Stat Methods Appl (2010) 19:127–139
DOI 10.1007/s10260-009-0116-1

ORIGINAL ARTICLE

On using Bayesian networks for complexity reduction in decision trees

Adriana Brogini · Debora Slanzi

Accepted: 5 March 2009 / Published online: 24 March 2009

Abstract In this paper we use the Bayesian network as a tool of explorative analysis: its theory guarantees that, given the structure and some assumptions, the Markov blanket of a variable is the minimal conditioning set through which the variable is independent from all the others. We use the Markov blanket of a target variable to extract the relevant features for constructing a decision tree (DT). Our proposal reduces the complexity of the DT so it has a simpler visualization and it can be more easily interpretable. On the other hand, it maintains a good classification performance.

Keywords Bayesian networks · Decision trees · Markov blanket · Complexity reduction · Classification

1 Introduction

Most real world domains are complex systems in which the analysis must identify the aspects of the system and their main interactions; this is a difficult task that involves great investment of time, effort and expertise. There is a growing interest in knowledge discovery in database (KDD) which is the process of identification of knowledge from a database leading to specify intuitive and easily interpretable models (Mitchell 1997). In this context, supervised machine learning (or, more specifically, classification) is

A. Brogini (✉)
Department of Statistics, University of Padova, Via Cesare Battisti 241, 35121 Padova, Italy
e-mail: brogini@stat.unipd.it

D. Slanzi
Department of Statistics, University Ca' Foscari of Venice, San Giobbe, Cannaregio 873, 30121 Venice, Italy
e-mail: debora@unive.it

an induction procedure typically presented with a set of training instances, where each instance (or case, example) is described by a vector of feature (or variable, attribute) values and a class label, or target. The task of the induction algorithm is to induce a classifier that will be useful in classifying future cases. The classifier is a mapping from the space of feature values to the set of class values. Several different representation formalisms are used to describe the extracted knowledge: in this paper we focus on the decision tree (DT for short) which is validated through good classification performances. DT induction has been extensively studied in the machine learning and statistics communities as a solution of classification tasks (Breiman et al. 1984; Quinlan 1986, 1993). Some real domains give us a wealth of features and/or very large databases to use for learning, and often the tree produced by the induction algorithms are not comprehensible to users due to their size and complexity. Many tree simplification approaches have been proposed, which can be grouped in five categories (Breslow and Aha 1997). We focus on the methods of database restriction, by eliminating certain case features from consideration by the search process. Feature selection is an effective technique in dealing with dimensionality reduction; in classification it is used to find a good subset of relevant features such that the overall accuracy of classification is increased, or not significantly decreased, while the data size is reduced and the comprehensibility is improved.

There are a number of different approaches to feature subset selection which can be organized into three methods depending on how the feature selection search is combined in machine learning with the construction of the classification model: filter, wrapper and embedded. For a review on this topic we refer to Saeys et al. (2007). The Bayesian networks (BN) (Pearl 1988; Cowell et al. 1999; Jensen 2001) will be used for the feature selection problem, by identifying the joint probability distribution of the features and the class and by selecting the minimal conditioning set through which the class is independent from the remaining variables. Afterwards, the DT induction algorithm is applied to the entire training set using only the relevant features discovered by the BN. We compare the results of this approach with those obtained by using different feature subset selection methods. We consider only discrete variables and all variables are observed.

The work is organized as follows: Section 2 introduces the BN and presents our proposal method for selecting the features used in the DT construction; Sect. 3 presents the databases and the learning algorithms to test the approach, Sect. 4 presents the experimental results and concludes with the discussion.

2 Bayesian networks

Let be $\mathbf{X} = \{X_1, \ldots, X_n\}$ a set of random variables, P a joint probability distribution over $\mathbf{X}$ and G a direct acyclic graph (DAG). A Bayesian network, BN for short, $B = (\mathbf{X}, G, P)$ is a graph-based model of P that capture properties of conditional dependence and independence between variables of $\mathbf{X}$, represented as nodes in G; all nodes correspond one-to-one to members of $\mathbf{X}$. If there is an edge pointing from variable X_i to variable X_j, it is said that X_i is a parent of X_j and X_j is a child of X_i.

The *Markov condition* is the basic property of a probability distribution modeled by a BN B and its DAG G. Recall that two variables X and Y are probabilistically independent if the joint probability distribution factors like $P(X, Y) = P(X)P(Y)$; this is written as $X \perp Y$. Two variables X and Y are conditional independent given a variable Z if $P(X, Y|Z) = P(X|Z)P(Y|Z)$, denoted as $X \perp Y|Z$. These concepts can be generalized for variable sets. In a BN B, the graph G encodes the Markov condition if each node X_i is probabilistically independent of all non descendants given its parents. From this condition the so called *chain rule* for BNs follows immediately: a BN can be factorized as a product, for all variables in the network, of their probabilities conditionally on their parents only

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|Pa(X_i)).$$

where $Pa(X_i)$ denotes the set of X_i parents. The conditional probability distributions $P(X_i|Pa(X_i))$ are also called the *parameters* of the BN.

d-Separation (Pearl 1988) is a graphical criterion which captures all in/dependence relations implied by the Markov condition on the random variables $\mathbf{X}$ represented in G: with d-separation the structure of P, modeled by B, can be easily investigated. Two variables X_i and X_j are d-separated given a subset of variables $\mathbf{S} \subset \mathbf{X}$ if and only if there exists no adjacency path between them (i.e. a path ignoring the ordering of the edges) such that (1) every collider (a collider being a node with two incoming edges) on the path is in $\mathbf{S}$ or has a descendant in $\mathbf{S}$, (2) every non-collider node on the path is in $\mathbf{S}$ (Glymour and Cooper 1999).

It is usually assumed that in addition to Markov condition, which is part of definition of a BN, another condition called *faithfulness* is also fulfilled. The graph G of a BN is faithful to a joint probability distribution over a set of variables $\mathbf{X}$ if and only if every dependence entailed by G is also present in P. A distribution P over a set of variables $\mathbf{X}$ is said to be faithful if and only if there exists a DAG G satisfying the faithfulness condition. We say that a data-generating process K is faithfully represented by $B = (\mathbf{X}, G, P)$ if K in the sample limit produces data with joint probability distribution P, and B is faithful to P. It follows from Markov condition that every conditional independence entailed by G is also present in P. Thus together faithfulness and Markov conditions establish a close relation between a graph G and a probability distribution P and allows us to associate statistical properties of P with graph properties of G.

A *Markov Blanket* of a node X_i, denoted as $MB(X_i)$, is a minimal set of variables, such that every other variable is independent of X_i, given $MB(X_i)$, i.e. $\forall X_j \in \mathbf{X} \backslash \{MB(X_i) \cup \{X_i\}\}$, X_i is independent from X_j given $MB(X_i)$, $X_i \perp X_j|MB(X_i)$. If B_1 and B_2 are two BNs, both faithful to the same joint probability distribution, then $MB_{B_1}(X_i) = MB_{B_2}(X_i)$ for any variable X_i. MBs are not unique and may vary in size, but any given faithful BN has a unique $MB(X_i)$ for any X_i, which is the set of parents, children and parents of children of X_i.

In the paper only discrete BNs and faithful probability distributions are considered; furthermore these distributions are a very large class as proven in Meek (1995). Finally

we emphasise that we aren't looking for the BN as a causal model and we don't require that the edges represent causal impact.

Learning BN from data consists in finding the BN that best fits the available data. The algorithms for learning BN must deal with two different but related tasks: learning the structure (the DAG) and learning the parameters (the conditional probabilities). Methods for automatic induction of BN model generally fall into two different classes: methods based on the examination of conditional independence constraints that hold over the empirical probability distributions on the variables represented in the data (also called Constraint-methods), and search methods that seek to maximize some scoring function that describes the ability of the network to explain the observed data (also called Score-methods). We concentrate in the paper on the latter approach, which aims to find the highest scoring BN model and may produce more accurate results in structure learning that Constraint-methods (Cooper and Herskovits 1993; Acid and de Campos 2003). The Score-methods are typically based on defining (1) a scoring function for evaluating the quality of a given structure, and (2) a search procedure for traversing the space of candidate models. The scoring functions are based on different principles such as: entropy and information (Chow and Liu 1968; Herskovits and Cooper 1990), the minimum description length (Lam and Bacchus 1994; Bouckaert 1995; Friedman and Goldszmidt 1996) or Bayesian approaches (Buntine 1991; Cooper and Herskovits 1992; Heckerman et al. 1995) We focus on Bayesian approaches starting from a prior distribution on the possible networks and computing the posterior probability distribution conditioned to the data; the best network is the one that maximizes the posterior distribution. In the search context, K2 and BDe metrics, are the most common choices for the scoring function. Additionally a Bayesian score can prevent the model from over-fitting the data (Hartemink et al. 2002). We consider heuristic rather than exhaustive search strategies since the identification of the highest scoring model, for a given data set, is known to be NP-complete (Chickering 1996). Local heuristic search process is often used, which starts from an initial structure and repeatedly applies some local transformations (e.g. adding, deleting or reversing an edge). In the paper we concentrate on hill-climbing search procedure and results generated through its use. In learning BN no distinction is made between the classification node and other nodes, since it models and graphically represents the data.

2.1 Using the Bayesian network to reduce the complexity of the decision tree

As mentioned above, when there is a wide number of explanatory variables, the use of DT to describe a complex system often leads to not easily interpretable results because the complete tree is very large and may be sensitive to statistical irregularities. We would have a method to identify which variables are the most relevant for the analysis and then, build the DT only with these variables. We need to adopt a model representation so that the relevant features can be extracted and studied. For this task we recall that in a BN, under the faithfulness condition, the Markov blanket $MB(X_i)$ completely shields (d-separates) variable X_i from any other variable outside $MB(X_i) \cup \{X_i\}$. If we know the states of all the variables in the Markov blanket of X_i, other information about any other variable which is not in $MB(X_i) \cup \{X_i\}$ can't

modify our knowledge in X_i. We propose to reduce the complexity of the DT using only the variables which are in the Markov Blanket of the class. Several algorithms have been developed or proposed for identifying the Markov blanket (Margaritis and Thrun 1999; Frey et al. 2003; Tsamardinos et al. 2003) and the idea of using Markov blanket methods for feature selection is not new. For example, see references (Acid and de Campos 2003; Cowell et al. 1999; Frey et al. 2003) in Aliferis et al. (2003). In particular, a Markov blanket based variable selection algorithm, named HITON, has been presented (Aliferis et al. 2003): it has been applied in combination with DTs, but also with other common classifiers, on several massive databases and it has been compared with some state-of-the-art variable selection methods in terms of classification accuracy. With respect to Aliferis et al. (2003), this paper deals specifically with classification trees, focusing on the possibility of reducing tree size without significantly decreasing prediction accuracy.

Another method which can permit to identify the Markov blanket for the class is to directly read it from the BN. Recall that any given faithful BN has a unique $MB(X_i)$ for any X_i, which is the set of parents, children and parents of children of X_i, we learn the BN from the data and we select the variables which form the MB of the class. Of course it is very important to find a good BN: if we are confident about the BN learned from the data, we can assume that the joint probability distribution underlying by the BN is the real one which has generated the data and we can be confident that the identified variables in the Markov blanket of the class are really the most relevant for the classification task (Liu and Motoda 2008, Chap. 4). In Madden (2003), empirical results for classification are presented comparing BNs constructed using different learning approaches; it has been proved that BNs constructed by the Bayesian approach perform well in classification on benchmark databases, so we adopt this procedure for learning the BN.

With respect to this method, we compare the performance of identifying the MB for the class by using non-standard MB based variable selection algorithms: usually they are based on statistical independence tests and they don't deal with the Bayesian approach. By reading the MB directly from the BN, one can choose which approach to use. In this context this paper presents new results about the performance of HITON and Bayesian learning methods in comparison to standard feature selection algorithms and focusing in a specific classification task, i.e. the DT induction.

3 Databases and setting

We have learned the BN from data with different Bayesian approaches:

- Hill climbing search procedure, adding deleting and reversing edges. The search is not restricted by an order of the variables and the BN is chosen by maximising the BDe scoring function;
- K2 algorithm which is a hill climbing algorithm restricted by an order of the variables. We fix this order as the variables are shown in the database.

We then identify the Markov blanket for the class which will be consider as our feature subset.

We focus on HITON algorithm (Aliferis et al. 2003) as Markov blanket discovery algorithm. It first identifies the parents and children of the class C, then discovers the parents and children of the parents and children of C. This is a superset of MB(C). False positive are removed by a statistical independence test. It is proven that it returns the minimal variable set for predicting C and it exhibits best classification performance also when the sample size is limited. We choose HITON algorithm as it is developed to improve the performance of others Markov blanket discovery algorithms in literature. In our experiments we apply HITON with a G^2 statistical independence test, also called Likelihood Ratio Statistic, with significance level set to 0.05.

In order to compare these procedure, several methods for feature subset selection are used. We adopt a filter approach, which performs as a pre-processing step to learning and assesses the relevance of features by the properties of the data and by a relevance score, removing low scoring features. Afterward this subset of features is presented as input to the classification algorithm. These methods are computationally simple, fast and independent of the classification algorithm. We focus on Information Gain criteria (IG for short) which evaluates variables by measuring their gain ratio with respect to the data. We choose $\alpha = 0.05$ as threshold to select the variable subset.

Following the wrapper approach proposed in John et al. (1994), which implies that the selection algorithm searches for a good subset of features using induction algorithm itself as a part of the evaluation function, we use forward stepwise selection as heuristic search through the set of features and C4.5, both pruned and unpruned, as induction algorithm using 5-fold cross validation to evaluate performance.

Once the feature subsets are identified, we induce the corresponding DTs and we compare them in terms of their main characteristics and percentage of corrected classified instances. For DT induction, we focus on C4.5 algorithm (Quinlan 1993) because it has been shown that it provides a good classification accuracy and it is the fastest among the compared main-memory algorithms for machine learning and data mining. It is an heuristic algorithm where the process of tree derivation uses the gain ratio based on entropy as criterion of variable selection. DT is constructed from the training sample and potentially unnecessary sub-trees can be removed by pruning, improving its classification accuracy.

Similar results can be applied to other DT induction algorithms, for example to the frequently used CART-type. As proven in Schauerhuber et al. (2008), the C4.5 algorithm is superior to CART tree learner in terms of classification performance but it produces more complex trees. For this reason we decided to highlight the results of the C4.5 algorithm, nevertheless without ignoring the CART-type induction algorithm: it is more useful to reduce the complexity of a larger tree maintaining a good classification accuracy rather then the dimension of a not so complex tree which could also not be sensitive to the benefits of feature selection.

We select seven databases from the UCI repository of machine learning database (http://www.ics.uci.edu/~mlearn/MLRepository.html) and from the Department of Statistics, University of Munich (http://www.stat.uni-muenchen.de/service/datenarchiv/welcome_e.html). We choose only databases with discrete or categorical variables and with no missing values. The accuracy of each DTs is measured based on

Table 1 Characteristics of the databases used in the analysis

Database name	Num. of variables (without T)	Num. of instances
Auto	45	793
KRvsKP	36	3,196
Spect *	22	80
Credit	20	1,000
Lympho *	18	148
TicTacToe	9	958
Nursery	8	12,960

10 fold cross validation and this is repeated usually 10 times, except for particularly small databases for which the number of repetitions increases to 50 in order to reduce the variability. We compare the results of unpruned C4.5 DTs (U-DT for short) and C4.5 DTs for which we control the tree size in a pre-pruning way by fixing the minimum number of instances per leaf (U-DT-M for short) and in a post-pruning way by pruning and sub tree raising (P-DT for short). To complete the experimental study, we present also the CART-type tree results, for which we fix the minimum number of instances per node equal to 10.

In Table 1, we report the main characteristics of each database used in the analysis. The databases for which the number of split repetitions is 50 are marked by $*$.

All the experiments are performed with R (http://www.r-project.org). The open source implementation J4.8 for C4.5 became available recently in the user-friendly WEKA machine learning package (Witten and Frank 2005) and is accessible from within R by means of the RWeka package (Hornik et al. 2007). The implementation of CART tree learner is available in the R package tree (Ripley 2007).

4 Experimental results and discussion

In Tables 2, 3 and 4, for each database and for each type of C4.5 DT, we report the number of selected features corresponding to the methods of feature subset selection, the percentage of correctly classified instances and its standard deviation, the average number of leaves and tree size of the induced DT.

To reduce the tree size by post-pruning usually produces a significant tree complexity reduction and an increased percentage of correctly classified instances, but these improvements are not such that sub tree raising performs better of inducing a pre-pruning by relevant features subset selection. When a feature subset performs well with the unpruned DT, it also performs well with the both pre-and post-pruned DTs.

When there are a lot of variables, learning the BN and identifying from it the Markov blanket of the class leads to identify a feature subset which generates a DT with better performance, also in comparing it with Hiton, which is a specific algorithms for

Table 2 Experimental results (1): summary of the classification performance and model complexity across the databases

	Original	BNBDe	BNK2	HITON	IG	C4.5 U	C4.5 P
	45	2	2	10	4	5	5
Auto							
U-DT	31.60 ± 4.16 (107/140)	38.64 ± 3.72 (4/5)	38.64 ± 3.72 (4/5)	33.03 ± 4.95 (66/85)	29.85 ± 3.68 (63/74)	39.50 ± 4.02 (22/35)	41.13 ± 4.70 (19/31)
U-DT-M	36.35 ± 4.05 (12/18)	38.75 ± 3.66 (4/5)	38.75 ± 3.66 (4/5)	37.19 ± 3.86 (20/23)	35.17 ± 3.52 (20/23)	38.27 ± 3.73 (9/13)	40.09 ± 4.35 (8/11)
P-DT	34.11 ± 3.48 (65/88)	38.50 ± 3.47 (4/5)	38.50 ± 3.47 (4/5)	37.00 ± 3.68 (12/16)	36.56 ± 1.29 (7/9)	39.35 ± 4.43 (12/19)	41.71 ± 4.72 (13/21)
	Original	BNBDe	BNK2	HITON	IG	C4.5 U	C4.5 P
	36	13	10	20	3	4	4
KRvsKP							
U-DT	97.95 ± 0.63 (26/49)	97.24 ± 0.84 (14/25)	97.25 ± 0.84 (14/25)	97.06 ± 0.88 (22/41)	90.43 ± 1.51 (4/7)	94.09 ± 1.34 (5/9)	94.09 ± 1.34 (5/9)
U-DT-M	95.13 ± 1.46 (9/16)	95.16 ± 1.40 (9/16)	95.01 ± 1.57 (9/16)	95.13 ± 1.43 (9/16)	90.43 ± 1.51 (4/7)	94.09 ± 1.34 (5/9)	94.09 ± 1.34 (5/9)
P-DT	97.89 ± 0.68 (24/25)	97.25 ± 0.84 (14/25)	97.25 ± 0.84 (14/25)	97.25 ± 0.84 (14/25)	90.43 ± 1.51 (4/7)	94.09 ± 1.34 (5/9)	94.09 ± 1.34 (5/9)
	Original	BNBDe	BNK2	HITON	IG	C4.5 U	C4.5 P
	22	9	3	3	11	3	3
Spect							
U-DT	62.96 ± 39.61 (22/43)	73.68 ± 37.15 (14/27)	81.40 ± 32.57 (6/11)	74.88 ± 36.19 (4/7)	73.68 ± 36.17 (21/41)	81.40 ± 32.57 (6/11)	81.40 ± 32.57 (6/11)
U-DT-M	67.32 ± 39.64 (5/9)	78.56 ± 34.66 (4/7)	78.72 ± 34.64 (4/7)	71.00 ± 37.39 (2/3)	69.68 ± 38.55 (5/9)	78.72 ± 34.64 (4/7)	78.72 ± 34.64 (4/7)
P-DT	69.04 ± 38.30 (6/11)	76.52 ± 35.44 (4/7)	81.28 ± 32.78 (4/7)	74.88 ± 36.19 (3/5)	68.16 ± 38.51 (6/11)	81.28 ± 32.78 (4/7)	81.28 ± 32.78 (4/7)

Table 3 Experimental results (2): summary of the classification performance and model complexity across the databases

	Original	BNBDe	BNK2	HITON	IG	C4.5 U	C4.5 P
	20	4	3	13	1	6	2
Credit							
U-DT	71.60 ± 4.26 (91/109)	71.94 ± 3.54 (62/73)	73.85 ± 4.29 (16/21)	71.81 ± 4.16 (93/110)	68.79 ± 2.63 (1/1)	74.53 ± 3.84 (32/42)	71.70 ± 2.25 (8/10)
U-DT-M	72.72 ± 3.56 (14/18)	70.63 ± 2.96 (17/20)	71.83 ± 3.15 (12/15)	72.74 ± 3.56 (14/18)	68.79 ± 2.63 (1/1)	71.77 ± 3.28 (12/15)	71.63 ± 2.67 (5/6)
P-DT	72.63 ± 4.09 (31/38)	71.77 ± 3.65 (10/13)	73.14 ± 4.21 (16/21)	73.24 ± 4.16 (36/43)	70.00 ± 0.00 (1/1)	73.36 ± 3.75 (22/29)	71.89 ± 2.41 (8/10)
	Original	BNBDe	BNK2	HITON	IG	C4.5 U	C4.5 P
	18	15	3	14	15	3	3
Lympho							
U-DT	75.12 ± 24.74 (35/48)	78.99 ± 23.55 (35/48)	81.75 ± 22.86 (8/12)	75.09 ± 24.64 (40/53)	74.21 ± 25.15 (40/53)	82.40 ± 22.48 (7/11)	82.40 ± 22.48 (7/11)
U-DT-M	71.28 ± 25.20 (15/20)	71.31 ± 25.18 (15/20)	74.37 ± 25.34 (5/7)	71.31 ± 25.18 (15/20)	71.28 ± 25.20 (15/20)	74.37 ± 25.34 (5/7)	74.37 ± 25.34 (5/7)
P-DT	79.00 ± 23.69 (20/30)	79.59 ± 23.52 (20/30)	80.44 ± 23.39 (8/12)	79.32 ± 23.70 (15/23)	79.21 ± 23.68 (15/23)	82.40 ± 22.48 (7/11)	82.40 ± 22.48 (7/11)
	Original	BNBayes	BNK2	HITON	IG	C4.5 U	C4.5 P
	9	6	3	5	1	4	1
TicTacToe							
U-DT	76.07 ± 3.47 (41/61)	74.83 ± 3.49 (41/61)	72.96 ± 3.12 (11/16)	74.65 ± 3.16 (41/61)	69.94 ± 4.31 (3/4)	75.37 ± 3.63 (25/37)	69.94 ± 4.31 (3/4)
U-DT-M	69.07 ± 3.62 (7/10)	69.09 ± 3.62 (7/10)	68.80 ± 3.38 (7/10)	69.09 ± 3.62 (7/10)	69.94 ± 4.31 (3/4)	70.02 ± 3.49 (7/10)	69.94 ± 4.31 (3/4)
P-DT	76.71 ± 3.29 (23/34)	74.37 ± 3.35 (21/31)	72.99 ± 3.15 (11/16)	73.52 ± 2.80 (21/31)	69.94 ± 4.31 (3/4)	73.38 ± 3.15 (19/13)	69.94 ± 4.31 (3/4)

Table 4 Experimental results (3): summary of the classification performance and model complexity across the databases

	Original	BNBDe	BNK2	HITON	IG	C4.5 U	C4.5 P
	8	8	3	6	3	7	7
Nursery							
U-DT	95.12 ± 0.59 (167/255)	95.12 ± 0.59 (167/255)	89.21 ± 0.74 (21/29)	92.71 ± 0.68 (123/179)	89.21 ± 0.74 (21/29)	95.12 ± 0.57 (167/255)	95.12 ± 0.57 (167/255)
U-DT-M	91.52 ± 0.68 (49/71)	91.52 ± 0.68 (49/71)	89.21 ± 0.74 (21/29)	91.52 ± 0.68 (49/71)	89.21 ± 0.74 (21/29)	91.52 ± 0.68 (49/71)	91.52 ± 0.68 (49/71)
P-DT	95.16 ± 0.57 (167/255)	95.16 ± 0.57 (167/255)	89.21 ± 0.74 (21/29)	93.20 ± 0.67 (123/179)	89.21 ± 0.74 (21/29)	95.16 ± 0.56 (167/255)	95.16 ± 0.56 (167/255)

Table 5 CART results: summary of the classification performance and model complexity across the databases

	Original	BNBDe	BNK2	HITON	IG	C4.5 U	C4.5 P
Auto	39.72 ± 2.79 (9/17)	36.82 ± 3.04 (2/3)	36.82 ± 3.04 (2/3)	37.45 ± 2.98 (5/9)	38.08 ± 3.01 (4/7)	37.33 ± 2.98 (4/7)	36.82 ± 3.04 (2/3)
KRvsKP	97.72 ± 0.19 (14/27)	97.25 ± 0.17 (11/21)	97.25 ± 0.19 (10/19)	82.77 ± 0.17 (11/21)	90.42 ± 0.50 (4/7)	94.09 ± 0.38 (5/9)	94.09 ± 0.38 (5/9)
Spect	81.25 ± 0.90 (7/13)	78.75 ± 0.96 (5/9)	78.75 ± 1.01 (4/7)	72.50 ± 1.12 (3/5)	81.25 ± 0.90 (7/13)	78.75 ± 1.01 (4/7)	78.75 ± 1.01 (4/7)
Credit	75.60 ± 1.01 (8/15)	75.00 ± 1.06 (4/7)	72.60 ± 1.08 (3/5)	75.60 ± 1.01 (8/15)	70.00 ± 1.10 (2/3)	72.60 ± 1.08 (3/5)	71.70 ± 1.16 (3/5)
Lympho	89.19 ± 0.45 (14/27)	86.49 ± 0.61 (11/21)	82.43 ± 0.96 (6/11)	86.49 ± 0.61 (11/21)	89.19 ± 0.45 (14/27)	81.08 ± 0.98 (6/11)	81.08 ± 0.98 (6/11)
TicTacToe	82.46 ± 0.61 (12/23)	74.95 ± 0.80 (9/17)	69.94 ± 1.03 (4/7)	73.17 ± 0.85 (8/15)	69.94 ± 1.18 (2/3)	71.29 ± 1.05 (5/9)	71.29 ± 1.05 (5/9)
Nursery	85.88 ± 0.55 (10/19)	85.88 ± 0.55 (10/19)	85.88 ± 0.58 (9/17)	85.88 ± 0.55 (10/19)	85.88 ± 0.58 (9/17)	85.88 ± 0.55 (10/19)	85.88 ± 0.55 (10/19)

Markov blanket discovery. Hiton performs tests of conditional independence, which can be sensitive to statistical errors.

Otherwise, when the number of variables is small, usually less than 10, the complexity of the BN is not such that the MB(C) is sensitively different from the whole set of database variables and all of them are used for classification.

Using filter univariate feature selection methods, as Information Gain, leads to select a small number of relevant variables, and often the performance of the classifier decreases. A way to enlarge the number of variables could be done by decreasing the threshold, for instances to 0.01, but with the drawback of missing the variable space simplification.

In Table 5 we present the CART-type DT results, in order to show how the proposed method works with CART as well. For each database, we report the percentage of correctly classified instances and its standard deviation, the average number of leaves and the tree size of the induced DT corresponding to the methods of feature subset selection.

Learning DT models from complex systems data is a challenging task. For this reason, we want to extract the most relevant features and to use them in order to construct a DT with good classification performances. We use the notion of Markov blanket and we prove that, under some assumptions, if the BN is a good model (in terms of score associated to it) and therefore if we are confident in the model, we can use the Markov Blanket of the target to extract the features. The associated DT is a good classifier as our results can prove. Our proposal reduces the complexity of the DT so it has a simpler visualization and it can be more easily interpretable, making easier further statistical analysis. On the other hand, it maintains the good classification performance of the complete DT, the one in which all the variables are used.

References

Acid S, de Campos L (2003) Searching for Bayesian network structures in the space of restricted acyclic partially directed graphs. J Artif Intell Res 18:445–490

Aliferis CF, Tsamardinos I, Statnikov A (2003) HITON: a novel Markov Blanket Algorithm for optimal variable selection. In: Proceedings of the 2003 American Medical Informatics Association (AMIA) annual symposium, pp 21–25

Bouckaert RR (1995) Bayesian belief networks: from construction to inference. PhD Thesis, University of Utrecht

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth International Group, Belmont

Breslow LA, Aha DW (1997) Simplifying decision trees: a survey. Knowl Eng Rev 12(1):1–40

Buntine W (1991) Theory refinement on Bayesian networks. In: Proceeding of the seventh conference on uncertainty in artificial intelligence, pp 52–60

Chickering DM (1996) Learning Bayesian networks is NP-complete. In: Fisher D, Lenz HJ (eds) Learning from data: artificial intelligence and statistics. Springer, New York

Chow CK, Liu CN (1968) Approximating discrete probability distributions with dependence trees. IEEE Trans Inf Theory 14(3):462–467

Cooper G, Herskovits E (1992) A Bayesian method for the induction of probabilistic networks from data. Mach Learn 9:309–347

Cowell RG, Dawid AP, Lauritzen SL, Spiegelhalter DJ (1999) Probabilistic networks and expert systems. Springer, New York

Frey L, Fisher D, Tsamardinos I, Aliferis CF, Statnikov A (2003) Identifying Markov Blankets with decision tree induction. In: Proceedings of third IEEE international conference on data mining (ICDM), Melbourne, pp 59–66

Friedman N, Goldszmidt M (1996) Learning Bayesian networks with local structures. In: Proceedings of the twelfth conference on uncertainty in artificial intelligence, pp 252–262

Glymour C, Cooper GF (1999) Computation, causation and discovery. MIT Press, Cambridge

Hartemink AJ, Gifford DK, Jaakkola TS, Young RA (2002) Combining location and expression data for principled discovery of genetic regulatory network models. In: Pacific symposium on biocomputing, pp 437–449

Heckerman D, Geiger D, Chickering DM (1995) Learning Bayesian networks: the combinations of knowledge and statistical data. Mach Learn 20:197–243

Herskovits E, Cooper GF (1990) Kutato: an entropy-driven system for the construction of probabilistic expert systems from databases. In: Proceedings of the sixth conference on uncertainty in artificial intelligence, pp 54–62

Hornik K, Zeileis A, Hothorn T, Buchta C (2007) RWeka: an R Interface to Weka. R package version 0.3-2

Jensen FV (2001) Bayesian networks and decision graphs. Springer, New York

John GH, Kohavi R, Pleger K (1994) Irrelevant features and the subset selection problem. In: Proceedings of the eleventh international machine learning conference, pp 121–129

Lam W, Bacchus F (1994) Learning Bayesian belief networks. An approach based on the MDL principle. Comput Intell 10(4):269–293

Liu H, Motoda H (2008) Computational methods of feature selection. Chapman & Hall/CRC, Taylor and Francis Group LLC, London

Madden MG (2003) The performance of Bayesian network classifiers constructed using different techniques. In: Working notes of the ECML PkDD-03 Workshop, pp 59–70

Margaritis D, Thrun S (1999) Bayesian network induction via local neighborhoods. In: Solla S, Leen T, Müller KR (eds) Proceedings of conference on neural information processing systems (NIPS-12), MIT Press, Cambridge

Meek C (1995) Strong completeness and faithfulness in Bayesian networks. In: Proceedings of the eleventh conference on uncertainty in artificial intelligence, pp 403–410

Mitchell T (1997) Machine learning. Mc Graw-Hill, New York

Pearl J (1988) Probabilistic reasoning in intelligence systems. Morgan Kaufmann, Los Altos

Quinlan JR (1986) Induction of decision trees. Mach Learn 1:81–106

Quinlan JR (1993) C4.5: programs for machine learning. Morgan Kaufmann, Los Altos

Ripley B (2007) The tree package. R package version 1.0-26

Saeys Y, Inza I, Larrañaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23(19):2507–2517

Schauerhuber M, Zeileis A, Meyer D, Hornik K (2008) Benchmarking open-source tree learners in R/RWeka. Data analysis, machine learning and applications. In: Proceedings of the 31st annual conference of the Gesellschaft für Klassification

Tsamardinos I, Aliferis C, Statnikov A (2003) Algorithms for large scale markov blanket discovery. In: The sixteenth international flairs conference, St. Augustine, USA

Witten I, Frank E (2005) Data mining: practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco