TPDA^2 ALGORITHM FOR LEARNING BN STRUCTURE FROM MISSING VALUE AND OUTLIERS IN DATA MINING

Benhard Sitohang, G.A. Putri Saptawati
Software Engineering & Data Research Group
School of electrical Engineering & Informatics, ITB
Email: benhard@informatika.org

ABSTRACT: Three-Phase Dependency Analysis (TPDA) algorithm was proved as most efficient algorithm (which requires at most O(N^4) Conditional Independence (CI) tests). By integrating TPDA with "node topological sort algorithm", it can be used to learn Bayesian Network (BN) structure from missing value (named as TPDA^1 algorithm). And then, outlier can be reduced by applying an "outlier detection & removal algorithm" as pre-processing for TPDA^1. TPDA^2 algorithm proposed consists of those ideas, outlier detection & removal, TPDA, and node topological sort node.

Keywords: missing value, noisy data, BN structure, TPDA.

INTRODUCTION

Currently, most of algorithms for structural learning of BN are developed based on assumption that data is complete. Although there have been many methods for handling missing value and noise, yet there has not been any method to improve capability of such algorithms in term of incompleteness and outlier.

In recent years, graphical probabilistic models, including Bayesian Networks, have become very popular. BNs are considered as classifier after the discovery that Naïve-Bayes are effective. Since then, learning Bayesian networks has become a very active research topic and many algorithms have been developed for it. As noted above, scoring-based and CI or constraint-based are two general approaches to structural learning of BN. Heckerman in [9] compare these two approach and conclude that, *in terms of modeling a distribution*, scoring-based methods BN are superior than constraint-based methods. Nonetheless, Friedman in [7] show theoretically that scoring-based methods may result in poor classifiers and less efficient in practice. On the other hand, constraint-based methods are usually much more efficient when the number of variables are large [2,3]. [3] summarizes the representative scoring-based and constraint-based algorithms. Another approaches called model averaging propose that instead of searching for a single best solution, the algorithms result in several networks and use the 'average' of these networks to perform inference [3,15]. Some example include in [3, 13, 21, 23]. Scoring-based and constraint-based algorithms have their own advantages and disadvantages. As a result, there are some effort to combine these two method in order to maximize the advantages. Some algorithms are [21,22] and [23]. Generally, they start learning process with constraint-based then scoring based.

As consideration, this section presents some of welknown algorithms for learning BN structure using different approaches. The algorithms are K2 (for scoring-based method), PC (for constraint-based method), CB (for hybrid method), BC and BSEM (learning from incomplete data).

K2 Algorithm

The algorithm was developed by [4]. It greedily searches for DAG that approximates maximizing score. The search space is a set of all DAGs containing the n variables. The algorithm proceeds as follows [4,19], and the complete algorithm can be found in [4]:

> Assume an ordering node. Let *Pred* (*Xi*) be the set of nodes that proceed Xi in the ordering. Set initial parents PAi of Xi to empty and compute scoreB. Next, examine the nodes in sequence according to the ordering. When examining Xi, determine the nodes in Pred (Xi) which most increases the score. Add greedily this node to PAi. Continue doing this until the addition of no node increases the score.

PC Algorithm

PC algorithm developed [23] to learn BN structure based on conditional independencies using CI tests [13]. As one of the famous constraint-based algorithms, the algorithm searches a DAG pattern by which distribution P admits faithfulness condition to the DAG representation. Moreover, given the set of

conditional independencies in a probability distribution, it assumes the following conditions:

- Faithfulness of distribution P
- Sufficiency of V, i.e., there is no missing value or hidden variable
- CI test

In general, PC algorithm can be divided into three phase as follows, and the complete algorithm can be found in [21]:

- Identification of undirected graph: find an initial graph in which every pair of nodes having dependencies will be connected by an edge. Whilst, every pair of nodes having independencies will not be connected.
- Identification of uncoupled head-to-head meeting: based on the result of previous phase, it search all uncoupled meeting *(X–Z–Y)* which is potentially being uncoupled head-to-head meeting ($X \rightarrow Z \leftarrow Y$). Then *edge* on that uncoupled *meeting* is oriented becomes *uncoupled head-to-head meeting*.
- Orienting remaining edges: orient remaining edges resulting from the previous phase. The orientation is such that the resulting graph having no cycle.

CB Algorithm

As stated above, scoring-based and constraint-based methods have their own advantages and disadvantages. Some algorithms were developed by combining those two methods to minimize the disadvantages while maximizing the advantages. CB algorithm developed by [22] is one example of such algorithms [21,22]. It does not require prior knowledge since it generates node ordering resulting from constraint-based algorithm. Then, it continues the process by employing scoring-based algorithm. The complete algorithm is in [22] assuming the following conditions:

- Variables are discrete
- Cases occur independently
- No missing value (data is complete)
- No information about prior probability regarding the structure

Two phases constitute CB algorithm as follows:

- 1st phase: apply modified PC algorithm to generate undirected graph and orient edges. Then topological sort is apply to the resulting graph to obtain node ordering
- 2nd phase: given node ordering from 1st phase, apply K2 algorithm to learn BN structure

Bound & Collapse (BC) Algorithm

Bound and Collapse (BC) method is proposed by [18] to overcome problems of Missing Information Principles [18,20,21]. BC method is a deterministic method estimating conditional probabilities which define the dependencies in BN. It is not rely on the Missing Information Principles. It starts by bounding the set of possible estimates (interval estimate) based on the available observations in the database. Then, it collapses the resulting interval estimate to a unique point estimate by combining extreme values of the interval estimate. Based on this unique point estimate, dependency in data is calculated for which it becomes the basis of learning the structure. Node ordering & scoring function are required for the learning process. Ramoni et all concludes that BC method is robust and independence of execution time regarding the number of missing data, assuming the following condition about the input data:

- Attributes in database have discrete values
- There is prior information about node ordering

Bayesian Structural EM (BSEM) Algorithm

This algorithm is a modification of EM algorithm. Originally EM algorithm is an algorithm to estimate missing data. It is then developed by Friedman in order to learn structure of BN from data with missing value (incomplete data) [15]. So, like EM algorithm, BSEM also consists of Expectation step and Maximation step.

The objective of Expectation step is to estimate the probability of missing value by considering other data. The algorithm used is Maximum at Posterior (MAP). It iteratively updates prior probability of variable until convergence. The resulting maximum posterior probability represents conditional probability of complete data.

Maximation step is intended to maximize the structure score. With DAGOPS algorithm, it begins by learning all possible structure, calculate the score, and then choose structure with highest score. The previous structure score is compared with the new one, and the higher score will be chosen. Although the algorithm does not need node ordering information, however, it might be needed to reduce the search space resulting from DAGOPS algorithm.

All of the above algorithms assume that data set are complete and sufficient. However, in real world, there are many cases where data is incomplete including missing values. There are also algorithms that can handling data sets with missing values. Such algorithms are Bound and Collapse [18], Structural

EM [7], EMCMC[14]. They are all based on scoring-based methods.

This paper is intended to describe an improvement of Three-Phase Dependency Analysis (TPDA) algorithm to become TPDA^2 Algorithm, which is usefull for learning BN structure from *missing value and outliers* by integrating scoring-based approach, in more efficient way. This approach can be used as basic algorithm for classification process in Data Mining (DM).

The TPDA algorithm chosen originally is developed using ideas from information theory, which requires at most $O(N^4)$ Conditional Independence (CI) tests to learn an N-variable BN [3]. Compared to the others methods based on "constraint-based algorithms" (as presented aboves) which are normally require exponential number of CI test, TPDA algorithm is chosen due to its improvement in efficiency. Moreover, the TPDA algorithm is correct given a sufficient quantity of training data and whenever the underlying model is DAG faithful [3].

THREE-PHASE DEPENDENCY ANALYSIS (TPDA) ALGORITHM

The TPDA algorithm was developed by Cheng et al [3] using the idea of information theory. Some assumptions are required about the input data:

- The data set is "Independent and Identically Distributed (IID)"
- The cases in the data are drawn IID from a DAG-faithful distribution
- The attributes of a table have discrete values
- There are no missing values in any of records
- The quantity of data is large enough for the CI test being reliable

TPDA consists of three phases of TPDA algorithm, i.e.: drafting, thickening and thinning, as described belows, and complete TPDA algorithm can be found in [3]:

1. Drafting Phase: produces an initial set of edges based on sufficient mutual information test. The draft is a singly-connected graph (a graph without loop) which is found using the Chow-Liu algorithm
2. Thickening Phase: adds edges to the current graph when the pairs of nodes cannot be separated using a set of relevant CI tests.The graph produced will contain all the edges of the underlying dependency model
3. Thinning Phase: each edge is examined and it will be removed if the two nodes of the edge are found to be conditionally independent. Finally, TPDA runs procedure for orienting edges.

TPDA^2 ALGORITHM

The objective of improvement expected (handling missing value and outliers), would be achieved by applying the following general approach:

1. modify TPDA algorithm to learn BN structure from data with missing value, named as TPDA^1
2. enhance TPDA^1 capability to learn BN structure from outliers, named as TPDA^2

TPDA^1

TPDA^1 is modified using the idea of CB Algorithm which combines scoring-based and constraint-based methods (termed as hybrid method). Analyzing of CB algorithm concludes that its schema can be employed for handling missing values. The advantage of the schema is it can generate the node ordering. So, prior information about node ordering is not required.

The schema of TPDA^1 algorithm is defined as following:

1. Given data set with missing value, apply any constraint-based algorithm.
2. Total node ordering is generated then, by applying any algorithm of topological sort
3. Apply any scoring-based algorithm for incomplete data with the resulting node ordering

As a result, like CB Algorithm, TPDA^1 consists of two phase as well. First phase of TPDA^1 is the original algorithm of TPDA. Based on the result of 1st phase, an algorithm for topological sort is applied in order to obtain total node ordering of the resulting DAG. Second phase, then, apply i.e. BC algorithm (scoring-based algorithm) to learn BN structure.

The general process of TPDA^1 can be described to the following phases:

1st phase : TPDA
Intermediate : Topological sort algorithm
2nd phase : BC Algorithm

TPDA^2

Currently, there has not been any algorithm for learning BN structure from data with noise (including outliers). However, there have been a number algorithms for outlier detection such as algorithms proposed by Aggarwal et al, Breunig et al, and Kubica et al. Regarding their characteristics, those algorithms can be combined into TPDA^2.

As a result, general approach for learning BN structure from outliers is as follows:

1. Detect outlier soon as possible to avoid overfitting problem

2. Remove those outliers
3. Apply any learning BN structure algorithms

Based on this approach, TPDA[1] is modified to the following schema, named TPDA[2]:

1st phase : Algorithm for outlier detection & removal
2nd phase : TPDA
Intermediate : Topological sort algorithm
3rd phase : BC Algorithm

Complet algorithm of TPDA[2] is as follow.

Let D is dataset, ε is threshold, G = (V, E) graph structure. Also, let u be an upper bound on number of parents a node may have. Let π_i be the set of parents of node i, $1 \le i \le n$

Step 1 (Algorithm for outlier detection & removal):
Apply "*Partitioned-based Algorithm*" to detect outliers, and Remove them

$u \leftarrow 0$, *Old_*$\pi_i \leftarrow \{\ \}\ \forall i,\ 1 \le i \le n$, and *old_Prob* ← 0

Step 2 (TPDA Algorithm):
Apply *"TPDA algorithm"* if it is first iteration, otherwise apply TPDA-π_i

Step 3:
Let $\pi_i \leftarrow \{\ \}\ \forall i, 1 \le i \le n$

For each node i, add to π_i the set of vertices j such that for each such j, there is an edge $j - i$ in the pdag G

Step 4 (Topological Sort Algorithm):
For each undirected edge in the pdag G choose an orientation as described below

If $i - j$ in a directed edge, and π_i and π_j are the corresponding parent sets in G, then calculate the following products

$$i_{val} = g(i, \pi_i) \times g(j, \pi_j \cup \{i\})$$
$$j_{val} = g(j, \pi_j) \times g(i, \pi_i \cup \{j\})$$

If $i_{val} > j_{val}$, then $\pi_j \leftarrow \pi_i \cup \{i\}$ unless the addition of this edge, i.e. $i - j$ leads to a cycle in the pdag. In that case, choose the reverse orientation, and change π_i (instead of π_j). Do similar thing in case $j_{val} > i_{val}$

The sets $\pi_i, 1 \le i \le n$ obtained by step 4 define a DAG since for each node i, π_i consists of those nodes that have a directed edge to a node i.

Generate a total order on the nodes from this DAG by performing a topological sort on it.

Step 5 (BC Algorithm) :
Apply the *"BC algorithm"* to find the set of parents of each node from incomplete dataset and using the order in step 5.

Let π_i be the set of parents, found by BC, of node i, $\forall i\ \ 1 \le i \le n$

Let *new_Prob* = $\hat{g}(X_i \mid P_i)$

Step 6:
If *new_Prob* > *old_Prob*, then
old_Prob ← *new_Prob*
$u \leftarrow u + 1$
*old_*$\pi_i \leftarrow \pi_i\ \ \forall i\ \ 1 \le i \le n$
Goto Step 2

Else goto Step 7

Step 7:
Output *old_*$\pi_i\ \forall i\ \ 1 \le i \le n$
Output *old_Prob*

Here is below an example of data that will be processed by the algorithm above.

Attribute	Description of Attribute	Value	Meaning of value
VisitAsia	Do the patient visite Asia in the past period ?	a1	*True*
		a2	*False*
Tuberculosis	Do the patient has an tuberculosis ?	b1	*True*
		b2	*False*
TubOrLung	Do the patient has an tuberculosis or cancer ?	c1	*True*
		c2	*False*
X-Ray	How about X-ray test ?	d1	*Abnormal*
		d2	*Normal*
Lung Cancer	Do the patient has an cancer in torax ?	e1	*True*
		e2	*False*
Smoking	Do the patient smoke ?	f1	*True*
		f2	*False*
Bronchitis	Do the patient has an bronchitis ?	g1	*True*
		g2	*False*
Dyspnoea	D the patient has a problem of respiration ?	h1	*True*
		h2	*False*

Assuming some data-set as belows:

	VisitAsia	Smoking	Tuberculosis	Lung Cancer	Tub Or Lung	Bronchitis	X_Ray	Dyspnoea
1	a2	f1	b2	e2	c2	g2	d2	h2
2	a2	f1	b2	e2	c2	g1	d2	h1
3	a2	f1	b2	e2	c2	g2	d2	h2
4	a2	f1	b2	e2	c2	g2	d2	h2
5	a2	f2	b2	e2	c2	g2	d2	h2
6	a2	?	b2	?	c2	g2	d2	?
7	a2	f1	b2	e2	c2	g1	d2	h1
8	a2	f1	b2	e1	c1	g1	d1	h1
9	?	f2	b2	e2	?	g1	?	h1
10	a2	f1	**b3**	e2	C2	g2	d2	h2
11	a2	f1	b2	e2	C2	g1	d2	h1
12	a2	f1	b2	e1	C1	g2	d1	h1
13	a2	f2	b2	e2	C2	g1	d2	h1
14	a2	f2	?	?	C2	?	d2	h1
15	a2	f2	?	e2	C2	g1	?	h1
16	a2	f1	b2	e2	C2	g1	d2	h1
17	a2	f2	b2	e2	C2	g1	d2	h2
18	a2	f2	b2	e2	C2	g2	d2	h1
19	?	f1	b2	e2	?	g2	?	?
20	a2	**f3**	b2	e1	C1	g1	d1	h1
21	a2	f1	b2	e2	C2	g1	d2	h2
22	a1	f2	b2	e2	C2	g2	d2	h1
23	a1	f1	b2	e1	C1	g2	d1	h3
24	a2	f2	b2	**e3**	C2	g2	d2	h2
25	a2	f2	b2	e2	C2	g2	d2	h2
26	a2	f1	b2	**e4**	C2	g2	d2	h2

("?" : unknown value, or missing value)

Refer to the algorithm above, step 1 will detect and delete any dataset recoqnized as "outlier" (consist of records no. 10 (b3), 20 (f3), 24 (e3), and 26 (e4).

Steps 2, 3, and 4 will contract DAG equivalent as shown in Fig. 1, and ignore temporally any records composed by "?" value. As an last result, considering any missing value existing in the DAG in Fig.1, steps 6 and 7 will constract new DAG as shown in Fig.2 below.

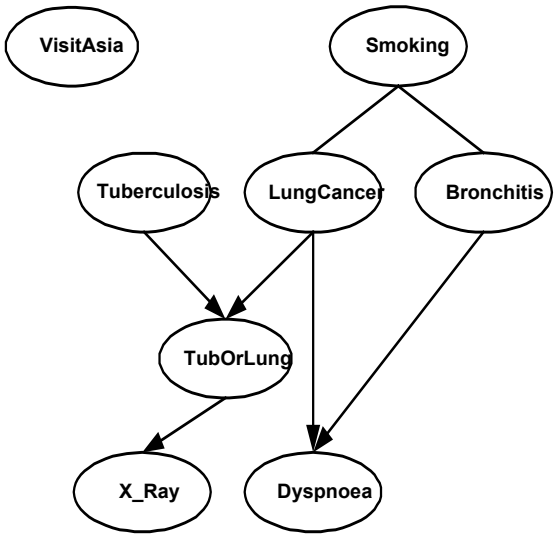

Figure 1. DAG equivalent

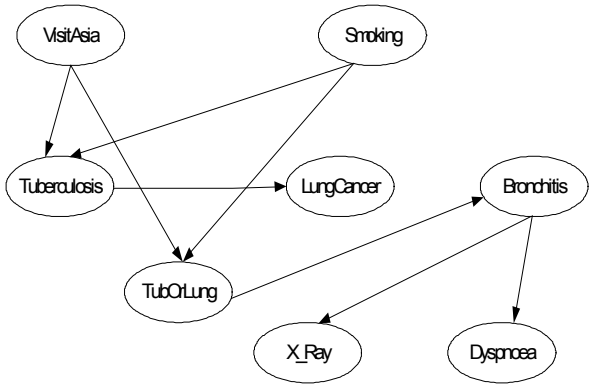

Figure 2. Final DAG

FUTURE RESEARCH

One of the characteristics of data mining that must be fulfill is scalability (the algorithm must be able to proceed any volume of data, not depend to the size of memory/memory based algorithm, large scale classification process). As consequence, TPDA[2] must be improved to be more efficient and can be executed in storage based approach. Currently, TPDA[2] is an memory based algorithm (all training data must be stored in memory, or must be presented once, or can be calssified as memory based algorithm). As consequence, TPDA[2] algorithm can not be yet applied for any voluminous data (bigger than the size of memory).

As an alternative, actually there is a research related to this problem. Basic idea applied is an discriminative model (by aggregating the prediction of multiple classifiers).

REFERENCES

1. Baesens, B., Peterson, M., Castelo, R., Vanthienen, J., *Learning Bayesian network Classifiers for credit scoring using Markov Chain Monte Carlo search*, 2003.

2. Cheng, J., Greiner, R., *Comparing Bayesian Network Classifier,* Proceedings of 15th International Conference on Uncertainty in AI, 1999.

3. Cheng, J., Greiner, R., Kelly, J., Bell, D., Liu, W., *Learning Bayesian Networks from Data: An Information-Theory Based Approach*, Department of Computing Sciences, University of Alberta, Faculty of Informatics, University of Ulster, 2001.

4. Cooper, G.F., Herkovits E., *A Bayesian Method for Induction of Probabilistic Networks from Data*, Machine Learning, Vol.9, 1992.

5. Cooper, G.F., *A Bayesian Method for Causal Modeling and Discovery Under Selection,* Proceeding of 16th Conference Uncertainty in AI, 2000.

6. Dash, D., Cooper, G.F., *Model Averaging with Discrete Bayesian Network Classifiers,* Centre of Biomedical Informatics, University of Pittsburgh, 2003.

7. Friedman, N., *The Bayesian Structural EM Algorithm*, Proceedings of 14th Conference Uncertainty in AI, 1998.

8. Hastie, T., Tibshirani, R., Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2001.

9. Heckerman, D., *Learning Bayesian Networks: The combination of Knowledge and Statistical Data,* Technical Report, MSR-TR-94-09, Microsoft Research, 1995.

10. Hiirsalmi, M., *Method feasibility Study: Bayesian Networks*, Research Report TTE1-2000-29, 2000.

11. Jensen, F.V., *Bayesian Networks and Decision Graph*, Springer, 2001.

12. Kantardzic, M., *Data Mining: Concepts, Models, Methods, and Algorithms*, IEEE Press, 2003.

13. Madden, M.G., *A New Bayesian Network Structure for Classifiation Tasks,* Departemen of Information Technology, National University of Ireland, Galway, Ireland, 2003.

14. Myers, J.W., Laskey, K.B., and DeJong, K.A., *Learning Bayesian Networks from Incomplete Data using Evolutionary Algorithms. http://ite.gmu.edu/~klaskey/papers/gecco99.pdf*

15. Neapolitan, R.E., *Learning Bayesian Networks*, Pearson Education, 2004.

16. Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference*, Morgan Kaufmann, 1988.

17. Rajagopalan, B., Krovi, R., *Benchmarking Data Mining Algorithms*, Journal of Database Management, 2002.

18. Ramoni, M., Sebastiani, P., *Learning Bayesian Networks from Incomplete Databases*, Technical Report, KMI-TR-43, 1997.

19. Saptawati, G.A., Rachmat, H.B., Laksana, D, *Construction of Bayesian Network Structure from Data with K2 and B Algorithm*, National Conference on SNIKTI, 2004.

20. Saptawati, G.A., Herastia, M., Sitohang, B., *Bound and Collapse Method: Data Mining for Bayesian Network Structure from Incomplete Data*, National Conference on SIIT, 2005

21. Saptawati, G.A., Sitohang, B., *Hybrid Algorithm for Learning Structure of Bayesian Network from Incomplete Databases,* International Conference on ISCT, Beijing, 2005.

22. Singh, M.; Valtorta, M., *Construction of Bayesian Network Structures from Data: a Brief Survey and an Efficient Algorithm*, Dept. of Computer Science, University of South Carolina, Columbia, USA, 1995.

23. Spirtes, P., Meek, C., *Learning Bayesian Networks with Discrete Varibales from Data*, Proceedings of the Conference on Knowledge Discovery & Data Mining, 1995.

24. Steck, H., Tresp, V., *Bayesian Belief Networks for Data Mining*, Siemens AG, Coroprate Technology, Information and Communication, 2003.

25. Street, W., Kim, Y., *A streaming ensemble algorithm (sea) for large scale classification*, International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2001.

26. Tan, P., Steinbach, M., Kumar, V., *Introduction of Data Mining*, Pearson Education Inc, 2006.

27. Chu, F., Wang. Y., Zaniolo, C., *An Adaptive Learning Approach for Noisy Data Streams*, Technical Report 040029, UCLA, CSD, 2004.