Multi-dimensional Bayesian Network Classifiers

Linda C. van der Gaag

Peter R. de Waal

Department of Information and Computing Sciences, Utrecht University

Technical Report UU-CS-2006-056

www.cs.uu.nl

ISSN: 0924-3275

Multi-dimensional Bayesian Network Classifiers

Linda C. van der Gaag and Peter R. de Waal
Department of Information and Computing Sciences, Utrecht University
P.O. Box 80.089, 3508TB Utrecht, The Netherlands
{linda,waal}@cs.uu.nl

Abstract

We introduce the family of multi-dimensional Bayesian network classifiers which include one or more class variables and multiple feature variables. The family does not require that every feature variable is modelled as being dependent on every class variable, which results in better modelling capabilities than families of models with a single class variable. Yet, our family of multi-dimensional classifiers includes as special cases the well-known naive Bayesian and tree-augmented classifiers. We describe the learning problem for a subfamily of multi-dimensional classifiers and show that the complexity of the solution algorithm is polynomial in the number of variables involved. Preliminary experimental results illustrate the benefits of the multi-dimensionality of our classifiers.

1 Introduction

Bayesian network classifiers have gained considerable popularity for solving classification problems. Many real-life problems can be viewed as a classification problem, where an instance described by a number of features has to be classified in one of several distinct classes. The success of especially naive Bayesian classifiers and the more expressive tree-augmented network classifiers then is readily explained from their ease of construction from data and their generally good classification performance.

Many application domains, however, include classification problems where an instance has to be assigned to a most likely combination of classes. Since the number of class variables in a Bayesian network classifier is restricted to one, such classification problems cannot be modelled straightforwardly. One approach is to construct a compound class variable that models all possible combinations of classes. The class variable may then easily end up with an inhibitively large number of values. Moreover, the structure of the classification problem is not reflected in the model. Another approach is to develop multiple classifiers, one for each of the original classes. Multiple classifiers, however, cannot model interaction effects among the various classes and may not properly reflect the problem. Also, even if the classifiers indicate multiple classes, the implied combination of classes may not be the most likely explanation of the observed features.

We now introduce the concept of multi-dimensionality in Bayesian network classifiers to provide for modelling classification problems in which instances can be classified in multiple classes. A multi-dimensional Bayesian network classifier includes one or more class variables and one or more feature variables. It models the relationships between the variables by acyclic directed graphs over the class variables and over the feature variables separately, and further connects the two sets of variables by means of a bi-partite directed graph. An example

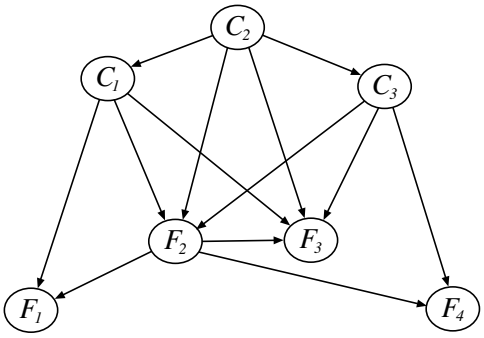

Figure 1: An example multi-dimensional Bayesian network classifier with class variables C_i and feature variables F_j.

multi-dimensional classifier is depicted in Figure 1. As for one-dimensional Bayesian network classifiers, we distinguish between different types of multi-dimensional classifier by imposing restrictions on their graphical structure.

For the subfamily of fully tree-augmented multi-dimensional classifiers, which have directed trees over their class variables as well as over their feature variables, we study the learning problem, that is, the problem of finding a classifier that best fits a set of available data. We show that the learning problem can be decomposed into optimisation problems for the set of class variables and for the set of feature variables separately, which can both be solved in polynomial time. Our learning algorithm assumes a fixed selection of feature variables per class, yet is easily combined with existing approaches to feature subset selection.

Preliminary experiments have hinted at the benefits of multi-dimensional classifiers. In a test on a small data set the multi-dimensional classifiers provided higher accuracy than their one-dimensional counterparts. In combination with feature selection they also led to classifiers with fewer parameters for the conditional probability tables. This is a desirable property for classifiers, as these parameters need to be estimated as part of the learning process.

The paper is organised as follows. In Section 2, we briefly review Bayesian network classifiers in general. In Section 3, we define the family of multi-dimensional classifiers and review the computational complexity of finding a joint value assignment of highest probability for the class variables of such a classifier. In Section 4, we address the learning problem for fully tree-augmented multi-dimensional classifiers and present polynomial algorithms for solving the problem. We report some preliminary results from an application in the biomedical domain in Section 5. The paper is rounded off with our concluding observations in Section 6.

2 Preliminaries

Before reviewing standard naive Bayesian and tree-augmented network classifiers, we introduce our notational conventions.

We consider Bayesian networks over a finite set $V = \{X_1, \ldots, X_k\}$, $k \geq 1$, of discrete random variables, where each X_i takes a value in a finite set $\mathit{Val}(X_i)$. For a subset of variables $Y \subseteq V$ we use $\mathit{Val}(Y) = \times_{X_i \in Y} \mathit{Val}(X_i)$ to denote the set of joint value assignments to Y. A Bayesian network now is a pair $B = \langle G, \Theta \rangle$, where G is an acyclic directed graph whose vertices correspond to the random variables and Θ is a set of parameters; the set Θ includes

a parameter $\theta_{x_i|\Pi x_i}$ for each value $x_i \in Val(X_i)$ and each value assignment $\Pi x_i \in Val(\Pi X_i)$ to the set ΠX_i of parents of X_i in G. The network B defines a joint probability distribution P_B over V given by

$$P_B(X_1, \ldots, X_k) = \prod_{i=1}^{k} \theta_{X_i|\Pi X_i} \tag{1}$$

Bayesian network classifiers are Bayesian networks of restricted topology that are tailored to solving classification problems in which instances described by a number of features have to be classified in one of several distinct predefined classes. The set of random variable V of a Bayesian network classifier is partitioned into a set $V_F = \{F_1, \ldots, F_m\}$, $m \geq 1$, of *feature variables* and a singleton set $\{C\}$ with the *class variable*. A *naive Bayesian classifier* has a directed tree for its graph G, in which the class variable C is the unique root and each feature variable F_i has C for its only parent. Since its graph is fixed, learning a naive Bayesian classifier amounts to establishing maximum-likelihood estimates from the available data for its parameters θ_C and $\theta_{F_i|C}$, $i = 1, \ldots, m$. A *TAN classifier* has for its graph G a directed acyclic graph in which the class variable C is the unique root and each feature variable F_i has C and at most one other feature variable for its parents; the subgraph induced by the set V_F, moreover, is a directed tree, termed the *feature tree* of the classifier. Learning a TAN classifier amounts to determining a graphical structure of maximum likelihood given the available data and establishing estimates for its parameters. For constructing a maximum-likelihood feature tree, an efficient algorithm is available from Friedman, Geiger and Goldszmidt (1997).

We would like to note that a classifier over $V_F \cup \{C\}$ in essence is a function $\mathcal{C}$: $Val(V_F) \rightarrow Val(C)$. Bayesian network classifiers typically build for this purpose upon the *winner-takes-all rule*. Using this rule, they associate with each value assignment $f \in Val(V_F)$, a class value c with $P_B(c \mid f) \geq P_B(c' \mid f)$ for all $c' \in Val(C)$, breaking ties at random.

3 Introducing multi-dimensional classifiers

Naive Bayesian and tree-augmented network classifiers include a single class variable and as such are one-dimensional. We now introduce the concept of multi-dimensionality in Bayesian network classifiers by defining a family of classifiers that may include multiple class variables.

A *multi-dimensional Bayesian network classifier* is a Bayesian network of which the graph $G = \langle V, A \rangle$ has the set V of random variables partitioned into the sets $V_C = \{C_1, \ldots, C_n\}$, $n \geq 1$, of class variables and the set $V_F = \{F_1, \ldots, F_m\}$, $m \geq 1$, of feature variables; the graph further has a restricted topology in which the set of arcs A is partitioned into the three sets A_C, A_F and A_{CF} with the following properties:

- the set $A_C \subseteq V_C \times V_C$ is composed of the arcs between the class variables, that is, the subgraph of G induced by V_C equals $G_C = \langle V_C, A_C \rangle$;
- the set $A_F \subseteq V_F \times V_F$ is composed of the arcs between the feature variables, that is, the subgraph of G induced by V_F equals $G_F = \langle V_F, A_F \rangle$;
- the set $A_{CF} \subseteq V_C \times V_F$ includes the arcs from the class variables to the feature variables such that for each $F_i \in V_F$ there is a $C_j \in V_C$ with $(C_j, F_i) \in A_{CF}$ and for each $C_i \in V_C$ there is an $F_j \in V_F$ with $(C_i, F_j) \in A_{CF}$.

The subgraph G_C of G is called the classifier's *class subgraph*; the subgraph G_F is called its *feature subgraph*. The subgraph $G_{CF} = \langle V, A_{CF} \rangle$ is called the *feature selection subgraph* of the classifier since it represents the selection of features that are deemed relevant for classification in view of the variables $C_1, \ldots, C_n$.

Within the family of multi-dimensional classifiers various different types of classifier can be distinguished based upon their graphical structures. An example is the *fully naive multi-dimensional classifier* in which both the class subgraph and the feature subgraph are empty. Note that this subfamily of bi-partite classifiers includes the one-dimensional naive Bayesian classifier as a special case. Further note that, reversely, any such bi-partite classifier has an equivalent naive Bayesian classifier with a single compound class variable. Another type of multi-dimensional classifier is the subfamily of classifiers in which both the class subgraph and the feature subgraph are directed trees. In the remainder of the paper, we will focus on this subfamily of *fully tree-augmented multi-dimensional classifiers*.

A multi-dimensional classifier in essence serves to find a joint value assignment of highest posterior probability for its set of class variables. Finding such an assignment given values for all feature variables involved, is equivalent to solving the MPA problem. This problem is known to be NP-hard in general, yet can be solved in polynomial time for networks of bounded treewidth (Bodlaender, Van den Eijkhof, and Van der Gaag, 2002). In the presence of unobserved feature variables, the problem of finding assignments of highest posterior probability remains intractable even for these restricted networks (Park, 2002). In view of the unfavourable computational complexity involved, we note that the practicability of multi-dimensional classifiers is limited to models with restricted class subgraphs, such as the fully naive and fully tree-augmented classifiers reviewed above.

4 Learning fully tree-augmented multi-dimensional classifiers

In this section we define the problem of learning a fully tree-augmented multi-dimensional classifier from data and show that this problem can be decomposed into two optimisation problems that have polynomial complexity.

4.1 Defining the learning problem

The general problem of learning a Bayesian network is to find a network B that matches the available set $D = \{u_1, \ldots, u_N\}$ of samples as closely as possible. In the most general setting, this optimisation problem is intractable. For restricted classes of networks, however, the problem becomes tractable, that is, solvable in polynomial time. Examples include the standard naive Bayesian and tree-augmented network classifiers and the one-dimensional classifiers of which the feature subgraph is a directed forest (Lucas, 2002). For other classes of networks, researchers have presented heuristics which provide good, though not optimal, solutions (Sahami, 1996; Keogh and Pazzani, 1999). In this section, we show that for the class of fully tree-augmented multi-dimensional classifiers, the learning problem is tractable.

Before formally defining the problem of learning a fully tree-augmented multi-dimensional classifier from data, we would like to note that the results mentioned above all relate to learning a network for a fixed set of relevant feature variables. To the best of our knowledge, the selection of an optimal set of features has not been solved as yet. We therefore formulate our learning problem as an optimisation problem for the subfamily of classifiers for which the

feature selection subgraph is fixed. We will return to the issue of feature subset selection in Section 4.3.

We begin by defining the subfamily of fully tree-augmented multi-dimensional classifiers with a fixed feature selection subgraph. The classifiers in this subfamily are considered *admissible* for the learning problem.

Definition 4.1 (Admissible Classifier). *Let $V = V_C \cup V_F$ with $V_C = \{C_1, \ldots, C_n\}$ and $V_F = \{F_1, \ldots, F_m\}$ be as before. Let $\underline{A}_{CF}$ be a given subset of $V_C \times V_F$ such that $\langle V, \underline{A}_{CF} \rangle$ is a feature selection subgraph for V. A multi-dimensional classifier B with graph $G = \langle V, A \rangle$ with $A = A_C \cup A_F \cup A_{CF}$ is called* admissible for $\underline{A}_{CF}$ *if*

- $G_C = \langle V_C, A_C \rangle$ *is a directed tree,*
- $G_F = \langle V_F, A_F \rangle$ *is a directed tree, and*
- $A_{CF} = \underline{A}_{CF}$.

The set of admissible multi-dimensional classifiers for $\underline{A}_{CF}$ is denoted as $\mathcal{B}_{\underline{A}_{CF}}$.

The learning problem now is to select from the set of admissible multi-dimensional classifiers, a model that best fits the available data. As a measure of how well a model describes the given set of samples, we propose to use the minimum description length (MDL) principle. The *MDL score of a network B given the data D* gives the amount of information needed to store the model and the data (Rissanen, 1978), and is defined as

$$MDL(B \mid D) = \frac{\log N}{2}|B| - LL(B \mid D). \tag{2}$$

The first term in the score captures the optimal number of bits to represent the network's parameters, where $|B|$ is used to denote its number of parameters. This term prohibits overfitting of the data, since it penalises networks that contain many parameters. In our approach this is taken care of by the structural requirements as formulated for the admissible classifiers in Definition 4.1.

The second term is the negation of the *log-likelihood* of the network B given the data D, which by the factorisation (1) equals

$$LL(B \mid D) = \sum_{i=1}^{N} \log \Big(P_B(u_i) \Big). \tag{3}$$

This term represents the optimal number of bits needed to represent the data D with the probability distribution P_B.

We now formally define the learning problem for fully tree-augmented multi-dimensional classifiers with a fixed feature selection subgraph.

Problem 4.2 (Learning Problem). *Let $V = V_C \cup V_F$ with $V_C = \{C_1, \ldots, C_n\}$ and $V_F = \{F_1, \ldots, F_m\}$ be as before. Let $\underline{A}_{CF} \subseteq V_C \times V_F$ be such that $\langle V, \underline{A}_{CF} \rangle$ is a feature selection subgraph for V. Let $\mathcal{B}_{\underline{A}_{CF}}$ be the set of admissible multi-dimensional classifiers for $\underline{A}_{CF}$. The learning problem is to find a classifier B in $\mathcal{B}_{\underline{A}_{CF}}$ that maximises the log-likelihood $LL(B \mid D)$.*

4.2 Solving the learning problem

In this subsection we show how the learning problem, defined as Problem 4.2, can be solved in polynomial time. For convenience we first extend the notation for parents of variables. For any variable X we let $\Pi_C X$ denote the class parents of X in G, i.e. $\Pi_C X = \Pi X \cap V_C$. We let $\Pi_F X$ denote the feature parents of X in G, i.e. $\Pi_F X = \Pi X \cap V_F$. Note that with this notation for any class variable C_i we have $\Pi_F C_i = \varnothing$ and $\Pi_C C_i = \Pi C_i$.

Assume that we have a set V_C of class variables and V_F of feature variables. Assume also that we have a fixed feature selection subgraph $(V_C \cup V_F, \underline{A}_{CF})$, so $\mathcal{B}_{\underline{A}_{CF}}$ is completely specified as in Definition 4.1.

We let $\hat{P}_D$ denote the *empirical distribution* as defined by the frequencies of occurrences in the data set D. It is well-known that, if the graph G of a network B is fixed, then the maximum likelihood estimators of the parameters in Θ, which by definition optimise the $LL(B|D)$ term in (2), are given by

$$\hat{\theta}_{x_i|\Pi x_i} = \hat{P}_D(x_i|\Pi x_i) \tag{4}$$

It can be shown (Friedman, Geiger and Goldszmidt, 1997) that by substitution of (4) into (3) and changing the order of summations the log-likelihood term can be rewritten as

$$\begin{aligned} & -N \sum_{i=1}^{n} H_{\hat{P}_D}(C_i|\Pi C_i) + N \sum_{j=1}^{m} H_{\hat{P}_D}(F_j|\Pi F_j) \\ & = N \sum_{i=1}^{n} I_{\hat{P}_D}(C_i;\Pi C_i) - N \sum_{i=1}^{n} H_{\hat{P}_D}(C_i) \\ & + N \sum_{j=1}^{n} I_{\hat{P}_D}(F_j;\Pi F_j) - N \sum_{j=1}^{m} H_{\hat{P}_D}(F_j). \end{aligned} \tag{5}$$

Here $H_P(X) = -\sum_x P(x) \log P(x)$ denotes the entropy of a random variable X with probability distribution P, $H_P(X|Y) = -\sum_{x,y} P(x,y) \log P(x|y)$ denotes the conditional entropy of X given Y, and

$$I_P(X;Y) = \sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

denotes the mutual information of X and Y.

The terms $H_{\hat{P}_D}(C_i)$ and $H_{\hat{P}_D}(F_j)$ on the right-hand side of (5) represent entropies of marginals of the empirical distribution. They depend only on the empirical distribution and not on the graph of the network. This implies that the admissible classifier that maximises the log-likelihood is the classifier in $\mathcal{B}_{\underline{A}_{CF}}$ that maximises the two sums of the mutual information terms. The mutual information $I_{\hat{P}_D}(P_j;\Pi F_j)$ can be rewritten further as follows. For any feature variable F_j, $j = 1, \ldots, m$, its parents are given by $\Pi F_j = \Pi_C F_j \cup \Pi_F F_j$. Using the chain law for mutual information (Cover and Thomas, 1991)) the term $I_{\hat{P}_D}(F_j;\Pi F_j)$ can now be rewritten as

$$\sum_{j=1}^{m} \left\{ I_{\hat{P}_D}(F_j;\Pi_C F_j) + I_{\hat{P}_D}(F_j;\Pi_F F_j|\Pi_C F_j) \right\} \tag{6}$$

In this expression $I_P(X;Y|Z)$ is the conditional mutual information of X and Y, given Z:

$$I_P(X;Y|Z) = \sum_{x,y,z} P(x,y,z) \log \frac{P(x,y|z)}{P(x|z)P(y|z)}$$

Since $\underline{A}_{CF}$ is fixed, the class parent set $\Pi_C F_j$ for feature F_j is the same for every admissible classifier. Thus the conditional mutual information $I_{\hat{P}_D}(F_j;\Pi_C F_j|\Pi_F F_j)$ is the same for all classifiers in $\mathcal{B}_{\underline{A}_{CF}}$. We can summarise the results of the above derivation in the following lemma.

Lemma 4.3. *The admissible classifier in $\mathcal{B}_{\underline{A}_{CF}}$ that solves Problem 4.2 is the classifier that maximises*

$$\sum_{i=1}^{n} I_{\hat{P}_D}(C_i;\Pi C_i) + \sum_{j=1}^{m} I_{\hat{P}_D}(F_j;\Pi_F F_j|\Pi_C F_j) \tag{7}$$

We now proceed by showing that the learning problem can be solved as two separable optimisation problems. Since class variables only have parents in V_C, the first term in (7) depends only on the arc set A_C. The second term in (7) depends on $\underline{A}_{CF}$, which is fixed, and on A_F. This implies that the two terms can be maximised separately. The maximisation of the first term was solved in Chow and Liu (1968). The solution is obtained as follows.

1. Construct a complete undirected graph by taking V_C as the set of vertices.
2. Let the weight of the edge between C_i and C_j, $i \neq j$, be given by $I_{\hat{P}_D}(C_i, C_j)$.
3. Build a maximum weighted spanning tree, for instance using the algorithm of Kruskal (1956).
4. Transform the undirected tree into a directed tree, by choosing an arbitrary vertex as root and setting all the arc directions outward from the root.

The second term in (7) depends on the structure of A_F and $\underline{A}_{CF}$. Since $\underline{A}_{CF}$ (and thus Π_C for F_j) is fixed, and the term is independent of A_C, this subproblem can be solved by finding the arc set A_F that makes (V_F, A_F) a directed spanning tree that maximises $\sum_j I_{\hat{P}_D}(F_j, \Pi_F F_j \mid \Pi_C F_j)$. This can be accomplished as follows.

1. Construct a complete directed graph on the set of vertices V_F.
2. Let the weight of the arc from F_i to F_j, $i \neq j$, be given by $I_{\hat{P}_D}(F_i; F_j|\Pi_C F_j)$.
3. Build a maximum weighted directed spanning tree, for instance using the algorithm of Chu and Liu (1968) or Edmonds (1967).

This spanning tree is the optimal arc set A_F. Note that for this subproblem we need to solve for a directed spanning tree, while for the first subproblem we need to compute an undirected spanning tree. This is due to $I_{\hat{P}_D}(C_i;C_j) = I_{\hat{P}_D}(C_j;C_i)$ and $I_{\hat{P}_D}(F_i;F_j|\Pi_C F_j) \neq I_{\hat{P}_D}(F_j;F_i|\Pi_C F_i)$.

We may now conclude that the learning problem for fully tree-augmented multidimensional classifiers is tractable.

Lemma 4.4. *The computational complexity of the solution to Problem 4.2 is polynomial in the number of variables.*

Proof. Solving Problem 4.2 amounts to computing an undirected maximum weighted spanning tree on the class variables and a directed maximum weighted spanning tree on the feature variables. The computation of the weight coefficients for the first subproblem has complexity of $O(n^2N)$, while the computation of the spanning tree has complexity of $O(n^2 \log n)$ according to Kruskal (1956). The computation of the weight coefficients for the second subproblem has complexity of $O(m^2N)$, and the computation of the directed spanning tree has complexity of $O(m^3)$. Since a typical data set will satisfy $N > \log n$ and $N > m$, this completes the proof. □

We conclude this subsection with the remark that we can formulate the learning problem also for classifiers in which we require that the arc set A_C or the arc set A_F is empty. The above described solution method can readily be adapted to this situation. If $A_C = \varnothing$, then we need only solve the second subproblem, i.e. maximise the second term in (7). For the case $A_F = \varnothing$, we need only solve the first subproblem.

4.3 Feature selection

We already mentioned in Section 4.1 that we defined the learning problem for a given feature selection subgraph. We conclude this section with some remarks on how the learning algorithm can be combined with a feature selection algorithm. We propose to use a feature selection algorithm with a wrapper approach based on accuracy (Kohavi and John, 1997)). Typically in such an approach an algorithm similar to the following is used.

1. Choose an initial feature selection subgraph. Denote this as the current subgraph.
2. Generate a number of new subgraphs by applying small changes to the current subgraph.
3. Compute the accuracy of the best classifier — the one that solves Problem 4.2 — for each new subgraph.
4. Determine the best new subgraph, that is the new subgraph with the highest accuracy.
5. If the accuracy of the best new subgraph is higher than that of the current classifier, then denote the best new subgraph as the new current subgraph and go to Step 2. If not, then stop and propose the best classifier for the current subgraph as the overall best.

Two approaches are popular. The forward feature selection approach starts with an empty subgraph (i.e. no features selected) and changes to the current subgraph are obtained by addition of an edge. The backward feature selection approach starts with all features selected as relevant for all class variables and changes to the current subgraph are obtained by removal of an edge.

5 Numerical Results

In this section we present some preliminary numerical results to illustrate the benefits of multi-dimensional Bayesian network classifiers. Since the established benchmark data repository in Newman *et al.* (1998) does not provide data sets with multiple class variables, we generated a data set to test the learning algorithm.

size of data set: 100		
classifier type	*acc.*	*param.*
Compound naive	0.46	695
Multi-dim naive	0.54	136
Compound TAN	0.35	1869
Multi-dim FTAN	0.41	740
size of data set: 200		
classifier type	*acc.*	*param.*
Compound naive	0.420	661
Multi-dim naive	0.555	179
Compound TAN	0.305	3060
Multi-dim FTAN	0.475	1092
size of data set: 400		
classifier type	*acc.*	*param.*
Compound naive	0.550	732
Multi-dim naive	0.605	276
Compound TAN	0.505	4604
Multi-dim FTAN	0.585	386

Table 1: Numerical results for different classifier types on the Oesophagus data set.

The data set was generated from the Oesophagus network as described in van der Gaag *et al.* (2002), which is a network for diagnosis of Oesophageal cancer. We generated three data sets of 100, 200 and 400 samples, respectively, using logic sampling. From these samples we removed the values for all non-observable variables, except for three variables, which were used as class variables. In the original Oesophagus network these three variables can be considered as natural candidates for this role.

We ran the learning algorithm on these data sets for two classifier types, namely naive and fully tree augmented. The accuracy of the classifiers was calculated using tenfold cross-validation, combined with a forward feature selection wrapper approach. In the multidimensional case we defined accuracy as the proportion of cases where all class variables were simultaneously classified correctly. We compared the accuracy performance of the two types of classifiers using a compound (one-dimensional) class variable and using a three-dimensional class variable.

The results are presented in Table 1. The accuracy of the best learned classifier is given in the second column. The number in the third column gives the number of parameters (entries in the conditional probability tables) that needed to be estimated for the learned network with the highest accuracy.

From the table we may conclude that the two multi-dimensional classifiers outperform their compound counterparts on accuracy. The number of estimated parameters is considerably smaller for the multi-dimensional classifiers. This difference is particularly striking for the case of naive classifiers with a small data set, where the multi-dimensional classifier needs only one fifth of the number of parameters of the compound naive classifier. Considering the small size of the data set from which these parameters must be estimated, this is a

considerable advantage.

These preliminary results look promising, but further experiments are necessary to substantiate any claims about better performance.

6 Conclusion

In this paper we introduced a new class of Bayesian network classifiers that include one or more class variables and multiple feature variables. We introduced a learning problem for this class of networks and showed how this problem can be solved with an algorithm that has a complexity that is polynomial in the number of variables. Preliminary numerical results for this class look promising.

In the future we foresee to test the performance of multi-dimensional classifiers on other data sets and with other feature selection strategies. We also consider to investigate the performance of classifiers where the restrictions of the spanning tree on the class subgraph or on the feature subgraph are relaxed. Possible less restrictive alternatives include forests or k-dependence-like classifiers. Since the number of class variables is usually small, we may also consider more complex class subgraphs than trees (or variants thereof) without the complexity becoming too prohibitive.

References

[1] Bodlaender, H.L., van den Eijkhof, F. and L.C. van der Gaag (2002). On the complexity of the MPA problem in probabilistic networks. In: *Proceedings of the 15th European Conference on Artificial Intelligence*, IOS Press, Amsterdam, pp. 675–679.

[2] Chow, V.K. and C.N. Liu (1968). Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Info. Theory*, 14, pp: 462–467.

[3] Chu, Y.J. and T.H. Liu (1965). On the shortest arborescence of a directed graph. *Scientia Sinica*, 14, pp. 1396–1400.

[4] Cover, T.M. and J.A. Thomas (1991). *Elements of Information Theory*, Wiley, New York.

[5] Edmonds, J. (1967). Optimum branchings. *J. Research of the National Bureau of Standards*, 71B, pp. 233–240.

[6] Friedman, N. D. Geiger and M. Goldszmidt (1997). Bayesian network classifiers. *Machine Learning*, 29, pp. 131–163.

[7] Van der Gaag, L.C., S. Renooij, C.L.M. Witteman, B.M.P. Aleman and B.G. Taal (2002). Probabilities for a probabilistic network: a case study in oesophageal cancer. *Artificial Intelligence in Medicine*, 25, pp. 123–148.

[8] Keogh, E.J. and M.J. Pazzani (1999), Learning augmented Bayesian classifiers: a comparison of distribution-based and classification-based approaches. In: *Uncertainty 99, 7th Int. Workshop on AI and Statistics*, pp. 225–230.

[9] Kohavi, R. and G.H. John (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97, pp. 273–324.

[10] Kruskal, J.B. (1956). On the shortest spanning subtree of a graph and the travelling salesman problem. *Proc. Amer. Math. Soc.*, 7, pp. 48–50.

[11] Lucas, P.J.F. (2004). Restricted Bayesian network structure learning. In: G.A. Gámez, S. Moral and A. Salmerón (Eds.), *Advances in Bayesian Network, Studies in Fuzziness and Soft Computing*, Vol. 146, Springer-Verlag, Berlin, pp. 217–232.

[12] Newman, D.J. A Hettich, C.L. Blake, and C.J. Merz (1998). UCI Repository of machine learning databases http://www.ics.uci.edu/∼mlearn/MLRepository.html. University of California, Department of Information and Computer Science, Irvine.

[13] Park, J. (2002). MAP complexity results and approximation methods. In: *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, pp. 388–396.

[14] Rissanen, J. (1978). Modeling by shortest data description, *Automatica*, 14, pp. 465–471.

[15] Sahami, M. (1996). Learning limited dependence Bayesian classifiers. In: *KDD-96, Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, pp. 335–338.