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. These classifiers include one or more class variables and multiple feature variables, which need not be modelled as being dependent on every class variable. Our family of multi-dimensional classifiers includes as special cases the well-known naive Bayesian and tree-augmented classifiers, yet offers better modelling capabilities than families of models with a single class variable. 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. We further present some preliminary experimental results to illustrate the benefits of the multi-dimensionality of our classifiers.

1 Introduction

Bayesian network classifiers have gained considerable popularity for solving classification problems 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 is readily explained from their ease of construction 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 problems cannot be modelled straightforwardly. One approach is to construct a compound class variable that models all possible combinations of classes. This class variable then easily ends up with an inhibitively large number of values. Also, the structure of the problem is not properly reflected in the resulting model. Another approach is to develop multiple classifiers, one for each original class. Multiple classifiers, however, cannot capture interactions among the various classes and may thus also not properly reflect the problem. Moreover, if the various classifiers indicate multiple classes, then the implied combination may not be the most likely explanation of the observed features.

In this paper we introduce the concept of multi-dimensionality in Bayesian network classifiers to provide for accurately modelling problems where instances are assigned to 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 a bi-partite directed graph; an example 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. Fully tree-augmented multi-dimensional classifiers, for example, have directed trees over their class variables as well as over their feature variables.

For the family of fully tree-augmented multi-dimensional classifiers, we study the learning problem, that is, the problem of finding a classifier that best fits a set of available data. We show that, given a fixed selection of feature vari-

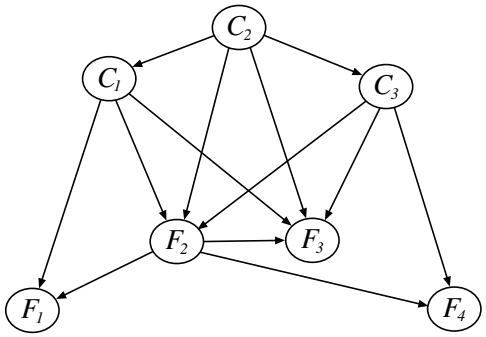

Figure 1: An example multi-dimensional Bayesian network classifier with class variables C_i and feature variables F_j.

ables per class variable, 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. We further argue that, although our learning algorithm assumes a fixed bi-partite graph between the class and feature variables, it is easily combined with existing approaches to feature subset selection.

The numerical results that we obtained from preliminary experiments with our learning algorithm clearly illustrate the benefits of multi-dimensionality of Bayesian network classifiers. Especially on smaller data sets, the constructed multi-dimensional classifiers provided higher accuracy than their one-dimensional counterparts. In combination with feature selection, moreover, our algorithm resulted in sparser classifiers with considerably fewer parameters and, hence, with smaller variance.

The paper is organised as follows. In Section 2, we review Bayesian network classifiers in general. In Section 3, we define our family of multi-dimensional classifiers. In Section 4, we address the learning problem for fully tree-augmented multi-dimensional classifiers and present a polynomial-time algorithm for solving it. In Section 5, we briefly address feature selection for our multi-dimensional classifiers. We report some preliminary results from an application in the biomedical domain in Section 6. The paper is rounded off with our concluding observations in Section 7.

2 Preliminaries

Before reviewing 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 $Val(X_i)$. For a subset of variables $Y \subseteq V$ we use $Val(Y) = \times_{X_i \in Y} Val(X_i)$ 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 V and Θ is a set of parameter probabilities; 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 now defines a joint probability distribution P_B over V that is factorised according to

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

Bayesian network classifiers are Bayesian networks of restricted topology that are tailored to solving classification problems where instances described by a number of features have to be classified in one of several distinct predefined classes (Friedman *et al.*, 1997). The set of variables 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 $V_C = \{C\}$ 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. A *tree-augmented network* (*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.

The general problem of learning a Bayesian network classifier from a given set of data samples $D = \{u_1, \ldots, u_N\}$, $N \geq 1$, is to find from among the family of network classifiers one that best matches the available data. As a measure

of how well a model describes the data, often the model's log-likelihood given the data is used; for a Bayesian network B and a data set D, the log-likelihood of B given D is defined as

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

For the family of Bayesian network classifiers in general, the learning problem is intractable. For various subfamilies of classifiers, however, the problem is solvable in polynomial time. Examples include the naive Bayesian and TAN classifiers reviewed above and the subfamily of Bayesian network classifiers of which the subgraph induced by the feature variables constitutes a forest (Lucas, 2004). For other subfamilies of Bayesian network classifiers, researchers have presented heuristic algorithms which provide good, though non-optimal, solutions (Sahami, 1996; Keogh and Pazzani, 1999). We would like to note that these results all relate to learning a classifier for a fixed set of relevant feature variables. To the best of our knowledge, the selection of an optimal set of feature variables has not been solved as yet.

Since its graph has a fixed topology, the problem of learning a naive Bayesian classifier amounts to just establishing maximum-likelihood estimates from the available data for its parameters. For a fixed graphical structure in general, the maximum-likelihood estimators of the parameter probabilities are given by

$$\hat{\theta}_{x_i|\Pi x_i} = \hat{P}_D(x_i \mid \Pi x_i),$$

where $\hat{P}_D$ denotes the empirical distribution defined by the frequencies of occurrence in the data. The problem of learning a TAN classifier amounts to first determining a graphical structure of maximum likelihood given the available data and then establishing estimates for its parameters. For constructing a maximum-likelihood feature tree, a polynomial-time algorithm is available from Friedman *et al.* (1997).

To conclude, we would like to note that a Bayesian network classifier computes for each instance a conditional probability distribution over the class variable. For classification purposes, it further is associated with a function $\mathcal{C}\colon \mathit{Val}(V_F) \rightarrow \mathit{Val}(C)$ which typically builds upon the *winner-takes-all rule*. Using this rule, a classifier B outputs for each value assignment $f \in \mathit{Val}(V_F)$, a class value c such that $P_B(c \mid f) \geq P_B(c' \mid f)$ for all $c' \in \mathit{Val}(C)$, breaking ties at random.

3 Multi-dimensional Classifiers

Bayesian network classifiers as reviewed above, 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 models 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 a restricted topology. The set V of random variables is 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 set of arcs A of the graph is partitioned into the three sets A_C, A_F and A_{CF} having the following properties:

- 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 of G that is induced by V_C equals $G_C = \langle V_C, A_C \rangle$;
- the subgraph of G that is induced by V_F equals $G_F = \langle V_F, A_F \rangle$.

The subgraph G_C is a graphical structure over the class variables and 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 a bi-partite graph that relates the various feature variables to the class variables; this subgraph is called the *feature selection subgraph* of the classifier and its set of arcs is termed the classifier's *feature selection arc set*. For any variable X in a multi-dimensional classifier, we now use $\Pi_C X$ to denote the class parents of X in G, that is, $\Pi_C X = \Pi X \cap V_C$. We further use $\Pi_F X$ to denote the feature parents of X in G. Note that for any class variable C_i we thus have that $\Pi_F C_i = \varnothing$ and $\Pi_C C_i = \Pi C_i$.

Within the family of multi-dimensional Bayesian network classifiers various different types of classifier are 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 have empty arc sets. Note that this subfamily of bi-partite classifiers includes the one-dimensional naive Bayesian classifier as a special case. 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 is used to find a joint value assignment of highest posterior probability to 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 *et al.*, 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.

4 Learning Fully Tree-augmented Multi-dimensional Classifiers

In this section we define the problem of learning a fully tree-augmented multi-dimensional classifier and show that this problem can be decomposed into two separate optimisation problems which can both be solved in polynomial time.

4.1 The learning problem

Before defining the problem of learning a fully tree-augmented multi-dimensional classifier from data, we recall that the related problem for Bayesian network classifiers has been studied for a fixed set of relevant feature variables. Following a similar approach, we now formulate our learning problem to pertain to a subfamily of classifiers for which the feature selection subgraph is fixed. We will return to the issue of feature subset selection in Section 5.

We begin by defining the subfamily of fully tree-augmented classifiers with a fixed selection of feature variables per class variable. These classifiers are considered *admissible* for the learning problem. We let the set of random variables V be partitioned into V_C and V_F; we further take a set of arcs A to be partitioned into A_C, A_F and A_{CF} as before. We now consider a subset $\underline{A}_{CF}$ of $V_C \times V_F$ such that $\langle V, \underline{A}_{CF} \rangle$ is a feature selection subgraph. A fully tree-augmented multi-dimensional classifier now is admissible for $\underline{A}_{CF}$ if we have for its set of arcs A_{CF} that $A_{CF} = \underline{A}_{CF}$. The set of all admissible classifiers for $\underline{A}_{CF}$ is denoted as $\mathcal{B}_{\underline{A}_{CF}}$.

The learning problem now is to find from among the set of admissible classifiers one that best fits the available data. As a measure of how well a model describes the data, we use its log-likelihood given the data. More formally, the learning problem for fully tree-augmented multi-dimensional classifiers with a fixed feature selection arc set $\underline{A}_{CF}$ then is to find a classifier B in $\mathcal{B}_{\underline{A}_{CF}}$ that maximises $LL(B \mid D)$.

4.2 Solving the learning problem

In this section we show that the learning problem for fully tree-augmented multi-dimensional classifiers can be solved in polynomial time.

We consider a fully tree-augmented classifier B with the class variables V_C and the feature variables V_F that is admissible for the feature selection arc set $\underline{A}_{CF}$. Building upon a result from Friedman *et al.* (1997), we have that the log-likelihood of B given a data set D can be written as

$$LL(B \mid D) =$$

$$= -N \cdot \sum_{i=1}^{n} H_{\hat{P}_D}(C_i | \Pi C_i) + N \cdot \sum_{j=1}^{m} H_{\hat{P}_D}(F_j | \Pi F_j)$$

$$= N \cdot \sum_{i=1}^{n} I_{\hat{P}_D}(C_i; \Pi C_i) - N \cdot \sum_{i=1}^{n} H_{\hat{P}_D}(C_i)$$
$$+ N \cdot \sum_{j=1}^{m} I_{\hat{P}_D}(F_j; \Pi F_j) - N \cdot \sum_{j=1}^{m} H_{\hat{P}_D}(F_j),$$

where $\hat{P}_D$ is the empirical distribution from D, $H_P(X) = -\sum_x P(x) \cdot \log P(x)$ is the entropy of a random variable X with distribution P, $H_P(X \mid Y) = -\sum_{x,y} P(x,y) \cdot \log P(x \mid y)$ denotes the conditional entropy of X given Y, and

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

denotes the mutual information of X and Y.

The two entropy terms $H_{\hat{P}_D}(C_i)$ and $H_{\hat{P}_D}(F_j)$ in the above expression for $LL(B \mid D)$ concern marginal distributions established from the available data. These terms therefore depend only on the empirical distribution and not on the graphical structure of the classifier. This observation implies that an admissible classifier that maximises the log-likelihood given the data is a classifier that maximises the sum of its two mutual-information terms.

We consider the mutual-information term $I_{\hat{P}_D}(F_j; \Pi F_j)$ in some more detail. We note that the set of parents ΠF_j of any feature variable F_j is partitioned into the set $\Pi_C F_j$ of class parents and the set $\Pi_F F_j$ of feature parents. Using the chain rule for mutual information (Cover and Thomas, 1991), the term $I_{\hat{P}_D}(F_j; \Pi F_j)$ can now be written as

$$\sum_{j=1}^{m} \Big(I_{\hat{P}_D}(F_j; \Pi_C F_j) + I_{\hat{P}_D}(F_j; \Pi_F F_j \mid \Pi_C F_j) \Big),$$

where $I_P(X;Y \mid Z)$ is the conditional mutual information of X and Y given Z with

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

Since the feature selection arc set $\underline{A}_{CF}$ is fixed, the set $\Pi_C F_j$ of class parents of the feature variable F_j is the same for every admissible classifier. We conclude that the mutual-information term $I_{\hat{P}_D}(F_j; \Pi_C F_j)$ is the same for all models in the set of admissible classifiers $\mathcal{B}_{\underline{A}_{CF}}$.

To summarise the above considerations, we have that a classifier that solves the learning problem for fully tree-augmented multi-dimensional classifiers with the fixed feature selection arc set $\underline{A}_{CF}$, is a classifier from $\mathcal{B}_{\underline{A}_{CF}}$ that maximises

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

We now proceed by showing that the learning problem can be decomposed into two separate optimisation problems which can both be solved in polynomial time. To this end, we first consider the mutual-information term pertaining to the class variables. We observe that, since class variables only have class parents, this term depends on the set of arcs A_C of the class subgraph only. With respect to the conditional mutual-information term above, we observe that this term depends on the feature selection arc set $\underline{A}_{CF}$, which is fixed, and on A_F, but not on the set A_C. These considerations imply that the two terms are independent and can be maximised separately.

The mutual-information term pertaining to the class variables can be maximised by using the procedure from Chow and Liu (1968) for constructing maximum-likelihood trees:

1. Construct a complete undirected graph over the set of class variables V_C.
2. Assign a weight $I_{\hat{P}_D}(C_i, C_j)$ to each edge between C_i and C_j, $i \neq j$.
3. Build a maximum-weighted spanning tree, for example using Kruskal's algorithm (1956).
4. Transform the undirected tree into a directed one, by choosing an arbitrary variable for its root and setting all arc directions from the root outward.

For the conditional mutual-information term pertaining to the feature variables, we observe that it is maximised by finding a maximum-likelihood directed spanning tree over the feature variables. Such a tree is constructed by the following procedure:

1. Construct a complete directed graph over the set of variables V_F.
2. Assign a weight $I_{\hat{P}_D}(F_i; F_j \mid \Pi_C F_j)$ to each arc from F_i to F_j, $i \neq j$.
3. Build a maximum-weighted directed spanning tree, for example using the algorithm of Chu and Liu (1968) or Edmonds' algorithm (1967).

We would like to note that for maximising the conditional mutual-information term for the feature variables, we have to construct a directed spanning tree, while for maximising the mutual-information term for the class variables we compute an undirected one. The need for this difference arises from the observation that $I_{\hat{P}_D}(C_i; C_j) = I_{\hat{P}_D}(C_j; C_i)$ for the class variables while $I_{\hat{P}_D}(F_i; F_j \mid \Pi_C F_j) \neq I_{\hat{P}_D}(F_j; F_i \mid \Pi_C F_i)$ for the feature variables.

Our algorithm for solving the learning problem for fully tree-augmented multi-dimensional classifiers with a fixed feature selection subgraph is composed of the two procedures described above. Solving the problem thus amounts to computing an undirected maximum-weighted spanning tree over the class variables and a directed maximum-weighted spanning tree over the feature variables. The computation of the weights for the undirected tree has a computational complexity of $O(n^2 N)$, while the construction of the tree itself requires $O(n^2 \log n)$ time (Kruskal, 1956). The computation of the weights for the directed tree has a complexity of $O(m^2 N)$, while the construction of the tree itself takes $O(m^3)$ time. Since a typical data set satisfies $N > \log n$ and $N > m$, we have that the overall complexity of our algorithm is polynomial in the number of variables involved.

We conclude this section by observing that the learning problem can also be formulated for classifiers in which the set A_C or the set A_F is empty. Our algorithm is readily adapted to these problems. With $A_C = \varnothing$, only the conditional mutual-information term for the feature variables has to be maximised using the second procedure above. With $A_F = \varnothing$, only the mutual-information term for the class variables has to be maximised using the first procedure.

5 Feature Subset Selection

In the previous section, we have addressed the learning problem for fully tree-augmented multi-dimensional classifiers with a fixed feature selection subgraph. We now briefly discuss feature subset selection for our classifiers.

It is well known that, if more or less redundant features are included in a data set, these features may bias the classifier that is learned from the data, which in turn may result in a relatively poor classification accuracy. By constructing the classifier over just a subset of the feature variables, a less complex classifier is yielded that tends to have a better performance (Langley *et al.*, 1992). Finding a minimum subset of features such that the selective classifier constructed over this subset has highest accuracy is known as the feature subset selection problem. The feature selection problem unfortunately is known to be NP-hard in general (Tsamardinos and Aliferis, 2003).

For Bayesian network classifiers, different heuristic approaches to feature subset selection have been proposed. One of these is the *wrapper approach* (Kohavi and John, 1997), in which the selection of feature variables is merged with the learning algorithm. We now argue that the same approach can be used for our multi-dimensional Bayesian network classifiers. The resulting procedure is as follows:

1. Choose the empty feature selection subgraph for the initial current subgraph.
2. From the current subgraph generate all possible feature selection subgraphs that are obtained by adding an arc from a class variable to a feature variable.
3. For each generated feature selection subgraph, compute the accuracy of the best classifier given this subgraph that is learned using the algorithm from the previous section.
4. Select the best generated subgraph, that is, the feature selection subgraph of the classifier of highest accuracy.

5. If the accuracy of the classifier with the selected subgraph is higher than that of the classifier with the current subgraph, then denote the selected subgraph as the current subgraph and go to Step 2. If not, then stop and propose the best classifier for the current subgraph as the overall best.

Starting with an empty graphical structure without any arcs, as in the above procedure, is known as *forward selection*. Alternatively, *backward elimination* can be used, which starts with a full graphical structure from which single arcs are removed in an iterative fashion.

6 Experimental Results

In this section we present some preliminary numerical results from our experiments to illustrate the benefits of multi-dimensionality of Bayesian network classifiers.

Since the UCI repository of benchmark data does not include any data sets with multiple class variables, we decided to generate some artificial data sets to test our learning algorithm. These data sets were generated from the oesophageal cancer network (Van der Gaag *et al.*, 2002). This network for the staging of cancer of the oesophagus includes 42 random variables of which 25 are observable feature variables. Three of the network's variables in essence are class variables, which in the current network are summarised in a single output variable. We generated three data sets of 100, 200 and 400 samples, respectively, using logic sampling. From the generated samples we removed the values of all non-observable variables, except for those of the three class variables.

From the three data sets we constructed fully naive and fully tree-augmented multi-dimensional classifiers. For this purpose, we used the learning algorithm described in the previous sections. For comparison purposes, we further learned naive and tree-augmented Bayesian network classifiers with a compound class variable from the data. For all classifiers, we used a forward-selection wrapper approach with the learning algorithm. The accuracies of the various classifiers were calculated using ten-fold cross-validation. For the multi-dimensional classifiers, we defined their accuracy as the proportion of samples that were classified correctly for all class variables involved.

size of data set: 100		
classifier type	*acc.*	*# par.*
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.*	*# par.*
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.*	*# par.*
Compound naive	0.550	732
Multi-dim naive	0.605	276
Compound TAN	0.505	4604
Multi-dim FTAN	0.585	386

Table 1: Experimental results for different types of classifier on data sets of different size generated from the oesophageal cancer network.

The results from our experiments are summarised in Table 1. The accuracy of the best learned classifier is given in the second column; the third column gives the number of parameter probabilities that were estimated for this classifier. From the table we may conclude that the multi-dimensional classifiers, without exception, outperform their compound counterparts in terms of accuracy. Also the numbers of estimated parameters are considerably smaller for the multi-dimensional classifiers. On average, the learned multi-dimensional classifiers require one-third of the number of parameters of their compound counterparts. The difference is particularly striking for the naive classifiers learned from the 100-sample data set, where the multi-dimensional classifier needs only one-fifth of the number of parameters of the compound one. The smaller numbers of parameters

required constitute a considerable advantage of the multi-dimensional classifiers over their compound counterparts since these parameters typically need to be estimated from relatively small data sets.

Although our preliminary experimental results look promising, we are aware that further experimentation is necessary to substantiate any claims about better performance of our multi-dimensional Bayesian network classifiers.

7 Conclusions and Future Research

In this paper we introduced a new family of Bayesian network classifiers that include one or more class variables and multiple feature variables that need not be modelled as being dependent upon every class variable. We formulated the learning problem for this family and presented a solution algorithm that is polynomial in the number of variables involved. Our preliminary experimental results served to illustrate the benefits of the multi-dimensionality of our Bayesian network classifiers.

In the near future we intend to perform a more extensive experimentation study of our learning algorithm, using other data sets and other approaches to feature subset selection. We further wish to investigate other types of model from our family of multi-dimensional classifiers. Possible alternatives include classifiers with k-dependence polytrees over their feature variables. Since the number of class variables usually is rather small, we also would like to investigate the feasibility of classifiers with slightly more complex class subgraphs.

References

H.L. Bodlaender, F. van den Eijkhof 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, pages 675 – 679.

V.K. Chow and C.N. Liu. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14: 462 – 467.

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

T.M. Cover and J.A. Thomas. (1991). *Elements of Information Theory*, Wiley.

J. Edmonds. (1967). Optimum branchings. *Journal of Research of the National Bureau of Standards*, 71B: 233 – 240.

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

E.J. Keogh and M.J. Pazzani. (1999). Learning augmented Bayesian classifiers: a comparison of distribution-based and classification-based approaches. In: *Proceedings of the 7th International Workshop on AI and Statistics*, pages 225 – 230.

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

J.B. Kruskal. (1956). On the shortest spanning subtree of a graph and the travelling salesman problem. *Proceedings of the American Mathematical Society*, 7: 48 – 50.

P. Langley, W. Iba and K. Thompson. (1992). An analysis of Bayesian classifiers. In: *Proceedings of the 10th Conference on Artificial Intelligence*, pages 223 – 228.

P.J.F. Lucas. (2004). Restricted Bayesian network structure learning. In: *Advances in Bayesian Networks. Studies in Fuzziness and Soft Computing*, vol. 146, Springer-Verlag, pages 217–232.

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

M. Sahami. (1996). Learning limited dependence Bayesian classifiers. In: *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, AAAI Press, pages 335 – 338.

I. Tsamardinos and C.F. Aliferis. (2003). Towards principled feature selection: relevance, filters, and wrappers. In: *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics.*

L.C. van der Gaag, 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: 123 – 148.