A New Polynomial Time Algorithm for Bayesian Network Structure Learning

Sanghack Lee[1], Jihoon Yang[2,⋆,⋆⋆], and Sungyong Park[2]

[1] Diquest Inc.
Sindo B/D, 1604-22, Seocho-dong, Seocho-gu
Seoul 137-070, Korea
shlee@diquest.com
[2] Department of Computer Science, Sogang University
1 Shinsoo-dong, Mapo-gu, Seoul 121-742, Korea
{yangjh, parksy}@sogang.ac.kr

Abstract. We propose a new algorithm called *SCD* for learning the structure of a Bayesian network. The algorithm is a kind of constraint-based algorithm. By taking advantage of variable ordering, it only requires polynomial time conditional independence tests and learns the exact structure theoretically. A variant which adopts the Bayesian Dirichlet scoring function is also presented for practical purposes. The performance of the algorithms are analyzed in several aspects and compared with other existing algorithms. In addition, we define a new evaluation metric named *EP power* which measures the proportion of errors caused by previously made mistakes in the learning sequence, and use the metric for verifying the robustness of the proposed algorithms.

1 Introduction

Pattern classification is a crucial and growing field with applications in many domains. Generally, classification is the task for discovering an unknown class of an instance with observed features of the instance which may or may not be closely related to each other in a given domain. Discovering the relationships among the features and making use of them in classification can shed light on the understanding of the domain. A Bayesian network has the capability of finding the relationships among the features. It decomposes variables with independence relations. Therefore, it has become necessary to construct Bayesian networks from the domain of interest in order to understand the relationships among the features for pattern classification.

⋆ This work was supported by grant No. R01-2004-000-10689-0 from the Basic Research Program of the Korea Science & Engineering Foundation and by the Brain Korea 21 Project in 2006.
⋆⋆ Corresponding author.

X. Li, O.R. Zaiane, and Z. Li (Eds.): ADMA 2006, LNAI 4093, pp. 501–508, 2006.

2 Bayesian Networks

A Bayesian network [1] is a probabilistic graphical model that encodes variables and their dependence relationships into nodes and edges, respectively. It is a formalism for representing and reasoning with models of problems involving uncertainty. For this reason, Bayesian networks are used in many domains where chasing causes and inferring effects are important (e.g. medical diagnosis [2]).

Bayesian networks can be acquired from data or domain experts' knowledge. For complex domains, a number of learning algorithms that generate Bayesian networks given data were developed. The algorithms are divided into two approaches - score-based [3] and constraint-based [4,5,6]. Simply speaking, the former is practical and the latter is theoretical in their use.

In this paper, we use the same symbols those appeared in [7] for consistency. Additionally, $X \prec Y$ means that a variable X precedes Y in a variable ordering $\mathcal{O}$. $S_{\prec X}$ is a structure of variables preceding X. $\mathcal{A}$ and $\mathcal{V}$ are arcs and vertices, respectively.

3 Sequential Causal Discovery

In this section, a new Bayesian network structure learning algorithm called *SCD* (Sequential Causal Discovery) is described with two theorems.

3.1 SCD Algorithm

The algorithm sequentially determines parents of a node. The sequence is the variable ordering (from the top to the bottom nodes of a given Bayesian network). Let's assume that we have n nodes. The first node trivially has no parents. The second node may or may not hold the first node as a parent. In the same manner, the last node determines its parent nodes from all preceding nodes. The algorithm decides the existence of the edge between two nodes with single conditional independence (CI) test. The test is executed by conditioning Markov blanket of the preceding node between two nodes. The Markov blanket is derived from the partially constructed Bayesian network on the learning process and may not be the same one from the original Bayesian network.

Now we review two important concepts - *d-separation* and a *Markov blanket* [8] - that will be used in the description and the proof for the correctness of the algorithm. D-separation [8,9] is a criterion for deciding the independency between two sets of variables given a conditioning set in a causal graph. A Markov blanket $MB(X)$ of a variable $X \in \mathcal{V}$ is any subset $\mathbf{Z}$ of variables for which satisfies $I(X, \mathbf{Z}, \mathcal{V} - \mathbf{Z} - X)$[1] and $X \notin \mathbf{Z}$. Generally, a Markov blanket of a

[1] The independency between A and C given B is expressed as $I(A, B, C)$, and conditional mutual information $I(A; C|B)$ is another expression for the independency. $I(A; C|B) > 0$ describes the dependency in this paper.
In the experiments, mutual information is acquired from $I(X; Y|\mathbf{Z}) = \hat{I}(X; Y|\mathbf{Z}) + \frac{r^*_{XZ} + r^*_{YZ} - r^*_{Z} - r^*_{XYZ}}{2N} + z_N\sigma$. For detailed explanations, see [10]. Like other independence tests, we apply a parameter $\alpha = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{z_N}{\sqrt{2}}\right)\right]$. $\alpha = 0.5$ matches $z_N = 0$.

variable X in directed graphs are defined by the following equation, $MB(X) = (\Pi_X \cup \mathbf{Y} \cup \Pi_{\mathbf{Y}}) \backslash \{X\}$ where $\mathbf{Y} = \{Y | X \in \Pi_Y\}$. Based on this background, we have the following theorem.

Theorem 1. *Given two variables X and Y where $X \prec Y$, $I(X; Y | MB(X)_{S_{\prec Y}})_S$ holds if and only if $X \notin \Pi_Y$.*

Proof. As known, $\langle X | \mathbf{Z} | Y \rangle_S \Leftrightarrow I(X; Y | \mathbf{Z})$. So we can prove conditional independency with d-separation $\langle X | MB(X)_{S_{\prec Y}} | Y \rangle_S$. Obviously, if $X \in \Pi_Y$ holds, two nodes are dependent.

When all the paths between X and Y are not active paths, d-separation is satisfied. All undirectional paths can be divided into the following four cases[2]. We assume that $\mathcal{A}_{XY} \notin \mathcal{A}$ because directly connected variables are trivially d-connected and can not be d-separated by conditioning on any subset.

1. $X \rightarrow Z$ where $Z \in \mathcal{V}_{\prec Y}$: We have to consider the paths in two ways - $X \rightarrow Z \rightarrow$ or $X \rightarrow Z \leftarrow$ - from the viewpoint of d-separation. If the path continues as $Z \rightarrow P$ then the variables are d-separated by Z as an intermediate cause. In another situation $Z \leftarrow Q$, Q is a parent of the child Z and is in the Markov blanket. So, Q becomes intermediate or common cause of Z and another neighbor node of Q. And Q d-separates X and Y, and it can not be Y since $Q \prec Z \prec Y$.
2. $X \rightarrow Z$ where $Z \notin \mathcal{V}_{\prec Y}$: Z will not be an element of $MB(X)_{S_{\prec Y}}$. The path from X to Y through Z should have at least one node as a common effect. But the common effect is not in $\mathcal{V}_{\prec Y}$. Surely, the node is not in conditioning set and the path is not active.
3. $X \leftarrow Z$ where $Z \in \mathcal{V}_{\prec Y}$: Z is one of $MB(X)_{S_{\prec Y}}$ and it blocks the route from X to Y, since Z is a common or intermediate cause of a variable next to Z and X.
4. $X \leftarrow Z$ where $Z \notin \mathcal{V}_{\prec Y}$: Z can not be $Z \notin \mathcal{V}_{\prec Y}$ since $X \prec Y$.

Above cases cover all the paths from X to Y. In all cases, X and Y are d-separated by $MB(X)_{S_{\prec Y}}$ if $X \notin \Pi_Y$. Trivially, X and Y are d-connected if $X \in \Pi_Y$. Thus $I(X; Y | MB(X)_{S_{\prec Y}})$ if and only if $X \notin \Pi_Y$. □

Several constraint-based algorithms find the Markov blanket of the nodes (which d-separates the nodes and others) and orient the directions of the edges. If we have an assumption that a variable ordering is given, then the algorithms find parents of a node (those d-separate the node and its ancestors except the parents) rather than its Markov blanket. The procedure for deciding an exact set of parents of a node is to find a subset of ancestors of the node which d-separates other ancestors from the node, which takes exponential time. To find the parents of a node, our algorithm has a *one-to-one* scheme which is not the same as the *all-to-one* approach. More precisely, *SCD* determines one parent of a variable at a time not a set of parents simultaneously. This difference makes our algorithm time efficient. Algorithm 1 is the *SCD* algorithm, whose correctness is proved in Theorem 2.

[2] $X \rightarrow Z$ and $X \leftarrow Z$ combined with $Z \prec Y$ and $Y \prec Z$.

Algorithm 1. *SCD*

Input: Variable ordering $\mathcal{O}$, Data D
Output: Bayesian network structure S
$\mathcal{A} \Leftarrow \phi$
$S \Leftarrow (\{\mathcal{O}_1\}, \mathcal{A})$
for $i = 2$ to n **do**
  $Y \Leftarrow \mathcal{O}_i$
  for all j such that $1 \leq j < i$ **do**
    $X \Leftarrow \mathcal{O}_j$
    if $I(X; Y | MB(X)_S) > 0$ **then**
      Add $\mathcal{A}_{XY}$ to $\mathcal{A}$
    end if
  end for
  $S \Leftarrow (\mathcal{V}_{\preceq Y}, \mathcal{A})$
end for

Theorem 2. *The SCD algorithm always constructs a correct structure of the Bayesian network.*

Proof. We use the mathematical induction to prove the correctness of the algorithm.

1. $S^h_{\prec \mathcal{O}_2}$ is a true structure. A Bayesian network is a directed acyclic graph. This fact implies that the $\mathcal{O}_1$ (the only element in $\mathcal{V}_{\prec \mathcal{O}_2}$) itself has no arcs. Thus, an initial structure $S^h_{\prec \mathcal{O}_2} = S_{\prec \mathcal{O}_2}$.
2. Assume that $S^h_{\prec \mathcal{O}_k}$ is a true structure for some k. By the Theorem 1, X and $\mathcal{O}_k$ is d-separated by the Markov blanket of X for all $X \prec O_k$ if and only if $\mathcal{A}_{X\mathcal{O}_k} \notin \mathcal{A}$. Therefore, $S^h_{\prec \mathcal{O}_{k+1}}$ is also true for some $1 < k \leq n$.

By the steps of mathematical induction, we proved that the algorithm *SCD* always constructs the true structure given the variable ordering. □

We have introduced two theorems. Theorem 1 showed that the Markov blanket of a node X, which precedes Y, made under subgraph $S_{\prec Y}$ showed precise independence relationships. Theorem 2 showed the correctness of *SCD* algorithm.

3.2 Variant Algorithm

Here, we propose a variant of *SCD* algorithm. The variation is a hybrid algorithm with the *K2 metric*. *SCD* algorithm uses mutual information for determining independency. Like Chi-square and Fisher's z test, CI tests adopt an arbitrary level of decision boundary such as $\alpha = 0.05$. We can set a new decision boundary (usually used term - threshold) for determining independent relation using soft (i.e. relative) value rather than hard (i.e. absolute) value 0. The K2 algorithm [3] stops adding new parents to a variable when the addition can not improve K2 score further. Originally, our constraint-based algorithm's

Algorithm 2. *SCD-K2 boundary*

Input: Variable ordering $\mathcal{O}$, Data D
Output: Bayesian network structure S
$\mathcal{A} \Leftarrow \phi$
$S \Leftarrow (\{\mathcal{O}_1\}, \mathcal{A})$
for $i = 2$ to n **do**
  $Y \Leftarrow \mathcal{O}_i$
  $MI \Leftarrow \phi$
  for all j such that $1 \leq j < i$ **do**
    $X \Leftarrow \mathcal{O}_j$
    $MI_j \Leftarrow I(X; Y | MB(X)_S)$
  end for
  $\mathbf{I} \Leftarrow$ Sort MI in decreasing order / Return indices
  $k \Leftarrow \min \{\arg\max_k \mathrm{K2}(Y, \mathcal{O}_{\mathbf{I}_{1:k}})\}$
  Add $\mathcal{A}_{O_{\mathbf{I}_{1:k}} Y}$ to $\mathcal{A}$
  $S \Leftarrow (\mathcal{V}_{\preceq Y}, \mathcal{A})$
end for

arc determining mechanism did not take advantage of the conditional independence between the parents and other ancestors of a variable. Adopting K2 will partly make up for the foible in score-based approaches. Algorithm 2 depicts this *SCD* variant.

4 Experimental Results

Experiments have been done with various Bayesian networks. We explain experimental settings and a new metric. The performance of proposed algorithms is compared and analyzed with other algorithms.

4.1 Data

Data for evaluating structure learning algorithms may be acquired from real world domains or sampled from artificial models. Usually, the evaluation for score-based learning algorithms can be achieved with both kinds of data sets. However, the data sets for constraint-based algorithms should satisfy the assumptions previously introduced. By this reason, we evaluate the algorithm with the data sampled from some Bayesian networks.

We've generated data from random Bayesian networks[3] varying size and complexity[4] using BNGenerator [11]. The probabilities for conditional probability tables are generated under between 'deterministic' and 'uniform' distribution.

[3] 20 networks have generated with the same size and complexity (induced width) to get accurate results.

[4] The complexity of the ALARM network in the below looks similar with random Bayesian network with induced width 2. Bayesian networks with induced width as 3 and 5 is complex enough.

Table 1. Results of SCD, Variant, K2, and Hill Climbing

Size of networks		7		10		30		ALARM
Induced width		3	5	3	5	3	5	
SCD[a]	E[b]	1.22	1.17	4.97	6.22	93.25	344.8	61.03
	P	98 ± 4	98 ± 4	97 ± 6	95 ± 7	84 ± 11	66 ± 15	96
	R	92 ± 11	86 ± 11	91 ± 10	86 ± 14	86 ± 7	88 ± 9	96
	F	94 ± 7	91 ± 7	94 ± 7	90 ± 10	85 ± 8	74 ± 9	96
SCD-K2 boundary	E	1.41	1.4	5.49	7.04	103.77	206.53	53.73
	P	99 ± 3	100 ± 0	100 ± 1	99 ± 3	92 ± 8	92 ± 5	98
	R	97 ± 8	96 ± 9	94 ± 8	90 ± 12	86 ± 6	83 ± 5	98
	F	**98 ± 5**	**98 ± 5**	**97 ± 5**	94 ± 8	89 ± 6	87 ± 4	**98**
K2	E	0.38	0.4	1.69	1.86	30.5	38.49	33.94
	P	98 ± 4	99 ± 3	99 ± 2	99 ± 3	98 ± 3	97 ± 2	82
	R	99 ± 6	97 ± 7	96 ± 7	98 ± 4	98 ± 2	98 ± 3	96
	F	**98 ± 4**	97 ± 4	**97 ± 4**	**98 ± 2**	**98 ± 2**	**97 ± 2**	88
Hill-climbing	E	40.47	43.92	230.04	262.02	-	-	-
	P	46 ± 23	57 ± 22	57 ± 14	54 ± 13	-	-	-
	R	51 ± 23	57 ± 20	62 ± 15	60 ± 13	-	-	-
	F	48 ± 23	57 ± 21	59 ± 14	57 ± 12	-	-	-

[a] α of *SCD* is 0.48 and of *SCD-K2 boundary* is 0.5.

[b] E, P, R, and F stands for elapsed time in seconds, precision, recall, and f-measure.

In addition, algorithms were evaluated with the ALARM network [12] which is a paragon of Bayesian networks and most widely used for structure learning. With models, 10000 samples are generated.

4.2 Experimental Results

Evaluation of algorithms is carried out by the accuracy of determining existence of edges in the Bayesian network. Thus, precision and recall are used that are related to false negatives and false positives. In addition, we introduce a new metric called *EP power* which is appropriate for analyzing the robustness of an algorithm.

Error Propagation Analysis. We analyze the characteristic of an algorithm called error propagation which is useful for measuring the degree of its *fragility*.

Definition 1 (Error Propagation). *A false positive which caused by one or more false negatives is called an error propagation, if*

1. *A false positive between variables X and Z.*
2. *The set of false negatives between the variable X and the set of variables $\mathbf{Y}$.*
3. *$X \notin \Pi_Z$, $Y \in \Pi_Z$, $X - Y$, and $X, \mathbf{Y} \prec Z$*

Simply speaking, all $Y \in \mathbf{Y}$ satisfy $X - Y \rightarrow Z$. This relation can be divided into two different relations: Y is a intermediate cause or same cause of X and Z. Thus, missing any one of $\mathbf{Y}$ in the hypothesis (one or more false negatives) causes an extra arc $X \rightarrow Z$ (a false positive).

Table 2. The *EP power* analysis of *SCD* and its variants

	α	$p(FP)$	$p(FP_{XZ})$	$p(EP_{XZ})$	*EP power*	$p(FP\|FP \cup FN)$
SCD	0.48	0.0706	0.1001	0.1762	0.0693	0.6918
	0.45	0.0242	0.0390	0.1051	0.1609	0.2854
SCD-K2 boundary	0.50	0.0135	0.0243	0.0819	0.0937	0.2922
	0.48	0.0114	0.0207	0.0761	0.1131	0.2411
	0.45	0.0144	0.0251	0.0771	0.1144	0.2268

Definition 2 (Error Propagation Power). *The increased or decreased portion of errors purely caused by error propagation over total errors.*

$$\frac{\left(p(EP_{XZ}) - p(FP_{XZ}|\mathcal{A}_{X\mathbf{Y}} \backslash \mathcal{A}^h_{X\mathbf{Y}} = \phi)\right)|XZ|}{|FP| + |FN|} \tag{1}$$

$p(EP_{XZ})$ can be rewritten as $p(FP_{XZ}|\mathcal{A}_{X\mathbf{Y}} \backslash \mathcal{A}^h_{X\mathbf{Y}} \neq \phi)$.

In Table 1, the original algorithm have shown the curse of false positives. Even though the probability of the false positive is low (about 5%), large number of false positives causes poor precision. By mixing *SCD* with the K2 metric, it properly filtered many false positives even though the K2 metric was just for determining threshold. In addition with a reasonable Bayesian network model ALARM network, our algorithms perform more accurate than the K2 algorithm (For more results on ALARM network, see [6]), because the results of the CI tests are more reliable and the algorithms do not fall into local maxima.

In *EP power* analysis in Table 2, we have shown that the error cascading effect is smaller (about 10%) than general false positives and negatives. Because the *EP power* is a part of false positives, the quality of CI tests should be improved to reduce such errors.

5 Conclusion and Future Work

Novel approaches to the structure learning of Bayesian network (*SCD* and variant) have been proposed and those were compared with existing algorithms through carefully designed experiments, yielding improved performance. The results of our original algorithm *SCD* and the results from experiments of the variation were variously analyzed in the previous section.

In conclusions, *SCD* algorithm itself is theoretically proved algorithm but is under the control of the quality of CI tests. To lessen the effect, we hybridize the algorithm with the K2 metric and obtained promising results. In this context, we suggest the following as future works:

- **Variable ordering:** In general, a variable ordering for a structure is not given. Hence, we have to find such ordering before running the algorithm. The ordering can be derived from a wrapper approach [13] for score-based algorithms. Transforming such methods for constraint-based algorithms should be achieved.

– **Heuristics and post process:** Constraint-based algorithms look compact and solid. But an imperfect probability distribution from limited finite data makes the algorithms fallacious. For the robust CI tests, more heuristics and rules [14] might be added to the algorithm.

References

1. J. Pearl, "Fusion, propagation and structuring in belief networks," in *Uncertainty in Artificial Intelligence* (Kanal and Lemmer, eds.), pp. 357–370, North-Holland, 1986.
2. P. Spirtes, C. Glymour, R. Scheines, S. Kauffman, V. Aimale, and F. Wimberly, "Constructing bayesian network models of gene expression networks from microarray data," in *Proceedings of the Atlantic Symposium on Computational Biology, Genome Information Systems and Technology*, 2001.
3. G. F. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, pp. 309–347, 1992.
4. P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search.* Springer-Verlag, 1993.
5. J. Pearl and T. S. Verma, "A theory of inferred causation," in *KR'91: Principles of Knowledge Representation and Reasoning* (J. F. Allen, R. Fikes, and E. Sandewall, eds.), (San Mateo, California), pp. 441–452, Morgan Kaufmann, 1991.
6. J. Cheng, R. Greiner, J. Kelly, D. A. Bell, and W. Liu, "Learning bayesian networks from data: An information-theory based approach.," *Artif. Intell.*, vol. 137, no. 1-2, pp. 43–90, 2002.
7. D. Heckerman, "A tutorial on learning bayesian networks," Tech. Rep. MSR-95-06, Microsoft Research, 1995.
8. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, Inc., San Mateo, California, 1988.
9. J. Pearl, *Causality: Modeling, Reasoning, and Inference.* Cambridge: Cambridge University Press, 2000.
10. M. S. Roulston, "Estimating the errors on measured entropy and mutual information," *Physica D: Nonlinear Phenomena*, vol. 125, pp. 285–294, 1 1999.
11. J. S. Ide, F. G. Cozman, and F. T. Ramos, "Generating random bayesian networks with constraints on induced width.," in *ECAI* (R. L. de Mántaras and L. Saitta, eds.), pp. 323–327, IOS Press, 2004.
12. I. Beinlich, H. Suermondt, R. Chavez, and G. Cooper, "The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks," in *Proceedings of the Second European Conference on Artificial Intelligence in Medicine*, (London), pp. 247–256, 1989.
13. W. H. Hsu, "Genetic wrappers for feature selection in decision tree induction and variable ordering in bayesian network structure learning.," *Inf. Sci.*, vol. 163, no. 1-3, pp. 103–122, 2004.
14. H. Steck, *Constraint-Based Structural Learning in Bayesian Networks using Finite Data Sets.* PhD thesis, Munich University of Technology, 2001.