LEARNING THE STRUCTURE OF BAYESIAN NETWORK FROM SMALL AMOUNT OF DATA

Adina COCU, Marian Viorel CRACIUN, Bogdan COCU

Department of Computer Science and Engineering,
University "Dunarea de Jos" of Galati, 2 Stiintei, 800146, Romania
Adina.cocu@ugal.ro, Marian.craciun@ugal.ro, Bogdan.cocu@windriver.com

Abstract: Many areas of artificial intelligence must handling with imperfection of information. One of the ways to do this is using representation and reasoning with Bayesian networks. Creation of a Bayesian network consists in two stages. First stage is to design the node structure and directed links between them. Choosing of a structure for network can be done either through empirical developing by human experts or through machine learning algorithm. The second stage is completion of probability tables for each node. Using a machine learning method is useful, especially when we have a big amount of leaning data. But in many fields the amount of data is small, incomplete and inconsistent. In this paper, we make a case study for choosing the best learning method for small amount of learning data. Means more experiments we drop conclusion of using existent methods for learning a network structure.

Keywords: Bayesian network, machine learning algorithm, structure learning

1. REPRESENTATION WITH BAYESIAN NETWORKS

Bayesian networks are used for uncertain knowledge representation through graphical models. A Bayesian network is a directed acyclic graph, in which the nodes have associated probabilities. Graphical models are practical, because there are intuitive and can be easily implemented in reasoning algorithms. Modeling the imperfection of information and different aspects of interest can be done with mathematical probability theory. Bayesian networks have the advantage of running reasoning methods. These methods combine represented knowledge with observed evidence in order to obtain new knowledge. Classical probability theory, which is the foundation of uncertainty representation in Bayesian networks, is a proved theory with obvious and unquestionable axioms, and also with advanced applications. The controversies relatives to the probability theory make allusion to the different interpretations (chance calculation, subjectivity, frequency) and also to the manner in which probability values can be obtained (statistical, empirical). The probability theory allows uncertain events representations using probability measure. The probability measure is defined through the frequency of the appearance of an event, based on the anterior observations. From this definition comes the idea to develop learning methods based on occurrence of a specific situation into a searching space. The searching space is composed from a set of events about a specified domain. The domain is defined using a set of interest variables, that can be dependence one of the others. Each variable can have two or more possible values. A situation from searching space means that the variables of interest take a combination of a certain values. The variables from domain of interest can be represented through the nodes of a Bayesian network and the dependences between them can be represented through directed edges in the graph structure. The uncertainty is represented throughout:

- Prior probability for the nodes without parents: P(A=a) – the probability that the variable A takes the value a;

This paper was recommended for publication by Cornelia Tudorie

- Conditional probability for the nodes with edges that come unto them: P(A=a|B=b,C=c) – the probability that the variable A takes the value a in the condition that the variables B and C take the values b and c.

So, the strength of dependence relation between nodes it is done with conditional probabilities. From mathematical point of view, the meaning of "|" in P(A|B) it is very well defined, but when it is used for model real world application an higher attention must be paid, because of existence of many interpretation for dependence relation (causal, prediction, structural) (Hulswitt, 2002).

2. LEARNING ALORITHMS FOR BAYESIAN NETWORKS

The networks with a small number of nodes (maximum 10) can be empirical developed. Establishing the structure and probability tables for a network is made by the knowledge engineer, relying on experience in modeling domain. In this situation, many times, the modeled system does not respond to the expectations. As a result, developing process must be reloaded in order to adjust the dependences and probabilities measures. In practice, when are available dates corresponding to different cases, then can be used learning algorithms. These algorithms proceed to the network discovery (structure, but also the probabilities tables) from the learning dataset. There are a set of reasons for it is indicated to use learning algorithms for Bayesian network development, instead of empirical design:

- There is not find always an expert in the field of modeling in order to create the network.
- If the expert exists, he can not be objective.
- Expert in the field is not also an expert in system modeling, so needs a knowledge engineer to deal effectively with modeling and implementation.
- If there are more experts, then it should be put in place an evaluation system for weighting of each proposed solution.
- The acquisition and modeling processes are costly in time and resources.
- There are different amounts of data, which are cheap and can be used to build models using learning algorithms.

In the field of artificial intelligence, there are many learning algorithms that allow knowledge discovery from data sets and these are already implemented in applications.

The algorithms for learning independences between variables in graphical models are based on Bayesian theory (Bayes theorem, marginalization, decomposition theorem) (Russell, Norvig, 1995).

Based on these considerations, emphasize that the structure and links between concepts can not be created by only one person (or few persons) for several reasons: persons should be experts of the domain (usually hard to find more than few such people), the experts already have solid knowledge and often ignores a number of exploratory needs in the designing process, it is difficult to predict all the possible combinations of values for the variable of interest.

Using a learning algorithm it is possible to create a network structure starting from a set of data. Learning algorithm results may be different depending on the completeness of the learning data set, the capacity of these data to cover all possible situations in the field of learning, noise of data, network type (classification or dependences).

For the situations where it is not know neither network structure or either the probability tables, there are already developed several algorithms for learning from data. First, it is necessary to apply a method to discover the dependencies between variables (Hill Climbing - HC, Tabu search - TS, Repeated Hill Climbing - RHC, Simulated Annealing - SA, Genetic search - GS). Then, for learning probability tables will be applied a method of expectation maximization. So, appear the question of selection a suitable algorithm for a given situation.

3. EXPERIMENTS

First, the study is made on the "iris" benchmark, which includes data corresponding to four predictive numerical attributes (variables named "sepallength, petallength, sepalwidth, petalwidth"), by which is classified a class (variable "class" with three possible values). Dataset includes 150 cases with an equal distribution upon the three classes. The numerical variables have real numerical values. The different learning algorithms will be run over the entire dataset and the results will be compared. The results were calculated assuming that the structure is not naive (and there may be dependencies between attributes, not only between attributes and class, and attributes are not considered independent between them).

For experiments it is used Weka application (Bouckaert, 2008), which has implemented learning algorithms for structure classification. In every running experiment the variable "class" was chosen for classification. In HC, TS and RHC it is allow to choose the maximum number of parents for each node. In the accomplish experiment the maximum number of parents was set to three.

Evaluating dependencies between variables usually require calculation of distance or similarity/ correlations among measured dataset. The most common choices for these measures are Pearson correlation or Euclidean distance. The Euclidean

distance is used especially in problems for classification and clustering, because it permits comparison of variable from dataset. The Pearson correlation measure indicates the strength and direction of a linear relationship between two random variables organized in two columns. In this study it will be used the correlations function between two different attributes that is expressed by the formula (1). The correlations with values over 0.85 indicate a close relationship between the variables (Gall, Borg, 1996).

$$\text{Correl}(\text{Col}_1,\text{Col}_2) = \frac{\sum(\text{Average}(\text{Col}_1)-\text{Col}_1)(\text{Average}(\text{Col}_2)-\text{Col}_2)}{\sqrt{\sum(\text{Average}(\text{Col}_1)-\text{Col}_1)^2\sum(\text{Average}(\text{Col}_2)-\text{Col}_2)^2}} \quad (1)$$

Then it will be run the classifying algorithms and will be compared the correlations between attributes and the dependencies found by each algorithms. The results are presented in table 1. In the table, the second column shows the correlation values. The first four correlation values are above 0.85, which indicates a close relationship between the variables. In the fifth row of the table the correlation is set to the 0.81 value, which can be interpreted as a satisfactory correlation between two variables, but can also signify a possible group correlation (which can be seen in Figure 1, "sepallength" is link by the "petalwidth" through "petallength"). The figure 1 contains the graphical representation of network structure obtained by the methods from HillClimber category (HC, TS and RHC), which all conduct to the same structure.

In addition, the classification methods found a link (between "sepalwidth" and "petalwidth") that has a negative correlation factor. On the other hand, algorithms can not find dependencies between variables that still have a high correlation factors (for example, "petallength" and "petalwidth" with 0.96). These results lead to the conclusion that the learning algorithms search those links with a big prediction influence over classification variable. So, the algorithms are channeled through class prediction capacities and not for discovery the most powerful dependences.

In the table 1, in the columns corresponding to the different classification algorithms, dependencies discovered by each method separately are marked with "x". From these results, one can draw the following conclusion: if the data set is completely and evenly distributed, then all methods are able to determine the dependencies between variables. Finding a direct or indirect dependency between the variables "sepallength-petalwidth" (by SA - simulated annealing algorithm) is questionable.

In conclusion, if the dataset is complete and consistent, then almost all classification algorithms lead to similar results, that can be seen in figure 1.

Similar experiments were performed for the same "iris" dataset, but were removed a set of learning cases (uniformly disctributed according with classification variable) that represent 25%, 50% and 75% from the initial dataset. Thus, in the dataset remained an equal number of cases for each of the three values of classification variable. In all these cases, the resulting network structures after classification have missing link between "petalwidth" and "petallength". This concludes that an incomplete learning data set, lead to incomplete results, in the direction of losing a part of links between dependent variables.

Table 1 Comparative classification results with different methods of learning Bayes network structure for dataset "iris".

Variable correlations	Correl. values	HC	TS	RHC	SA	GS
sepallength-petallength	0.87	x	x	x	x	x
petallength-petalwidth	0.96	x	x	x	x	x
petallength-class	0.95	x	x	x	x	x
petalwidth-class	0.96	x	x	x	x	x
sepallength-petalwidth	0.81				x	
sepalwidth-petalwidth	-0.37	x	x	x	x	x

Fig.1. The Bayes network structure for "iris" dataset

For another set of experiments, from the "iris" dataset was dropped an unbalanced number of cases for each possible value of classification variable. Surprisingly, in the case of reduction with 25% of the initial dataset, the results were similar with those obtained for the full data set, the algorithms being able to find all dependence links. This it was possible because, in this particularly case, the removed cases don't destroy the consistence of the dataset. In the real world situations, usually an incomplete dataset it is also inconsistent.

In the case of decreasing the learning dataset with percentages greater that 25%, the quality of obtained structure decreased, because the algorithms found strange dependence connections and not discover the most important ones.

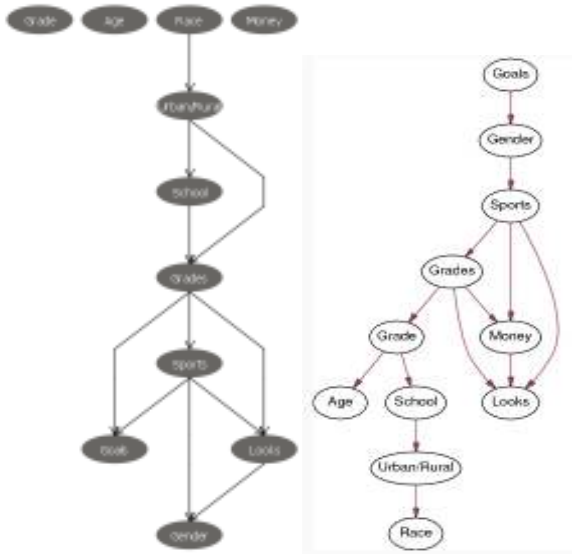
The same types of experiments were run on another data learning set proposed by B-course application (Myllymaki *et all*, 2002). This dataset, called "popular kids", doesn't follow the classification of a variable depending on other variables. It is design for discover the dependences between all the dataset variables. The dataset has 11 variables with 478 training cases. Five of the variables have nominal values and six of them have between 2 and 6 distinct integer numerical values, which show a large distribution range of possible cases. This dataset is an example for seeking the strongest dependences between variables and not for classification of one variable depending from the others.

Direct correlations between variables calculated with correlation function (1) have low correlation values (smaller than 0.4). The cause of these small values of correlation function can be nominal values of variables. In order to calculate the correlation function, the nominal values must be transformed into numerical ones. In this process it is possible to loose information. Thus, in all dataset it is obtained only one correlation with value of 0.85 (between two numerical variables). This means that the variables are not directly correlated to each other. In this context, the correlation function is not a good indicator for existence of a link between variables. So, forward, the structure learning algorithms should look for group dependences.

The experiments in Weka were done for choosing different nominal class variables. For the cases with complete training set, algorithms HC, TS, RHC from Weka application conduct to the same results. This is because HC, TS and RHC algorithms are similar and use the same quality measure. The SA algorithm from Weka has inclination to discover lots of links between variables. The GS algorithm do not offer results in acceptable time interval. For the same algorithm, the same input data, but for a different class variable, the results are different. There is little number of dependences that are preserved between results for the same running conditions but a different class variable. This is because the algorithms from Weka are for classification, but the "popular kids" dataset is for dependence discovery. This lead to the conclusion that learning structure algorithm implemented in Weka are highly dependent from class attribute.

For exemplification, in order to run the Weka experiments, one of the variables (named "goals") was chosen to be class. Thus, three variables are found to be independent from the others. The other variables are linked like in figure 2.a. The results obtained for this dataset with another application, B-course, are show in figure 2.b. The B-course has two running components: one for classification purposes and one for dependence discovery. In this experiments it is used the dependence discovery algorithm from B-course Web application.

In the figure 2 can be observed that the running of B-course, a special design application for dependence discovery, conduct to the better results, because the algorithm is capable to determine more dependences. The disadvantage of B-course is that the each time when run the application, the results can be different. This means that the power of criterion for choosing a link between variables is weak.



a b

Fig.2. The dependences between variables for "popular kids" dataset

If the dataset is diminished uniformly (for the variable "goals" considering being class) with 25%, then the classification algorithms from Weka discover an average of six links between those eleven variables of "popular kids" dataset. Dependences found by running different algorithms are different and depend on the algorithm type, on the particular

settings of each algorithm and on chosen variable for class. From these experiments can make another observation: for nominal variables are found more links easier, compared with the variables that have numeric values. The Weka application, in order to work with Bayesian network structure learning algorithms, transforms the numeric values into discrete values. In this process is lost data accuracy. There are more discretization methods, but in Weka it is used MDL method (Fayyad, Irani, 1993) to discretize numeric attributes into nominal ones, based on the class information. This leads to the strengthening the influence of class variable on the learned structure of a network.

In conclusion, the learning algorithms of a structure network for classification do not lead also to the satisfactory results for discovery dependence between majorities of variables of interest.

4. CONCLUSIONS

After these experiments we have reached to the several conclusions:

- any of the algorithms proposed by the Weka application are useful for classifying networks;
- learning algorithms based on classification are not able to find strong dependencies between variables of interest or group dependencies without be influenced by class variable;
- for discover through learning dependence network between variables is necessary developing of others search algorithms or modifying the existing ones, so they do not take into account the classification variable;
- the Hill Climbing algorithms (HC, RHC, TS) are more suitable for modification for dependence discovery, because allow setting of maximum number of parents for a node;
- the Simulated Annealing algorithm is most suitable for using in problems that try to find a great number of dependences between interest variables;
- satisfactory results are obtained even if the training data set is not complete, the condition being to contain those data for the most representative cases;
- the quality of learned network decreases with decrease completeness of dataset;
- in order to learn a dependence structure is a requirement to find and calculate a quality measure for the entire network;

5. REFERENCES

Bouckaert R. (2008) *Bayesian Network Classifiers in Weka for Version 3-5-7*, http://www.cs.waikato.ac.nz/~remco/weka_bn/

Gall M. D., Borg W. R., Gall J. P. (1996),*Educational research: An introduction*, White Plains, NY: Longman Publishing Group.

Hulswit M. (2002) *Causality and Causation: The Inadequacy of the Received View*, In João Queiroz (ed.). The Digital Encyclopedia of Charles S. Peirce.

Myllymaki P., Silander T., Tirri H., Uronen P. (2002) *B-Course: A Web-Based Tool for Bayesian and Causal Data Analysis*, International Journal on Artificial Intelligence Tools, Vol 11, No. 3, pp. 369-387.

Russell S. J., Norvig P. (1995) *Artificial Intelligence - A Modern Approach*, Prentice-Hall Inc., ISBN 0-13-103805-2.

Fayyad U., Irani K. (1993) *Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning*, JPL Technical Report.