הכינוס ה-16 להנדסת תעשייה וניהול

16th Industrial Engineering And Management Conference

23-24 במרץ 2010 מלון דויד אינטרקונטיננטל, תל-אביב

# OPTIMIZATION BY TARGETED BAYESIAN NETWORK IN DECISION SUPPORT SYSTEMS

**A. Gruber** and I. Ben-Gal

Department of Industrial Engineering, Faculty of Engineering, Tel Aviv University,

Ramat-Aviv, Tel-Aviv 69978, Israel

E-mail: avivgrub@post.tau.ac.il / bengal@eng.tau.ac.il

## ABSTRACT

We present a Bayesian network learning method that can support optimization processes in industrial and service systems. The proposed method aims at learning the space of unknown systems from real data by a Bayesian network. While the underlying learning objectives of previous works were to best approximate the joint probability distribution of the learned domain, we aim at best approximating the conditional probability distribution of a predetermined target variable as a function of the rest of the domain variables. We prove that the proper Bayesian network for such a task is one for which the sum of mutual-information weights on the target variable and among its obtained parents is maximized. To address the trade-off between the network's complexity and its accuracy, we suggest information-gain criteria.

## 1. INTRODUCTION

In this work, we suggest a learning model that supports optimization of processes and systems, for which the exact underlying physical model is unknown. In particular, we propose a Bayesian network model that maximizes a predetermined target variable, as a function of the most influencing conrollable variables that describe the process.

Bayesian network (BN) is a probabilistic model representing the relations between variables in a certain domain with stochastic properties. Bayesian networks have been extensively employed in various applications in engineering and decision making. Essentially, a BN encodes the joint probability distribution $P(\mathbf{X})$ of the domain's random variables, and since BNs can be presented graphically they are fairly intuitive (Heckerman [1], Ben-Gal [2]).

A Bayesian network $\mathrm{B}(\mathrm{G}, \mathbf{\Theta})$ can often be used to represent the joint probability distribution of a vector of random variables $= (X_1, \dots, X_N)$ . The structure $\mathrm{G}(\mathbf{V}, \mathbf{E})$ is a directed acyclic graph (DAG) composed of $\mathbf{V}$, a vector of nodes representing the random vector $\mathbf{X}$, and $\mathbf{E}$, a set of directed edges connecting the nodes. An edge $E_{ji} = V_j \rightarrow V_i$ manifests dependence between the variables $X_j$ and $X_i$, while the absence of an edge demonstrates independence between the variables. A directed edge $E_{ji}$ connects a parent node $V_j$ to its child node $V_i$ (Heckerman [1], Yehezkel and Lerner [3]). We denote by $Z_i = \{X_i^1, \dots, X_i^{L_i}\}$ the set of the parent variables of the random variable $X_i$*'s*, represented by the set of parent nodes $D_i = \{V_i^1, \dots, V_i^{L_i}\}$ in G. The set of parameters $\mathbf{\Theta}$ holds local conditional probabilities over $\mathbf{X}$, $p(x_i|z_i)$ that quantify the edges for each node state $x_i$ (of the form $V_i = v_i$ where $v_i \in \{v_i^1, \dots, v_i^{s_i}\}$, the values that node $V_i$ can take) and each state $z_i$ (a conjunction of states $x_i^1 \cap \dots \cap x_i^{L_i}$) of $Z_i$.

A BN can be learned from observable data. Numerous BN learning methods have been suggested in recent years (e.g., Chickering [4], Heckerman [1], Heckerman et al. [5], Cheng et al. [6] and Pearl [7]). In particular, the K2 (Cooper and Herskovits [8]) and the PC (Spirtes et al. [9]) are two predominant BN learning algorithms.

We follow here is the *adding-arrows* (Williamson [10]). The *adding-arrows* is an algorithm which attempts to maximize the total information weight for the BN. Williamson [10] showed that a BN, satisfying some arbitrary constraint, that best approximates a joint probability distribution is one for which mutual information (MI) weight is maximized. He generalized the arguments presented earlier by Chow and Liu [11] regarding spanning trees. Chow and Liu proved that minimizing the Kullback–Leibler (KL) divergence between distributions is equivalent to maximizing the total MI weight of the tree. Yet, their underlying objective was to best approximate the joint probability distribution describing the domain, while they overlooked the fact that BNs are often used for system optimization, where some of the variables might influence the target variable more than others. Our objective, thus, is to best approximate the conditional probability distribution of the target variable, conditioned on the influencing variables within the domain, as similarly proposed by Ginsburg and Ben-Gal [12] in a different context of design-of-experiments.

The proposed approach, comparing to that of Williamson [10], suggests to reduce the BN's complexity while taking a predefined target variable into account, already at the BN learning stage. In this short paper we present the basic principles of the *Targeted Bayesian Network Learning* (*TBNL*) -- a BN learning method oriented for optimization purpose.

## 2. AN ILLUSTRATIVE AND MOTIVATED EXAMPLE

Consider the example in Table 1. Let us define the variable $X_3$ as our target variable. The *TBNL* algorithm draws an edge at a time - the one for which IG weight is the greatest.

Table 1: Illustrative Example of data[1]

| Case | $X_1$ | $X_2$ | $X_3$ | $X_4$ |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 2 |
| 2 | 1 | 1 | 2 | 2 |
| 3 | 1 | 1 | 2 | 2 |
| 4 | 1 | 2 | 2 | 3 |
| 5 | 1 | 2 | 2 | 3 |
| 6 | 1 | 2 | 2 | 3 |
| 7 | 2 | 2 | 1 | 1 |
| 8 | 2 | 2 | 1 | 1 |
| 9 | 2 | 2 | 1 | 1 |
| 10 | 2 | 2 | 2 | 1 |
| 11 | 2 | 2 | 2 | 1 |
| 12 | 2 | 2 | 2 | 1 |

Let us limit the number of parents to one ($K = 1$). Fig. 1 and Fig. 2 show the results obtained by the *adding-arrows* and by the *TBNL* algorithms respectively, both with $K = 1$, (a *CL* compatible with regard to the former). The total information weight ascribed to the BN obtained by the *adding-arrows* is 1.811 bits. This is comparing with 1.5 bits obtained by the *TBNL*. Although the weight of the network learned by the *TBNL* is considerably smaller, the *TBNL* fulfills our pursued objective, that is, the requirement for a target variable.

Notwithstanding, the outstanding consequence of the BN shown in Fig. 1 is that the target variable $X_3$ does not appear there at all. This is where the profound gap comes in: while the result shown in Fig. 1 would be satisfying for Williamson's and Chow's & Liu's objective, it bears an unfeasible result for our objective. Our objective in this case is to support the optimization of the target variable $X_3$ via the other variables which comprise the entire joint distribution. It will be shortly shown that such an objective may be accomplished by maximizing the information weight relative to the target variable and the total information weight among its parents, rather than just maximizing the total information weight of the BN.

## 3. THE TARGETED BAYESIAN NETWORK LEARNING (*TBNL*) METHOD

Our underlying assumption is that a target variable $X_i \in \mathbf{X}$ is given and that we aim at best approximating its probability distribution $p(X_i)$ as a function of the entire domain. Namely, we wish to represent $p(X_i) = \sum_{x_i^c \in X_i^c} p(X_i|x_i^c)\, p(x_i^c)$ by $q(X_i) = \sum_{x_i^c \in X_i^c} p(X_i|z_i) p(x_i^c)$, where $x^c$

[1] This example has been taken from Shkolnik [13].

denotes the atomic states of $X^c = \mathbf{X} \backslash X_i$ and $z_i$ denotes the parents of $X_{i,},$. $Z_i \subseteq X^c \in \mathbf{X}$. For any subset of variables $\mathbf{X}^{'} \subseteq \mathbf{X}$ we shall denote with the respective literals $\mathbf{X}^{'c} \in \mathbf{X}$ the complementary subset of nodes in $\mathbf{X}$ that are not in $\mathbf{X}^{'}$, such that $\mathbf{X}' \cap \mathbf{X}'^c = \emptyset$ and $\mathbf{X}' \cup \mathbf{X}'^c = \mathbf{X}$.

By the law of total probability we know that $p(x_i) = \sum_{X_{\boldsymbol{i}}^{c}} p(x_1, \dots, x_n)$. Each element within the summation is a component of $p(\mathbf{X}) = p(X_i|X_{\boldsymbol{i}}^{\boldsymbol{c}})p(X_{\boldsymbol{i}}^{\boldsymbol{c}})$. If one wishes to approximate $p(X_i|X_{\boldsymbol{i}}^{\boldsymbol{c}})$ by $q_1 = p(X_i|Z_i)$, then the approximation for $p(X_i)$ becomes $q(X_i) = \sum_{x_{\boldsymbol{i}}^{c} \in X_{\boldsymbol{i}}^{c}} p(X_i|z_i)p(x_{\boldsymbol{i}}^{\boldsymbol{c}})$ where $Z_i \subseteq X_{\boldsymbol{i}}^{\boldsymbol{c}}$ and therefore it is also a probability distribution after normalization, namely,

$$p(Z_i) = P_r\{Z_i = (z_{i_1}, \dots, z_{i_k})\} = \sum_{Z_i^c} p(x_1, \dots, x_n) = p(\mathbf{X})/p(Z_i^c|Z_i)$$

For simplicity of literals annotation, we shall denote $Z_i^c \cap X_{\boldsymbol{i}}^{\boldsymbol{c}}$ with $\hat{Z}_i$ - the variables in $\mathbf{X}$ for which representing nodes are neither the parents of $X_i$ nor $X_i$ itself. By that definition $X_{\boldsymbol{i}}^{\boldsymbol{c}} = Z_i \cup \hat{Z}_i$ and hence $d\big(p(X_i|X_{\boldsymbol{i}}^{\boldsymbol{c}})||p(X_i|Z_i)\big) = \sum_{x_1,\dots,x_n \in \mathbf{X}} p(x_i|x_{\boldsymbol{i}}^{\boldsymbol{c}}) log[p(x_i|x_{\boldsymbol{i}}^{\boldsymbol{c}})/p(x_i|z_i)] = \sum_{x_1,\dots,x_n \in \mathbf{X}} p(x_i|x_{\boldsymbol{i}}^{\boldsymbol{c}}) log\big[p(x_i, \hat{z}_i|z_i)/\big(p(x_i|z_i)p(\hat{z}_i|z_i)\big)\big] \equiv I\big(X_i; \hat{Z}_i|Z_i\big)$. Accordingly, we obtain

$$d\big(p(X_i|X_{\boldsymbol{i}}^{\boldsymbol{c}})||p(X_i|Z_i)\big) = I\big(X_i; \hat{Z}_i|Z_i\big) \qquad (1)$$

Eq. (1) suggests that minimizing the KL distance between $p(X_i|X_{\boldsymbol{i}}^{\boldsymbol{c}})$ and its estimator $q_1 = p(X_i|Z_i)$ is equivalent to minimizing the IG between $X_i$ and $\hat{Z}_i$ conditional on $Z_i$. This is similar to maximizing $I(X_i; Z_i)$ over all possible sets of $Z_i \in \mathbf{X}$.

As to $p(X_{\boldsymbol{i}}^{\boldsymbol{c}})$, by virtue of Williamson's proof, we know that

$$d\big(p(X_i^c)||q(X_i^c)\big) = -H(p) - \sum_{X_j \in X_i^c} I(X_j; D_j) + \sum_{X_j \in X_i^c} H(p|X_j) \qquad (2)$$

Eq. (2) governs the theorem that a BN that best approximates $p(X_{\boldsymbol{i}}^{\boldsymbol{c}})$ is one for which MI weight is maximized, i.e., $Max \sum_{X_j \in X_{\boldsymbol{i}}^{\boldsymbol{c}}} I(X_j; D_j)$. Now, if one represents $p(X_i)$ by $q(X_i) = \sum_{z_i \in Z_i} p(X_i|z_i)p(z_i)$ exclusively by a BN, then he gets that

$$d\big(p(Z_i)||q(Z_i)\big) = \sum_{z_i \in Z_i} p(z_i) log \frac{p(z_i)}{\prod_{X_j \in Z_i} p(X_j|z_j)} = -H(Z_i) - \sum_{X_j \in Z_i} I(X_j; Z_j) + \sum_{X_j \in Z_i} H(p|X_j) \qquad (3)$$

Minimizing the term $-\sum_{X_j \in Z_i} I(X_j; Z_j)$ is equivalent to Williamson's result, only within $Z_i \in X_{\boldsymbol{i}}^{\boldsymbol{c}}$. Finally, as a result of eqs. (1) and (3) we obtain that

$$Min\left(d\big(p(X_i)||q(X_i)\big)\right) \rightarrow Z_i = argmax_{Z' \in X_{\boldsymbol{i}}^{\boldsymbol{c}}}\big(I(X_i; Z')\big) \,\&\, Max\left(\sum_{j: X_j \in Z_i} I(X_j; Z_j)\right)$$

This process holds independently for any desired $X_i \in \mathbf{X}$, and therefore can be recursively applied for each variable $X_j \in Z_i$in and so forth. As a result, the obtained BN can be built such that each variable is best predicted as well as explained by other variables in the domain, where the entire network is oriented towards the target variable.

The proposed method, the ***Targeted Bayesian Network Learning*** (*TBNL*) employs a recursive procedure that can be applied on any given current variable and any set of potential parents. We define minimum percentage relative information gain (*PRIG*) as a quantitative parameter for setting up an information-based constraint that serves as a stopping condition. For any variable $X \in \mathbf{X}$, with parent variables set $Z \in \mathbf{X}$, and a potential parent $Z' \in \mathbf{X}$, the stopping condition of the *TBNL* by this parameter is when $I(X; Z'|Z)/H(X) \times 100 \leq PRIG$. The range of the *PRIG* is [0-100], where zero implies that the procedure will add edges from each potential parent, except for those contributing a zero weight, whereas 100 implies that the current node will not have parents.

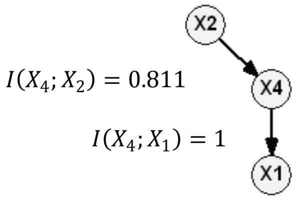

Fig. 1: A Bayesian network resulted from the *adding-arrows* algorithm by example 1 with at most one parent allowed for each node.

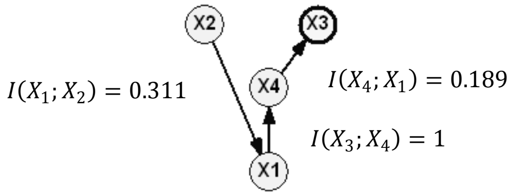

Fig. 2: Bayesian network resulted from the *TBNL* algorithm by example 1 with at most one parent allowed for each node.

Gruber and Ben-Gal [14] suggest additional constraints criteria through information measures in a detailed and full disscusion regarding complexity considerations, including a practical example.

## 4. CONCLUSIONS

We show that the best approximation to the marginal probability distribution of a given target variable as a function of the entire domain is the one that i) maximizes the MI weight between the target variable and its potential parents; and ii) given those parents, it maximizes the total MI weight within the rest of the domain variables. Further to that, we saw that maximizing the total MI weight solely within those parents suffices that goal and can reduce the computation cost dramatically.

We claim that the proposed *TBNL* algorithm handles well the trade-off between information gain and complexity when learning a BN from data. Having drawn a decision line of complexity as a function of the *PRIG*, one may be able to better control the model's accuracy and computation cost.

## 5. REFERENCES

1. Heckerman, D. A tutorial on learning with Bayesian networks, MS TR-95-06, 1995.
2. Ben-Gal I. Bayesian Networks. In: Ruggeri F., Faltin F. & Kenett R. (Eds.), Encyclopedia of Statistics in Quality and Reliability, John Wiley & Sons. 2007.
3. Yehezkel R. and B. Lerner. Bayesian Network Structure Learning by Recursive Autonomy Identification. The 10th International Workshop on Artificial Intelligence and Statistics, pages 429–436, AISTATS, 2005.
4. Chickering D.M. Optimal structure identification with greedy search. Journal of Machine Learning Research, 3, 507–554, 2002.
5. Heckerman, D., D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning 20, 197–243, 1995.
6. Cheng, J. and R. Greiner. Learning Bayesian Belief Network Classifiers: Algorithms and System. Lecture Notes in Computer Science, Pages 141-151, Springer, 2001.
7. Pearl J. Causality: Models, Reasoning, and Inference. University Press: Cambridge, 2000.
8. Cooper G. F. and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. Machine Learning, 9, 309-347, 1992.
9. Spirtes P., C. Glymour, and R. Scheines. Causation, Prediction and Search. 2nd edition, MIT Press, 2000.
10. Williamson J. Approximating discrete probability distributions with Bayesian networks, in Proc. of the International Conference on Artificial Intelligence in Science and Technology, 16-20 December :Hobart Tasmania, 2000.
11. Chow C.K. and C.N. Liu. Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory IT-14, pages 462-467, 1968.
12. Ginsburg H. and I. Ben-Gal. Designing Experiments for Robust Optimization Problems: The Vs-optimality criterion. IIE Transactions on Quality and Reliability, 38, 445 – 461, 2006.
13. Shkolnik N. Constructing of Classification Trees by the Dual Information Distance Method. Master of Science thesis, Department of Industrial Engineering Tel -Aviv University, 2008.
14. Gruber A. and I. Ben-Gal, Targeted Bayesian Network Learning, In Proc. ICML, Haifa, June 2010.