BAYESIAN NETWORK STRUCTURAL LEARNING FROM DATA: AN ALGORITHMS COMPARISON

Francesco Colace, Massimo De Santo, Mario Vento
Dipartimento di Ingegneria dell'Informazione ed Ingegneria Elettrica, Università degli Studi di Salerno, Via Ponte Don Melillo 1, 84084, Fisciano (Salerno), Italia

Pasquale Foggia

Dipartimento di Informatica e Sistemistica, Università di Napoli "Federico II", Via Claudio, 21, 80125 Napoli, Italia

Keywords: Bayesian Networks, Structural Learning algorithms, Machine Learning

Abstract: The manual determination of Bayesian Network structure or, more in general, of the probabilistic models, in particular in the case of remarkable dimensions domains, can be complex, time consuming and imprecise. Therefore, in the last years the interest of the scientific community in learning bayesian network structure from data is considerably increased. In fact, many techniques or disciplines, as data mining, text categorization, ontology description, can take advantages from this type of processes. In this paper we will describe some possible approaches to the structural learning of bayesian networks and introduce in detail some algorithms deriving from these ones. We will aim to compare results obtained using the main algorithms on databases normally used in literature. With this aim, we have selected and implemented five algorithms more used in literature. We will estimate the algorithms performances both considering the network topological reconstruction both the correct orientation of the obtained arcs.

1 INTRODUCTION

A Bayesian Network is a graphical model for probabilistic relationship among a set of variables. In the last period this modelling has become a popular representation for encoding uncertain knowledge in expert systems (Heckermann, 1995). It could be useful and interesting to learn the structure of Bayesian Networks given the data. The main aim of structural learning algorithms is to make clear the relationship between the entities of the domain and to specify the causality ties starting from the observations of domain variables values. In very general terms the different learning methods of probabilistic network structures from data can be classified into three main groups (Singh, 1995): some of these methods are based on linearity and normality assumptions others are more general but require extensive tests of independence relations (Fung, 1990)(Pearl,1991); others are based on a Bayesian approach (Cooper, 1992). In this paper a comparison between the obtained results of some structural learning algorithms is performed. With this aim we have selected five algorithms among the most important: the Bayesian algorithm (Heckermann, 1995), K2 (Cooper, 1992), K3 (Bouckaert, 2002), PC (Spirtes, 2001) and TPDA (Cheng, 1997). The paper is organized as follows: in the section 2 we will describe the various approaches and the general structure of an algorithm for the structural learning. In the section 3 we will describe the selected networks and the reference data sets. and the obtained results. We will finish with a brief set of conclusion.

Colace F., De Santo M., Vento M. and Foggia P. (2004).
BAYESIAN NETWORK STRUCTURAL LEARNING FROM DATA: AN ALGORITHMS COMPARISON.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 527-530
DOI: 10.5220/0002625305270530

2 ALGORITHMS OF STRUCTURAL LEARNING

The main aim of structural learning algorithms is to point out the relations between the entities of the domain and to specify the causality ties starting from the observations of domain variables values. In general, the structural learning algorithms can follow two main research lines: a not Bayesian approach (dependence analysis) and a Bayesian approach (search and score). In order to infer automatically the existence of dependence relations between the domain variables algorithms, that follow the not Bayesian approach, execute statistical independence tests on the network data set samples. Algorithms based on Bayesian approach codifies the uncertainty on the structure of a dominion $X = \{X_1,...,X_n\}$ introducing an aleatory variable M. The states of this aleatory variable represent the possible structures associated to X. After this phase the algorithm chooses the model m that maximizes the "a posteriori" probability P(m|D), where D is the data set. In the next paragraphs we will describe in detail the main characteristics of five algorithms that we aim to compare. These algorithms are the most important in literature representing the main approaches in the structural learning field.

2.1 The Bayesian algorithm

The bayesian algorithm resolves the problem of the Structural Learning from data determining the structure **m** that maximizes the probability p(**M** = **m**|D), where **M** = {$\mathbf{m}_1$, ...,$\mathbf{m}_n$} is a set of models that contains the *true model* of a domain **X,** D is the set of the samples. According this approach if we have two models $\mathbf{m}_i$ and $\mathbf{m}_j$ representing the domain **X**, we will choose $\mathbf{m}_i$ if p($\mathbf{m}_i$|D) > p($\mathbf{m}_j$|D). So we choose as scoring function the logarithm of p(D|**m**) that with simple passages becomes equal to: **Score(m) = log (p(m|D)) = log(p(m))+log(p(D|m))-log(p(D)) =log(p(D|m)).**
The introduced approximation is acceptable because both log(p(D)) both the prior knowledge on the model, log(p(**m**)), in the hypothesis of complete "a priori" ignorance on the domain structure, are constant values. The statistical criterion of reference is the Maximum Likelihood while if the contribution of log(p(**m**)) is not negligible the algorithm can use the Maximum a Posteriori (MAP) principle. In this paper we will refer to the algorithm based on a "model selection" approach and will make also the hypothesis that the *best model* has the maximum of the distribution p(**m**|D) localized around a model **μ**. In order to select the **μ** we introduce a function whose value is higher when the model **m** is closer to **μ**.

2.2 The K2 algorithm

This algorithm derives from a bayesian algorithm in which the assumption of complete ignorance on the probability distribution of the models is made (Cooper, 1992). The K2 procedure differs from a typical bayesian algorithm also for the initialization phase: while in the bayesian approach we could use a starting graph with the "a priori" knowledge of an expert that can describe a starting topological ordering (from fathers to sons nodes) of the nodes. In fact this information reduces the cardinality of the searching space of the models. In this approach the scoring function is defined as: $p(B_s, D) = P(B_s)\prod_{i=1}^{n} g(X_i, \pi_i)$ where D is a data set of the **m** complete cases and B_s is the structure of a bayesian network. The function $g(X_i,\pi_i)$ represents the variations obtained in the scoring function after the introduction of a new dependence relation so of a new father node for X_i. The core of this approach is a greedy search algorithm where at its beginning no nodes have fathers. A real disadvantage of this approach is the impossibility of delete an arc after its introduction in the network.

2.3 The K3 algorithm

This type of algorithm, introduced in the paper (Bouckaert, 2002), is based on a bayesian approach, but as in K2 algorithm gives a new definition for the scoring function. In this case the scoring function is based on the Minimum Description Length (MDL) metric. The MDL approach is so formalized: the learned network must minimize the *total description length* defined as: the description length of the samples and the description length of a pre-existent network structure supplied from an expert or generate in a previous process of learning In this approach samples and pre-existent network structure are independent in order to elaborate them separately. The scoring function in this approach is so defined:

$$L(B,D) = \log(P(B)) + N * H(B,D) - \frac{1}{2} k \log(N)$$

$$H(B,D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} -\frac{N_{ijk}}{N} \log(\frac{N_{ijk}}{N_{ji}})$$

$$k = \sum_{i=1}^{n} q_i (r_i - 1)$$

where B represents a possible structure, D is the n samples of data set, the value r_i represents the states number of node X_i, q_i is the number of possible configurations of father nodes for each

node X_i and N_{ijk} are the occurrences in D of X_i with state k and fathers configuration j.

2.4 The PC algorithm

This algorithm is based on a constraint satisfaction approach (Spirtes, 2001). The PC procedure consists of an initialization phase where a fully connected DAG, associated to a domain X, is set up and an iterative phase that searches the implicit relations of independence between the samples. In every iteration we consider a set C(X,Y) of adjacent nodes to X without Y with cardinality greater or equal to the current n value. So for every subset S, with cardinality n and extracted from C, the algorithm carries out the order n statistical test in order to determine if X and Y are d-separated from S. In the affirmative case the arc X-Y is removed and a new S set is examined with the same procedure. After the investigation of all possible S in C the n value is increased and the algorithm is repeated until C has cardinality greater or equal to n. In order to determinate the arcs orientation the algorithm uses consideration based on conditional independence.

2.5 TPDA Algorithm

The TPDA algorithm is a dependence-based algorithm. It divides the process of learning in three phases: Drafting, Thickening and Thinning. The Drafting phase produces an initial relations set through test on cross entropy value between the variables of the domain. After this phase we obtain a graph where it is present only a path between two nodes. The second phase, "thickening", adds arcs to the single connected graph if it is not possible to d-separate two nodes. The resulting graph contains all arcs of the true model and some extra-links. These false arcs are produced by errors in the test. The third phase, "thinning", consists in the examination of all arcs and its exclusion if the two nodes are conditionally independent. At the end of this phase the algorithm orients arcs with an approach similar to PC algorithm.

3 EXPERIMENTAL RESULTS

The main idea of this paper is to compare some of most important structural learning algorithms. We have implemented all algorithms previously described and we have tested them using seven bayesian networks and their relative datasets. A briefly description of networks and datasets is showed in the next paragraph.

3.1 Test Networks description

We have selected seven networks and their related dataset in order to test the algorithms previously described. In table 1 there is a briefly description of all selected networks and related datasets.

Table 1: Analysed Networks and Datasets

Network Name	Nodes Number	Arcs Number	Data Set Samples
Alarm (Pearl, 1991)	37	46	10.000
Angina (Cooper, 1992)	5	5	10.000
Asia (Glymour, 1987)	8	8	5.000
College (Singh, 1995)	5	6	10.000
Led (Fung, 1990)	8	8	5.000
Pregnancy (Buntime, 1996)	4	3	10.000
Sprinkler (Suzuki, 1999)	5	5	400

We used the previously described algorithms on these networks. We have experimented the algorithms using two different sorting for the nodes of the networks: ordered (correct sorting of node starting from fathers to sons) and inverse. We have choosen two different sorting in order to test in any case the performances of algorithms. We have defined two indexes:

$$\text{Topological Learning} = \frac{\sum \text{Correct Arcs}}{\sum \text{Correct Arcs} + \sum \text{Missing Arcs} + \sum \text{Added Arcs}}$$

$$\text{Global Learning} = \frac{\sum \text{Correctly Oriented Arcs}}{\sum \text{Correctly Oriented Arcs} + \sum \text{Wrongly Oriented Arcs} + \sum \text{Added Arcs} + \sum \text{Missing Arcs}}$$

The first index measures the ability of the algorithm in the learning of correct topology of the net. The second index, instead, measures the ability of the algorithm in the learning of correct networks (topology and correct orientation of arcs). In figures 1 and 2 we have depicted the average indexes values obtained by every algorithm in the learning processes of the various networks. In figure 1 we have the results for ordered nodes with and in figure 2 the results for inverse nodes. Algorithms that are based on a scoring function maximization approach have the best results in the case of ordered starting structure. In particular the K2 algorithm has the best performance: 88% as topological index and 88% for the global index. The constraint-based algorithms have the worst results and show an important difference between the two indexes (18% for PC and 24% for TPDA). So we can say that these algorithms also when are able to identify the topology of the network often

mistakes on the orientation of the arcs. On the other hand, if we use inverse nodes sorting the bayesian algorithm, in particular K2 and K3, deteriorate their performances: 29% for Topological Learning index and 67% for Global Learning index for the K2 algorithm and 27% for Topological Learning index and 67% for Global Learning index for the K2 algorithm. In particular, they are not able to learn the correct topology: arcs and their orientations. The performances of constraints based algorithms remain fundamentally the same.

4 CONCLUSION

In this paper we have described some algorithms for the structural learning of Bayesian networks. We have selected five algorithms able to represents the most important and common approaches that are present in literature. We have implemented, according the authors specifications, these algorithms and we have tested them on the most common datasets. We have made experimentations in two different ways: ordered and inverse starting nodes sorting. We can say that in the case of correct starting nodes sorting algorithms based on a Bayesian approach and more in general on the maximization of a predefined scoring function obtain better results than algorithms based on statistical independence tests. On the other hand these algorithms have a more stable behaviour to nodes sorting. An interesting future work could be a complete characterization of examined algorithms trying to make clear the relationship between the starting sorting of the networks nodes and learned networks.

REFERENCES

Singh, M., Valtorta, M., 1995. Construction of Bayesian Network Structures from Data: a Brief Survey and an Efficient Algorithm. *In International Journal of Approximate Reasoning*

Fung R. M., Crawford S. L., 1990. Constructor: a System For The Induction of Probabilistic Models. *In Proceedings of AAAI-90*

Pearl J., Verma T., 1991. A Theory of Inferred Causation, *In Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann

Cooper G. F., E. Herskovits, 1992. A Bayesian Method For The Induction of Probabilistic Networks From Data, In *Machine Learning, 9*

Buntime W., 1996. A Guide to the Literature on Learning Probabilistic Network from Data, In *IEEE Transaction on KDE*, vol. 8, no. 2

Suzuki J., 1999. Learning Bayesian Belief Networks Based on the MDL Principle: an Efficient Algorithm Using the Branch and Bound Technique, In *IEICE Trans. Inf. & Syst., Vol. E82*

Bouckaert R., 2002. Probabilistic Network Construction Using the Minimum Description Length Principle, *Lecture Notes in Computer Science, Vol. 747*

Spirtes, P., Glymour, C., Scheines, R, 2001. Causation, Prediction and Search, *In MIT press*

Cheng , J., Bell, D., Liu, W., 1997. Learning belief networks from data: an information theory based approach, In *Proceedings of the Sixth ACM International Conference on Information and Knowledge Management*

Heckermann, D., Geiger, D., and Chickering, D., 1995. Learning Bayesian Networks. In *The Combination of Knowledge and Statistical Data. Machine Learning, 20(3)*

APPENDIX

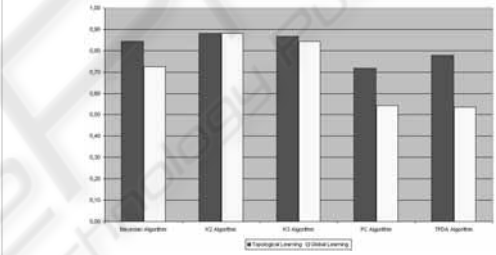

Figure 1: Topological and Global Average values obtained using ordered data sets

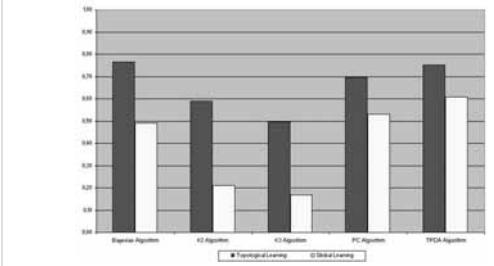

Figure 2: Topological and Global Average values obtained using inverse data sets