Improving Bayesian Network Structure Learning using Heterogeneous Experts

Hossein Amirkhani[1], Arjen Hommersom[2], Mohammad Rahmati[1], and Peter Lucas[2]

[1] Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran
{amirkhani,rahmati}@aut.ac.ir
[2] Institute for Computing and Information Sciences, Radboud University, Nijmegen, The Netherlands
{arjenh,peterl}@cs.ru.nl

Abstract. We consider the problem of learning Bayesian network structures by exploiting both data as well as experts' opinions about the graph. In practice, experts will have different individual probabilities of correctly labelling the inclusion or exclusion of edges of the network structure. Therefore, we propose in this paper to estimate the accuracy of experts and then exploit this accuracy during the learning of the structure. We use an expectation maximization (EM) algorithm to estimate the accuracies, considering the true structure as the hidden variable. As a second contribution, we develop a Bayesian score that considers the training data as well as the experts' opinions to score different possible structures, and then a greedy search is done in the space of possible structures. The experimental results demonstrate the effectiveness of considering the experts' accuracies in improving the accuracy of the predicted structures.

Keywords: Bayesian networks. Structure learning. Expectation maximization. Heterogeneous experts. Knowledge based Bayesian score.

1 Introduction

In many fields, regression modelling has long been seen as a standard method for identifying statistical associations. The use of generalizations of regression models such as Bayesian networks (BNs) is now happening in many fields such as epidemiology [1]. They offer a richer framework for identifying structure in a model. While BNs have been studied as predictive frameworks for single and multi-label classifiers, in epidemiology one is normally more interested in how all the variables are related to each other. This is a multi-target prediction problem involving multiple variables and their relationships.

A common way to look at Bayesian network learning is to identify edges of an acyclic directed graph in such a way that the total network is optimal according to a given score; in many papers the score corresponds to the sum of marginal likelihood of nodes given their parents as represented by the BDe metric [2]. However, identifying the structure between random variables is a hard task. The first problem is that in many real-world domains, there is few data or the data is noisy, so that the score that is being used is not always reliable. A further complication is that many structures are likelihood equivalent, meaning that they cannot be distinguished based on the data. This latter point is especially relevant if one tries to learn models that are understandable to experts, in particular in such a way that they are given a causal interpretation.

Given these limitations of Bayesian network structure learning, some researchers have proposed the use of experts for building models. For example, [3] describes the construction of the model structure for therapy selection for the treatment of oesophagus cancer. Such a process requires a significant amount of work, includes many sessions with experts and typically a lot of preparation time. Moreover, such a process typically cannot systematically deal with conflicting opinions between experts.

In this paper, we propose a new method to combine knowledge from multiple experts with data to learn a Bayesian network structure, building upon an approach proposed by Richardson and Domingos [4]. The main advantages of their model are that (i) experts only have to label some of the edges (included in the graph, or not), (ii) can deal with conflicting between experts, and (iii) data can be used to fill in the gaps in the existing knowledge such as if none of the experts have an opinion about an edge.

However, a limitation of Richardson and Domingos' research is that they assume that all experts have an equal probability to correctly label the edge, which clearly is an unrealistic assumption. In this paper, we extend their work by learning the accuracies of different experts. To this end, we propose an expectation maximization algorithm, where the true structure is taken as a hidden variable. Subsequently, we use this knowledge on the accuracy of experts in the Bayesian network learning process, using a novel Bayesian score that not only takes into account the data, but also the opinions of heterogeneous experts.

The rest of this paper is organized as follows. Section 2 gives the preliminaries regarding the BN structure learning. In Section 3, we provide a description of our accuracy estimate method and our knowledge based score. Section 4 presents the experimental results. Finally, the paper concludes in Section 5.

2 Preliminaries

The structure of a BN is a directed acyclic graph (DAG) $G = (V, E)$, where V is the set of variables and $E \subseteq V \times V$ is the set of edges. An edge $X \to Y$ corresponds to a direct relationship between X and Y. This means that there is no set $\boldsymbol{Z} \subseteq V \setminus \{X, Y\}$ such that X is independent of Y given $\boldsymbol{Z}$. On the other hand, BN structures often can be given a causal meaning, where an edge $X \to Y$ is interpreted as X “causes” Y [5].

The problem of learning the structure of a BN from a training set D is to find the DAG that, in some sense, best matches D. To solve this problem, many researchers use the “score and search” approach [2, 6–9]. This approach scores each DAG using a scoring function and attempts to find the DAG that maximizes this score.

A widely-used score and search algorithm for learning the BN structures is described by Heckerman et al. [2]. They use the *Bayesian Dirichlet* (BD) score

$$P(G,D) = P(G)P(D|G) = P(G) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(n'_{ij})}{\Gamma(n'_{ij}+n_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(n'_{ijk}+n_{ijk})}{\Gamma(n'_{ijk})} \tag{1}$$

where $P(G)$ is the prior probability of the structure, n is the number of variables, $\Gamma()$ is the gamma function, q_i is the number of states of the Cartesian product of the parents of i^{th} node, r_i is the number of states of the i^{th} node, n_{ijk} is the number of occurrences of the k^{th} state of i^{th} node with the j^{th} state of its parents, and $n_{ij} = \sum_{k=1}^{r_i} n_{ijk}$. Finally, n'_{ijk} are the parameters of the Dirichlet prior distribution, and $n'_{ij} = \sum_{k=1}^{r_i} n'_{ijk}$.

Using the likelihood equivalence assumption, which says that equivalent structures must have the same score, Heckerman et al. constrain the values that n'_{ijk} can take on and provide a method to calculate these parameters. With this method, the user provides a prior Bayesian network and an equivalent sample size n' that says how confident they are in it. The resulting scoring criterion is named BDe (Bayesian Dirichlet with likelihood equivalence). With a further constraint, such that all configurations are equally likely, $n'_{ijk} = n'/r_i q_i$ and the criterion is named BDeu (Bayesian Dirichlet with likelihood equivalence and a uniform joint distribution).

Heckerman et al. use a greedy search in the DAG space. At each step, the algorithm generates all neighbors of the current network that can be obtained by adding, deleting or reversing a single edge, without creating cycles, and selects the best one. The search ends when no neighbor achieves a higher score than the current network.

3 Proposed Method

3.1 Accuracy Estimation

Assume that the structure has n nodes. Therefore, there are $N = n(n-1)/2$ different node pairs in the structure. We indicate the number of experts by R and the experts' predictions regarding these node pairs are collected in an $R \times N$ matrix, denoted by O, where $O(i,j) \in \{\emptyset, \rightarrow, \leftarrow, \nrightarrow\}$. $O(i,j) = '\emptyset'$ means that the i^{th} expert has not provided any prediction about the j^{th} pair. $' \rightarrow '$ and $' \leftarrow '$ indicate the inclusion of an edge in a particular direction, and $' \nrightarrow '$ means the exclusion of edges.

We model the accuracy of each expert by a 3×3 confusion matrix:

Prediction

True class	→	←	$\nrightarrow$
→	$\pi_{1,1}$	$\pi_{1,2}$	$\pi_{1,3}$
←	$\pi_{2,1}$	$\pi_{2,2}$	$\pi_{2,3}$
$\nrightarrow$	$\pi_{3,1}$	$\pi_{3,2}$	$\pi_{3,3}$

If all confusion matrices are collectively denoted by $\mathbf{\Pi}$, the maximum likelihood estimate of $\mathbf{\Pi}$, with the independence assumption, is

$$\mathbf{\Pi}_{MLE} = \mathrm{argmax}_{\mathbf{\Pi}}\{\log \Pr(O|\mathbf{\Pi})\} \ = \mathrm{argmax}_{\mathbf{\Pi}} \left\{\sum_{\substack{1\le i\le R, 1\le j\le N \\ O(i,j)\neq\emptyset}} \log \Pr(O(i,j)|\mathbf{\Pi})\right\} \qquad (2)$$

To solve this optimization problem, we consider the true structure as a hidden variable and use the EM algorithm. We model the true structure as an $N \times 3$ matrix T, where each row is related to one pair of variables. Columns of this matrix are labeled by $\{\rightarrow, \leftarrow, \not-\}$. According to the status of each pair, the corresponding column in the related row is equal to one and the other two elements are zero.

The EM algorithm iterates between two steps: an Expectation (E)-step and a Maximization (M)-step. In the E-step, it computes a new estimate of the expectation of the hidden variable given the current estimate of the model parameters and available experts' opinions. In the M-step, it uses the current estimate of the expectation of the hidden variable to compute a new estimate of the model parameters by maximizing the conditional expectation.

In addition to the confusion matrices, we also have the prior probabilities as model parameters. The prior probability of each element of $\{\rightarrow, \leftarrow, \not-\}$ is the probability of that element for a randomly selected pair prior to viewing the opinions. We denote the model parameters by $\boldsymbol{\theta} = \{\mathbf{\Pi}, p_1, p_2, p_3\}$, where $\mathbf{\Pi}$ denotes the confusion matrices, and p_1, p_2, and p_3 are the prior probabilities of $\{\rightarrow, \leftarrow, \not-\}$.

Since $T(j,k)$ is a binary variable, in the E-step we have:

$$\mathbb{E}\left[T(j,k)\middle|O,\boldsymbol{\theta}^{(t)}\right] = \Pr\left(T(j,k) = 1\middle|O,\boldsymbol{\theta}^{(t)}\right) = \frac{\Pr\left(T(j,k)=1\middle|\boldsymbol{\theta}^{(t)}\right)\Pr\left(O\middle|T(j,k)=1,\boldsymbol{\theta}^{(t)}\right)}{\Pr\left(O\middle|\boldsymbol{\theta}^{(t)}\right)}. \qquad (3)$$

where $\boldsymbol{\theta}^{(t)}$ is the current estimate of the model parameters. $\Pr\left(T(j,k) = 1\middle|\boldsymbol{\theta}^{(t)}\right)$ is simply the prior probability of the k^{th} element in $\{\rightarrow, \leftarrow, \not-\}$ which can be obtained from $\boldsymbol{\theta}^{(t)}$. In addition, by assuming that the opinions are independent, we can simply compute $\Pr\left(O\middle|T(j,k) = 1,\boldsymbol{\theta}^{(t)}\right)$ using the confusion matrices available in $\boldsymbol{\theta}^{(t)}$. Finally, $\Pr\left(O\middle|\boldsymbol{\theta}^{(t)}\right)$ is a normalization factor that can be computed in a straightforward manner.

In the M-step, a new estimate of the model parameters is computed by the following maximization:

$$\boldsymbol{\theta}^{(t+1)} = \mathrm{argmax}_{\boldsymbol{\theta}}\left\{\mathbb{E}\left[\log \Pr(O,T|\boldsymbol{\theta}) \ \middle|O,\boldsymbol{\theta}^{(t)}\right]\right\} \qquad (4)$$

By equating the partial derivatives to zero, we obtain the following estimate for the prior probabilities:

$$p_k^{(t+1)} = \frac{1}{\sum_{1\le j\le N} n_j}\sum_{1\le j\le N} n_j \times \mathbb{E}\left[T(j,k)\middle|O,\boldsymbol{\theta}^{(t)}\right], \qquad (5)$$

where $p_k^{(t+1)}$ is the next estimate for the prior probability of the k^{th} element in $\{\rightarrow, \leftarrow, \not-\}$, n_j is the number of experts that have expressed their opinions about the j^{th} pair, and $\mathbb{E}\left[T(j,k)\middle|O,\boldsymbol{\theta}^{(t)}\right]$ has been computed during the E-step.

Also, the next estimate for the confusion matrix elements of the i^{th} expert is:

$$\pi_{k,r}^{(t+1)} = \frac{\sum_{1\leq j\leq N}\mathbb{E}[T(j,k)|O,\boldsymbol{\theta}^{(t)}]\times\delta(\mathrm{O}(i,j),r)}{\sum_{l\in\{\rightarrow,\leftarrow,\not\rightarrow\}}\sum_{1\leq j\leq N}\mathbb{E}[T(j,k)|O,\boldsymbol{\theta}^{(t)}]\times\delta(O(i,j),l)} \quad (6)$$

where δ is the Kronecker delta function.

We start the EM algorithm with initializing the expectation of the hidden variable T and continue with the M-step. The initial value of $\mathbb{E}[T(j,k)]$ is set to the ratio of available opinions about the j^{th} pair that are equal to the k^{th} element of $\{\rightarrow,\leftarrow,\not\rightarrow\}$.

3.2 Knowledge Based Bayesian Score

To score a candidate DAG G, we can consider $P(G|D,O)$ as a reasonable measure, where D is the training data and O is the matrix including the experts' opinions. It is obvious that

$$P(G|D,O) \propto P(G,D,O) = P(G)P(D|G)P(O|D,G) \quad (7)$$

It is reasonable to assume that given G, the training data and the experts' opinions are independent, and therefore we have $P(O|D,G) = P(O|G)$.

We define our knowledge based scoring function as

$$\mathrm{Score_{KB}}(G;D,O) = \log P(G) + \log P(D|G) + \log P(O|G) \quad (8)$$

Therefore, the KB score has three parts: a prior part, a data part, and a knowledge part. For the prior $P(G)$, inspired from [4], we assume that each pair of nodes independently has some prior probability p_0 of being connected by an edge in a given direction. Therefore, $P(G) = \prod_{j=1}^{N} P(s_j)$, where s_j is the j^{th} node pair, and $P(s_j) = p_0$ if it includes an edge in any direction, and $P(s_j) = 1 - 2p_0$ if it lacks an edge.

For the data part $P(D|G)$, we use the likelihood part of the BDeu score. Finally, for the knowledge part $P(O|G)$, we use the maximum likelihood estimate of the confusion matrices $\mathbf{\Pi}_{MLE}$, and compute $P(O|G,\mathbf{\Pi}_{MLE})$ assuming that the opinions are independent given the structure and the confusion matrices.

4 Experiments

In order to illustrate the effectiveness of the proposed method, we use two standard BNs: 1) the 'Insurance' network with 27 variables and 52 edges, and 2) the 'Alarm' network with 37 variables and 46 edges. For these BNs, the experts' opinions are simulated. We use three parameters for each expert to generate his/her confusion matrix: the probability of correctly selecting the existing edges, the probability of inversely selecting the existing edges, and the probability of correctly selecting the absent edges. We denote these parameters by α_1, α_2, and α_3, respectively.

We generate two different populations each with $R = 10$ experts, and label them as 'Worse' and 'Better'. Table 1 lists the parameters assigned to each expert in these populations, as well as the average parameters of each population. There are six experts that are equally accurate in both populations. The parameters of these experts are selected from the whole range of possible values. The other four experts have higher accuracies in the 'Better' population than the 'Worse' population.

A parameter $\beta \in [0,1]$ controls the number of total opinions provided by experts. This parameter indicates the ratio of opinions provided by all experts. Since the maximum possible number of opinions is $R \times N$, the number of opinions in a particular experiment is $\beta \times R \times N$. We use $\beta \in \{0.3, 0.4, 0.5, 0.6\}$ in our experiments.

We generate 1000 data samples from the networks in each experiment. In order to reduce the effect of randomness (in the simulated opinions and generated data samples), we repeat each experiment 10 times and report the average results.

For the evaluation, two different measures are reported:

- Structural Hamming Distance (SHD) which is equal to the number of edge deviations (missing plus additional plus orientation errors) between the learned structure and the gold-standard network (lower is better).
- Not Detected Ratio (NDR) which is equal to the ratio of edges not detected by the algorithm (or detected with the wrong orientation) in comparison to the gold-standard network (lower is better).

We compare our algorithm with four scenarios:

- Only data, neglecting the knowledge part of the KB score. This scenario is similar to the approach of the methods which use the BDeu score [2].
- Only knowledge, neglecting the data part of the KB score.
- Best equal confusions, considering both data and knowledge parts in the KB score, but using equal confusion matrices for all experts. We use the best estimable confusion matrix by comparing the opinions with the gold-standard networks. This is the best achievable outcome from the Richardson and Domingos' method [4].
- Best multiple confusions, considering both data and knowledge parts in the KB score, but using the best estimable confusion matrices for each expert. The best estimable confusion matrix of an expert is obtained by comparing his/her opinions with the gold-standard network. The results of this scenario are the best achievable outcomes from our method.

As the search procedure, we use the same greedy search as Heckerman et al. [2] introduced in Section 2, starting from an empty network. Finally, we set p_0 in the prior $P(G)$ to 0.1 and equivalent sample size in $P(D|G)$ to 1.

Table 1. The parameters assigned to each expert in the simulated populations.

		1	2	3	4	5	6	7	8	9	10	**Mean**
Worse	$\boldsymbol{\alpha_1}$	0.3	0.2	0.15	0.7	0.9	0.75	0.4	0.6	0.45	0.55	0.5
	$\boldsymbol{\alpha_2}$	0.3	0.15	0.8	0.2	0.05	0.1	0.25	0.3	0.35	0.2	0.27
	$\boldsymbol{\alpha_3}$	0.3	0.95	0.85	0.7	0.8	0.9	0.5	0.65	0.45	0.6	0.67
Better	$\boldsymbol{\alpha_1}$	0.3	0.2	0.15	0.7	0.9	0.75	0.85	0.8	0.7	0.75	0.61
	$\boldsymbol{\alpha_2}$	0.3	0.15	0.8	0.2	0.05	0.1	0.05	0.1	0.15	0.15	0.21
	$\boldsymbol{\alpha_3}$	0.3	0.95	0.85	0.7	0.8	0.9	0.85	0.9	0.8	0.7	0.78

Results obtained by the ‘Worse’ and ‘Better’ populations for the ‘Insurance’ network are presented in Table 2 and Table 3, respectively. Also, Table 4 and Table 5 display the results obtained for the ‘Alarm’ network. It is obvious that in the majority of cases, our method outperforms the ‘Only Data’ and ‘Only Knowledge’ scenarios. Therefore, the KB score can effectively utilize both the training data and the experts’ opinions for scoring different structures. In addition, in all cases, the ‘Best Multiple Confusions’ scenario outperforms the ‘Best Equal Confusions’ scenario. Therefore, we conclude that considering different accuracies for different experts is a promising idea for improving the learned network structures.

5 Conclusion

In this paper we developed a new method for learning Bayesian network structures taking into account labelled data about the existence of edges given by heterogeneous experts, i.e., experts with different levels of accuracy in labelling. Our preliminary experiments show that, if experts are reasonably accurate, this new method improves upon learning Bayesian networks from data or expert knowledge alone. Additionally, in such reasonable settings, estimating the accuracy of each individual expert improves upon a method where a fixed accuracy among expert is assumed, even if we would know the exact average accuracy of the experts. In conclusion, our method seems to be a promising approach for learning Bayesian network from both data and expert knowledge.

Table 2. The results obtained by the ‘Worse’ population for the ‘Insurance’ network.

β	0.3		0.4		0.5		0.6	
	SHD	NDR	SHD	NDR	SHD	NDR	SHD	NDR
Only Data	28.8	0.47	28.8	0.47	28.8	0.47	28.8	0.47
Only Knowledge	68.5	0.50	52.5	0.49	32.0	0.34	26.4	0.27
Our Method	33.9	0.39	32.8	0.44	21.1	0.32	21.0	0.34
Best Equal Conf.	25.3	0.41	22.6	0.36	24.9	0.41	24.7	0.39
Best Mult. Conf.	18.7	0.31	15.4	0.26	17.4	0.30	15.2	0.26

Table 3. The results obtained by the ‘Better’ population for the ‘Insurance’ network.

β	0.3		0.4		0.5		0.6	
	SHD	NDR	SHD	NDR	SHD	NDR	SHD	NDR
Only Data	28.8	0.47	28.8	0.47	28.8	0.47	28.8	0.47
Only Knowledge	47.7	0.38	23.9	0.23	12.3	0.15	7.7	0.09
Our Method	23.9	0.35	16.1	0.25	15.7	0.27	12.9	0.23
Best Equal Conf.	21.9	0.35	17.9	0.29	21.2	0.34	19.6	0.32
Best Mult. Conf.	15.3	0.26	13.2	0.23	11.7	0.21	10.2	0.18

Table 4. The results obtained by the 'Worse' population for the 'Alarm' network.

β	0.3		0.4		0.5		0.6	
	SHD	NDR	SHD	NDR	SHD	NDR	SHD	NDR
Only Data	27.5	0.47	27.5	0.47	27.5	0.47	27.5	0.47
Only Knowledge	86.7	0.54	62.5	0.46	55.6	0.43	71.2	0.40
Our Method	51.2	0.42	29.5	0.41	37.6	0.47	34.1	0.35
Best Equal Conf.	24.2	0.42	24.5	0.43	26.4	0.45	18.8	0.33
Best Mult. Conf.	18.1	0.30	11.4	0.19	11.5	0.20	9.8	0.17

Table 5. The results obtained by the 'Better' population for the 'Alarm' network.

β	0.3		0.4		0.5		0.6	
	SHD	NDR	SHD	NDR	SHD	NDR	SHD	NDR
Only Data	27.5	0.47	27.5	0.47	27.5	0.47	27.5	0.47
Only Knowledge	48.4	0.38	39.1	0.30	17.4	0.13	18.4	0.12
Our Method	25.1	0.33	19.6	0.27	13.4	0.20	10.9	0.18
Best Equal Conf.	15.0	0.26	14.9	0.27	13.4	0.23	15.4	0.28
Best Mult. Conf.	10.1	0.18	8.7	0.14	5.3	0.10	4.9	0.10

References

1. Lewis, F.I., Ward, M.P.: Improving epidemiologic data analyses through multivariate regression modelling. Emerging themes in epidemiology. 10, 4 (2013).
2. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: The combination of knowledge and statistical data. Machine learning. 20, 197–243 (1995).
3. Van der Gaag, L.C., Renooij, S., Witteman, C.L., Aleman, B.M., Taal, B.G.: How to elicit many probabilities. Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence. pp. 647–654. Morgan Kaufmann Publishers Inc. (1999).
4. Richardson, M., Domingos, P.: Learning with knowledge from multiple experts. ICML. pp. 624–631 (2003).
5. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT press (2009).
6. Chickering, D.M.: Optimal structure identification with greedy search. The Journal of Machine Learning Research. 3, 507–554 (2003).
7. Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max-min hill-climbing Bayesian network structure learning algorithm. Machine learning. 65, 31–78 (2006).
8. Chen, X.-W., Anantha, G., Lin, X.: Improving Bayesian network structure learning with mutual information-based node ordering in the K2 algorithm. Knowledge and Data Engineering, IEEE Transactions on. 20, 628–640 (2008).
9. Acid, S., de Campos, L.M., Fernández, M.: Score-based methods for learning Markov boundaries by searching in constrained spaces. Data Mining and Knowledge Discovery. 26, 174–212 (2013).