auxiliary BN, which encodes all the information (i.e., data observations, parameters we wish to learn, and exterior constraints encoded in monotonic causalities) in parameter learning. By converting the parameter learning problem into a Bayesian inference problem, we are able to perform robust and effective parameter learning even with heterogeneous monotonic causalities and zero data observations in some cases. Our approach applies with categorical variables, and is robust to any degree of data sparsity. Standard BNs experiments show that MPL-EC consistently outperforms the conventional methods (MLE and MAP) and former learning with constraints algorithms. Finally, experiments with a real-world software defects network show the practical value of our method. In future work we will investigate the extension to the continuous variables, and integrating expert constraints with structure learning so structure can also be refined.

References

1. Fenton, N., Neil, M.: Risk Assessment and Decision Analysis with Bayesian Networks. CRC Press, New York (2012)
2. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: The combination of knowledge and statistical data. Mach. Learn. 20(3), 197–243 (1995)
3. Hutchinson, R.A., Niculescu, R.S., Keller, T.A., Rustandi, I., Mitchell, T.M.: Modeling fMRI data generated by overlapping cognitive processes with unknown onsets using hidden process models. NeuroImage 46(1), 87–104 (2009)

4. Yet, B., Perkins, Z., Fenton, N., Tai, N., Marsh, W.: Not just data: A method for improving prediction with knowledge. J. Biomed. Inform. 48, 28–37 (2014)
5. Šingliar, T., Hauskrecht, M.: Learning to detect incidents from noisily labeled data. Mach. Learn. 79(3), 335–354 (2010)
6. Liao, W., Ji, Q.: Learning Bayesian network parameters under incomplete data with domain knowledge. Pattern Recogn 42(11), 3046–3056 (2009)
7. Wellman, M.P.: Fundamental concepts of qualitative probabilistic networks. Artif. Intell. 44(3), 257–303 (1990)
8. Druzdzel, M.J., Van Der Gaag, L.C.: Elicitation of probabilities for belief networks: Combining qualitative and quantitative information. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc, pp. 141–148. Morgan Kaufmann, San Francisco (1995)
9. Cano, A., Masegosa, A.R., Moral, S.: A method for integrating expert knowledge when learning Bayesian networks from data. IEEE Trans. on Sys. Man Cyber. Part B 41(5), 1382–1394 (2011)
10. Niculescu, R.S., Mitchell, T., Rao, B.: Bayesian network learning with parameter constraints. J. Mach. Learn. Res. 7, 1357–1383 (2006)
11. Yang, S., Natarajan, S.: Knowledge intensive learning: Combining qualitative constraints with causal independence for parameter learning in probabilistic models. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) ECML PKDD 2013, Part II. LNCS, vol. 8189, pp. 580–595. Springer, Heidelberg (2013)
12. Zhou, Y., Fenton, N., Neil, M.: Bayesian network approach to multinomial parameter learning using data and expert judgments. Int. J. Approx. Reasoning 55(5), 1252–1268 (2014)
13. Altendorf, E.E.: Learning from sparse data by exploiting monotonicity constraints. In: Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, pp. 18–26. Morgan Kaufmann Publishers Inc., San Francisco (2005)
14. van der Gaag, L.C., Renooij, S., Geenen, P.L.: Lattices for studying monotonicity of Bayesian networks. In: Proceedings of the 3rd European Workshop on Probabilistic Graphical Models, Prague, Czech Republic, pp. 99–106 (2006)
15. van der Gaag, L.C., Tabachneck-Schijf, H.J.M., Geenen, P.L.: Verifying monotonicity of bayesian networks with domain experts. Int. J. Approx. Reasoning 50(3), 429–436 (2009)
16. Neil, M., Tailor, M., Marquez, D.: Inference in hybrid Bayesian networks using dynamic discretization. Stat. and Comput. 17(3), 219–233 (2007)
17. Lunn, D.J., Thomas, A., Best, N., Spiegelhalter, D.: WinBUGS-a Bayesian modelling framework: concepts, structure, and extensibility. Stat. and Comput. 10(4), 325–337 (2000)
18. Shenoy, P.P., West, J.C.: Inference in hybrid Bayesian networks using mixtures of polynomials. Int. J. Approx. Reasoning 52(5), 641–657 (2011)
19. Shenoy, P.P.: Two issues in using mixtures of polynomials for inference in hybrid Bayesian networks. Int. J. Approx. Reasoning 53(5), 847–866 (2012)
20. Feelders, A., van der Gaag, L.: Learning Bayesian network parameters under order constraints. Int. J. Approx. Reasoning 42(1), 37–53 (2006)
21. Xiang, Y., Jia, N.: Modeling causal reinforcement and undermining for efficient CPT elicitation. IEEE Trans. on Knowl. and Data Eng. 19(12), 1708–1718 (2007)
22. Xiang, Y., Truong, M.: Acquisition of causal models for local distributions in Bayesian networks. IEEE Trans. on Cyber (2013), doi:10.1109/TCYB.2013.2290775

23. Neil, M., Chen, X., Fenton, N.: Optimizing the calculation of conditional probability tables in hybrid Bayesian networks using binary factorization. IEEE Trans. on Knowl. and Data Eng. 24(7), 1306–1312 (2012)
24. AgenaRisk (2014), `http://www.agenarisk.com/`
25. Cover, T.M., Thomas, J.: Entropy, relative entropy and mutual information. Elements of Information Theory, 12–49 (1991)
26. Lauritzen, S., Spiegelhalter, D.: Local computations with probabilities on graphical structures and their application to expert systems. J. Roy. Statist. Soc. Ser. B, 157–224 (1988)
27. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Mach. Learn. 29(2-3), 131–163 (1997)
28. Korb, K.B., Nicholson, A.E.: Bayesian Artificial Intelligence. CRC Press, New York (2003)
29. Fenton, N., Neil, M., Marsh, W., Hearty, P., Radliński, Ł., Krause, P.: On the effectiveness of early life cycle defect prediction with Bayesian nets. Empirical Softw. Engg. 13(5), 499–537 (2008)