

Selective Augmented Bayesian Network Classifiers Based on Rough Set Theory

Zhihai Wang¹ Geoffrey I. Webb² Fei Zheng²

¹ School of Computer and Information Technology,
Beijing Jiaotong University, Beijing, 100044, China
`zhhwang@center.njtu.edu.cn`

² School of Computer Science and Software Engineering,
Monash University, Clayton, Victoria, 3800, Australia
`{webb, feizheng}@mail.csse.monash.edu.au`

Abstract. The naive Bayes classifier is widely used in interactive applications due to its computational efficiency, direct theoretical base, and competitive accuracy. However, its attribute independence assumption can result in sub-optimal accuracy. A number of techniques have explored simple relaxations of the attribute independence assumption in order to increase accuracy. *TAN* is a state-of-the-art extension of naive Bayes, that can express limited forms of inter-dependence among attributes. Rough sets theory provides tools for expressing inexact or partial dependencies within dataset. In this paper, we present a variant of *TAN* using rough sets theory and compare their tree classifier structures, which can be thought of as a selective restricted trees Bayesian classifier. It delivers lower error than both pre-existing *TAN*-based classifiers, with substantially less computation than is required by the *SuperParent* approach.

Keywords: Naive Bayes, Bayesian Network, Machine Learning,

1 Introduction

A classification task in data mining is to build a classifier which can assign a suitable class label to an unlabelled instance described by a set of attributes. Many approaches and techniques have been developed to create a classification model. The naive Bayesian classifier is one of the most widely used in interactive applications due to its computational efficiency, competitive accuracy, direct theoretical base, and its ability to integrate the prior information with data sample information [1, 7, 3, 5, 18, 15, 14]. It is based on Bayes' theorem and an assumption that all attributes are mutually independent within each class. Assume X is a finite set of instances, and $A = \{A_1, A_2, \dots, A_n\}$ is a finite set of n attributes. An instance $x \in X$ is described by a vector $\langle a_1, a_2, \dots, a_n \rangle$, where a_i is a value of attribute A_i . C is called the class attribute. Prediction accuracy will be maximized if the predicted class

$$L(x) = \operatorname{argmax}_c (P(c | \langle a_1, a_2, \dots, a_n \rangle)). \quad (1)$$

Unfortunately, unless the vector occurs many times within X , it will not be possible to directly estimate $P(c | \langle a_1, a_2, \dots, a_n \rangle)$ from the frequency with which

each class $c \in C$ co-occurs with $\langle a_1, a_2, \dots, a_n \rangle$ within the training instances. Bayes' theorem provides an equality that might be used to help estimate the posterior probability $P(c_i | x)$ in such a circumstance:

$$P(c_i | x) = \alpha \cdot P(c_i) \cdot P(\langle a_1, a_2, \dots, a_n \rangle | c_i) \quad (2)$$

where $P(c_i)$ is the prior probability of class $c_i \in C$, $P(\langle a_1, a_2, \dots, a_n \rangle | c_i)$ is the conditional probability of $x \in T$ given the class c_i , and α is a normalization factor. According to the chain rule, equation 2 can be written as:

$$P(c_i | x) = \alpha \cdot P(c_i) \cdot \prod_{k=1}^n P(a_k | a_1, a_2, \dots, a_{k-1}, c_i) \quad (3)$$

Therefore, an approach to Bayesian estimation is to seek to estimate each $P(a_k | a_1, a_2, \dots, a_{k-1}, c_i)$.

If the n attributes are mutually independent within each class value, then the conditional probability can be calculated in the following way:

$$P(\langle a_1, a_2, \dots, a_n \rangle | c_i) = \prod_{k=1}^n P(a_k | c_i). \quad (4)$$

Classification selecting the most probable class as estimated using formulae 2 and 4 is the well-known naive Bayesian classifier.

2 Approaches of Improving Naive Bayesian Method

In real world problems, the performance of a naive Bayesian classifier is dominated by two explicit assumptions: the attribute independence assumption and the probability estimation assumption. Of numerous proposals to improve the accuracy of naive Bayesian classifiers by weakening its attribute independence assumption, both Tree Augmented Naive Bayes (*TAN*) [4, 3, 5] and Lazy Bayesian Rules (*LBR*) [18] have demonstrated remarkable error performance [14]. Friedman, Geiger and Goldszmidt presented a compromise representation, called tree-augmented naive Bayes (*TAN*, simply called the basic *TAN*), in which the class node directly points to all attributes' nodes and an attribute node can have only at most one additional parent to the class node. Keogh & Pazzani took a different approach to constructing tree-augmented Bayesian networks [5] (simply called *SuperParent* or *SP*). The two methods mainly differ in two aspects. One is the criterion of attribute selection used to select dependence relations among the attributes while building a tree-augmented Bayesian network. Another is the structure of the classifiers. The first one always tends to construct a tree including all attributes, the second one always tends to construct a tree with fewer dependence relations among attributes and better classification accuracy. Zheng and Webb proposed a lazy Bayesian rule (*LBR*) learning technique [18]. *LBR* can be thought of as applying lazy learning techniques to naive Bayesian

rule induction. Both *LBR* and *TAN* can be viewed as variants of naive Bayes that relax the attribute independence assumption [14].

In this paper, however, we concentrate on the eager strategy, which holds a computational advantage when a single model can be applied to classify many instances. First of all, we mainly analyze the implementations of two different *TAN* classifiers and their tree classifier structures, and experimentally show how different dependence relations impact on the accuracy of *TAN* classifiers. Second, based on the definition of dependence in the basic rough set theory, we propose a definition of dependence measurement given the class variable, and use it to build a new dependence relation matrix. We believe that the directions of dependence relations are very important for performance of a classifier. Using this kind of definition, we can actually gain a directed-graph description. Third, we present a new algorithm for building selective augmented Bayesian network classifiers, which reduce error relative to the *TAN* classifiers, and has similar computational overheads. Experimental results also show that can deliver some improvements on performance, while requiring substantially less computation.

3 Some Issues in the Implementations

Now, we discuss some extended issues in the implementations of the *TAN* classifiers. First of all, the problem is related to the probability estimation assumption. In the basic *TAN*, for each attribute we assess the conditional probability given the class variable and another attribute. This means that the number of instances used to estimate the conditional probability is reduced as it is estimated from the instances that share three specific values (the class value, the parent value and the child value). Thus it is not surprising to encounter unreliable estimates, especially in small datasets. Friedman, Geiger and Goldszmidt dealt with this problem by introducing a smoothing operation [3]. There is a problem with this strategy when attribute value a does not occur in the training data (this situation can occur during cross validation testing), the value of the estimate will be zero. In our implementation, we use both these smoothing adjustments to estimate any conditional probability with $|\pi(a)| = 2$, i.e.,

$$\hat{P}(a | \pi(a)) = \frac{\text{counts}(a, \pi(a)) + N^0 \cdot \frac{\text{counts}(a)+1}{|T|+|A|}}{\text{counts}(\pi(a)) + N^0}. \quad (5)$$

where $|A|$ is the number of values for attribute A . We use Laplace estimation to estimate any other probability. In Keogh and Pazzani's *SuperParent* algorithm, they replace zero probabilities with a small epsilon (0.0001). Kohavi, Becker and Sommerfield [8] have shown that different methods for estimating the base probabilities in naive Bayes can greatly impact upon its performance. Similarly, different estimation methods will gain different effects on the same *TAN* classification model. We think that estimation methods should depend on the distribution of variables, as well as the number of training instances to support these kinds of estimation methods. Estimation methods should be independent of classification models for the same probability. In order to compare

the performances of classification models, we use the same method to estimate probabilities. In all the algorithms mentioned in our experiments, we always use formula 5 to estimate any conditional probability with $|\pi(a)| = 2$ and only standard Laplace estimation applied to any other probability.

Secondly, regarding the problem of missing values, in both the basic *TAN* classifiers and *SuperParent* classifiers, instances with missing values were deleted from the set of training instances. We keep all the instances, but ignore missing values from the counts for missing attribute values. Also, when we estimate a conditional probability $P(a_k | c_i)$, for a prior probability of class value c_i we exclude the occurrences of class value c_i with missing values on attribute A_k . Obviously, this makes the estimation of the condition more reliable while estimating any conditional probability.

Thirdly, although the choice of root variable does not change the log-likelihood of the basic *TAN* network, we have to set the direction of all edges for classification. When each edge (A_i, A_j) is added to the current tree structure, we always set the direction from A_i to A_j ($i < j$) at once. In Keogh and Pazzani's *SuperParent* classifiers, the direction of an arc is always from the super parent node to the favorite child node. That means that when a dependence relation is singled out, it always has the specific direction.

4 Selective Augmented Bayesian Classifiers

There are further factors that influence the performance of an augmented naive Bayesian classifier. The first one is the criterion for selecting dependence relations among the attributes. The second one is the criterion of terminating the selection. In this section, we describe our new algorithm for selective augmented Bayesian classifiers, explain how it works and is different from the basic *TAN* classifiers and *SuperParent* classifiers, and experimentally show its better performance and preferable computational profile.

4.1 Dependence Relation Matrix Based on Rough Set Theory

Friedman, Geiger and Goldszmidt explain why they use the conditional mutual information as the criterion of selecting dependence relations among the attributes [3, 4]. One problem with this criterion, as mentioned above, is how to decide the directions of dependence relations. Keogh and Pazzani use leave-one-out cross-validation to handle this problem in the process of building a classifier [5, 6]. When the best super parent and its favorite child are found, the dependence relation between them is naturally from the best super parent to its favorite child, i.e., the favorite child depends on the corresponding best super parent. For each arc to be added into the structure, *SuperParent* needs many times to execute leave-one-out cross-validations on the whole set of training instances. Although they proposed some shortcuts to speed up the process of evaluating many classifiers, the algorithm is still very time consuming. In our algorithm, we will use a dependence measurement based on rough sets theory.

Rough set [11] provides, in particular, tools for expressing inexact or partial dependencies within dataset. Given a dataset of some description or measurements concerning available instances, rough set methods enable to extract dependence relations between corresponding attributes or variables. These dependence relations can be applied to inductive reasoning about new, so far unseen cases, in a way well understandable for the user. Above advantages, as well as very effective computational framework for extraction of the most interesting dependencies from real-life data, cause a rapid development of applications of rough sets to more and more scientific fields and practical tasks [13].

According to the values of class variable, we define a new dependence relation matrix, in which each item of conditional dependence relation, $D(A_i, A_j | C)$, can be described as follows:

$$D(A_i, A_j | C) = \sum_{A_i, A_j, Y_c} \frac{|POS_{A_i}^{Y_c}(A_j)|}{counts(Y_c)} \quad (6)$$

where $POS_{A_i}^{Y_c}(A_j)$ represents the positive region of attribute A_j relative to attribute A_i within the class value Y_c [11]. Using this kind of definition, we can actually gain a directed-graph description. Each item not only reflects the degree of the dependence between two attributes, but also tells us the direction of the dependence relation.

4.2 A New Selective Augmented Bayesian Algorithm

In an augmented Bayesian classifier, the second important issue is how to decide the candidate dependence set and when terminate the the selection. There are $n(n - 1)$ different dependence relations among n attributes. When there are $n - 1$ or no any edge with the conditional mutual information more than 0, a basic *TAN* structure will have $n - 1$ arcs. We also try to add $n - 1$ arcs to our augmented Bayesian classifier, but the candidate set is different from the basic *TAN*. Because the way of weighting a candidate arc and the way of setting the direction of an arc are different from the basic *TAN* structure, the candidate arc set is different.

Based on above discussions, we can describe our selective augmented Bayesian algorithm, simply called *Select* in all tables, as follows.

1. Compute the dependence relation matrix conditional mutual information using formula 6.
2. Select an arc using a near maximum branching directed arborescence algorithm [10], based on the dependence relation matrix.
3. Use leave-one-out cross-validation to evaluate the current set of arborescences to decide whether this arc should be added into the current structure or not, adding it only if doing so reduces cross-validation error.
4. Repeat the previous iteration $n - 1$ times, or until no more arcs can be tested.

An important difference from the basic *TAN* algorithm is that this algorithm tries to build a maximum directed branch or arborescence [10], not a maximum

undirected spanning tree. We believe that the direction of dependence relations is a critical issue to minimizing error. In the procedure of building an augmented Bayesian classifier the network is always a directed structure.

5 Experimental Results

There are thirty-two natural domains used in our experiments, shown in Table 1. Twenty-nine of these are totally drawn from the previous research paper [18]. The other three (*Satellite*, *segment*, and *Shuttle*) are larger datasets. “ $S\#$ ” means the number of instances. “ $C\#$ ” means the number of values of a class attribute. “ $A\#$ ” means the number of attributes, not including the class attribute. All the experiments were performed in the Weka system [17]. The error rate of each classification model on each domain is determined by running 10-fold cross-validation on a dual-processor 1.7Ghz Pentium 4 Linux computer with 2Gb RAM. We use the default discretization method “weka.filters.DiscretizeFilter” as the discretization method for continuous values, which is based on Fayyad and Irani’s method [2].

Table 1 also shows the error rates of naive Bayes classifier (*NB*), the basic *TAN* classifier, the *SuperParent* classifier (*SP*), and our selective augmented Bayesian classifier (*Select*) on each domain, respectively. The last row contains the mean error rates for each column. The best one for a given dataset is shown in bold text. It shows the selective augmented Bayesian classifier has the lowest mean error rate. Table 2 presents the WIN/LOSS/DRAW records for comparing with all other classifiers. This is a record of the number of data sets for which the nominated algorithm achieves lower, higher, and equal error to the comparison algorithm, measured to two decimal places. The table also includes the outcome of a two-tailed binomial sign test. This indicates the probability that the observed outcome or more extreme should occur by chance if wins and losses were equiprobable. The selective augmented Bayesian classifier demonstrates the best performance. In the thirty-two databases, there are nineteen databases which the selective augmented Bayesian classifier has higher classification accuracy than Naive Bayes, twenty-one databases than the basic *TAN* classifier, sixteen databases than the *SuperParent* classifier. It is remarkable that there are eighteen databases which the basic *TAN* classifier has higher classification accuracy than Naive Bayes, and twelve datasets worse than Naive Bayes.

This suggests that selective augmented Bayesian classifier has similar error to the *SuperParent* classifier, but the selective augmented Bayesian classifier is much more efficient than the *SuperParent* classifier. Table 3 shows the time of building each classifier. On the most of domains, the selective augmented Bayesian classifier is much faster than the *SuperParent* classifier.

Table 4 shows the number of arcs in the Bayesian network built by the *SuperParent* classifier (*SP*), the basic *TAN* classifier, and our (*Select*) on each domain, respectively. These experimental results show that the basic *TAN* always tends to construct a tree including all attributes, the *SP* always tends to construct a tree with fewer dependence relations among attributes and better

Table 1. Descriptions of Data and Average Error Rates

| | Domain | S# | C# | A# | M | <i>NB</i> | <i>SP</i> | <i>TAN</i> | <i>Select</i> |
|----|-----------------|-------|----|----|---|--------------|-------------|--------------|---------------|
| 1 | Anealing | 898 | 6 | 38 | Y | 5.46 | 3.3 | 4.34 | 4.12 |
| 2 | Audiology | 226 | 24 | 69 | Y | 29.20 | 27.88 | 27.43 | 26.55 |
| 3 | Breast Cancer | 699 | 2 | 9 | Y | 2.58 | 2.58 | 5.01 | 2.58 |
| 4 | Chess(kr-vs-kp) | 3169 | 2 | 39 | N | 12.36 | 5.19 | 6.54 | 8.01 |
| 5 | Credit | 69 | 2 | 15 | Y | 15.07 | 15.22 | 14.93 | 15.23 |
| 6 | Echocardiogram | 131 | 2 | 6 | Y | 27.48 | 29.01 | 35.88 | 28.24 |
| 7 | Glass | 214 | 7 | 9 | N | 41.12 | 41.59 | 37.85 | 35.98 |
| 8 | Heart | 270 | 2 | 13 | N | 15.19 | 16.30 | 20.74 | 17.04 |
| 9 | Hepatitis | 155 | 2 | 19 | Y | 16.13 | 16.13 | 11.61 | 14.84 |
| 10 | Horse Colic | 368 | 2 | 21 | Y | 20.11 | 19.29 | 19.84 | 19.02 |
| 11 | House Votes 84 | 435 | 2 | 16 | N | 9.89 | 6.67 | 7.59 | 9.20 |
| 12 | Hypothyroid | 3163 | 2 | 25 | Y | 2.94 | 2.81 | 2.66 | 2.75 |
| 13 | Iris | 150 | 3 | 4 | N | 6.00 | 6.67 | 5.33 | 8.00 |
| 14 | Labors | 57 | 2 | 16 | Y | 3.51 | 3.51 | 12.28 | 3.51 |
| 15 | LED | 1000 | 10 | 7 | N | 26.20 | 26.60 | 25.90 | 26.70 |
| 16 | Bupa | 345 | 2 | 6 | N | 36.81 | 38.84 | 37.68 | 39.13 |
| 17 | Lung Cancer | 32 | 3 | 56 | Y | 46.88 | 50.00 | 46.88 | 43.75 |
| 18 | Lymphography | 148 | 4 | 18 | N | 14.19 | 14.86 | 18.92 | 12.84 |
| 19 | PID | 768 | 2 | 8 | N | 25.00 | 25.52 | 25.00 | 24.74 |
| 20 | Post Operative | 90 | 3 | 8 | Y | 28.89 | 31.11 | 35.56 | 30.00 |
| 21 | Primary Tumor | 339 | 22 | 17 | Y | 48.97 | 51.15 | 54.28 | 50.15 |
| 22 | Promoters | 106 | 2 | 57 | N | 8.49 | 8.48 | 16.04 | 11.32 |
| 23 | Satellite | 6435 | 6 | 36 | N | 18.90 | 12.18 | 12.29 | 12.15 |
| 24 | Segment | 2310 | 7 | 19 | N | 11.08 | 7.01 | 6.15 | 5.54 |
| 25 | Shuttle | 58000 | 7 | 9 | N | 10.07 | 5.09 | 8.06 | 6.95 |
| 26 | Solarflare | 1389 | 3 | 10 | N | 3.89 | 1.08 | 1.08 | 1.87 |
| 27 | Sonar | 208 | 2 | 60 | N | 25.48 | 25.96 | 29.33 | 23.08 |
| 28 | Soybean | 683 | 19 | 35 | Y | 7.17 | 6.59 | 10.98 | 6.44 |
| 29 | Splice | 3177 | 3 | 60 | N | 4.66 | 4.50 | 4.60 | 4.41 |
| 30 | TTT | 958 | 2 | 9 | N | 29.54 | 27.45 | 26.10 | 29.54 |
| 31 | Wine | 178 | 3 | 13 | N | 3.37 | 3.37 | 3.93 | 2.81 |
| 32 | Zoology | 101 | 7 | 16 | N | 5.94 | 6.93 | 4.95 | 6.93 |
| | Error Mean | | | | | 17.58 | 16.93 | 18.11 | 16.67 |

classification accuracy. In order to show the candidate arc set and the structure are different from the basic *TAN* classifier, we give an example produced by the basic *TAN* algorithm, the *SuperParent* algorithm, and our selective augmented Bayesian algorithm on dataset *Soybean* respectively. The basic *TAN* algorithm produced a tree with $n - 1$ arcs ($n = 35$), shown in figure 1, where a node which parent is node 0 and has no any child is omitted for simplicity. This tree includes all the attribute nodes. The selective augmented Bayesian algorithm produced eight arcs in four branches, shown in figure 2. The structure of this Bayesian classifier is a forest, where some of arcs belong to the *TAN* structure,

Table 2. Comparison of *Select* to others

| | <i>WIN</i> | <i>LOSS</i> | <i>DRAW</i> | <i>P</i> |
|--------------------|------------|-------------|-------------|----------|
| <i>NaiveBayes</i> | 19 | 10 | 3 | 0.1360 |
| <i>SuperParent</i> | 16 | 11 | 5 | 0.4420 |
| <i>TAN</i> | 21 | 11 | 0 | 0.1102 |

Table 3. Time in CPU Seconds

| Domain | <i>SP</i> | <i>TAN</i> | <i>Select</i> |
|-------------------|-----------|------------|---------------|
| 1 Annealing | 110.31 | 0.17 | 13.95 |
| 2 Audiology | 530.96 | 1.79 | 126.38 |
| 3 Breast Cancer | 1.20 | 0.17 | 0.40 |
| 4 Chess | 662.98 | 1.14 | 61.81 |
| 5 Credit | 5.02 | 0.14 | 1.35 |
| 6 Echocardiogram | 0.07 | 0.03 | 0.10 |
| 7 Glass | 0.37 | 0.11 | 0.31 |
| 8 Heart | 0.31 | 0.11 | 0.25 |
| 9 Hepatitis | 2.13 | 0.11 | 1.59 |
| 10 Horse Colic | 6.78 | 0.13 | 1.42 |
| 11 House Votes | 1.51 | 0.10 | 1.50 |
| 12 Hypothyroid | 130.58 | 0.25 | 28.8 |
| 13 Iris | 0.07 | 0.02 | 0.07 |
| 14 Labor | 0.09 | 0.09 | 0.17 |
| 15 LED | 1.48 | 0.07 | 1.54 |
| 16 Bupa | 0.24 | 0.06 | 0.13 |
| 17 Lung Cancer | 7.23 | 0.14 | 5.56 |
| 18 Lymphography | 0.36 | 0.10 | 1.52 |
| 19 PID | 1.37 | 0.12 | 1.32 |
| 20 Post Operative | 0.10 | 0.02 | 0.11 |
| 21 Ptn | 2.63 | 0.11 | 5.22 |
| 22 Promoters | 11.75 | 0.34 | 10.77 |
| 23 Satellite | 6247.56 | 4.9 | 269.66 |
| 24 Segment | 202.69 | 1.84 | 24.36 |
| 25 Shuttle | 137.68 | 5.14 | 79.82 |
| 26 Solarflare | 8.85 | 0.07 | 2.56 |
| 27 Sonar | 165.06 | 1.88 | 22.62 |
| 28 Soybean | 51.59 | 0.43 | 51.63 |
| 29 Splice | 1090.46 | 3.71 | 313.44 |
| 30 TTT | 1.07 | 0.06 | 0.53 |
| 31 Wine | 0.17 | 0.13 | 0.48 |
| 32 Zoo | 1.80 | 0.09 | 0.61 |

Table 4. Arcs for each algorithm

| Domain | <i>SP</i> | <i>TAN</i> | <i>Select</i> |
|-------------------|-----------|------------|---------------|
| 1 Annealing | 9 | 27 | 12 |
| 2 Audiology | 6 | 61 | 11 |
| 3 Breast Cancer | 0 | 8 | 1 |
| 4 Chess | 8 | 35 | 17 |
| 5 Credit | 4 | 13 | 7 |
| 6 Echocardiogram | 0 | 5 | 17 |
| 7 Glass | 3 | 5 | 7 |
| 8 Heart | 0 | 10 | 3 |
| 9 Hepatitis | 2 | 18 | 3 |
| 10 Horse Colic | 4 | 18 | 7 |
| 11 House Votes | 1 | 15 | 3 |
| 12 Hypothyroid | 7 | 24 | 14 |
| 13 Iris | 0 | 3 | 1 |
| 14 Labor | 0 | 10 | 3 |
| 15 LED | 1 | 6 | 4 |
| 16 Bupa | 1 | 2 | 1 |
| 17 Lung Cancer | 2 | 19 | 6 |
| 18 Lymphography | 1 | 17 | 4 |
| 19 PID | 1 | 5 | 1 |
| 20 Post Operative | 2 | 7 | 3 |
| 21 Ptn | 0 | 16 | 5 |
| 22 Promoters | 0 | 56 | 4 |
| 23 Satellite | 35 | 35 | 35 |
| 24 Segment | 13 | 15 | 17 |
| 25 Shuttle | 2 | 6 | 8 |
| 26 Solarflare | 6 | 9 | 8 |
| 27 Sonar | 3 | 56 | 7 |
| 28 Soybean | 2 | 34 | 8 |
| 29 Splice | 2 | 59 | 7 |
| 30 TTT | 1 | 8 | 0 |
| 31 Wine | 0 | 12 | 2 |
| 31 Zoo | 2 | 15 | 3 |

but some others do not belong to the *TAN* structure. This example also shows our selective augmented Bayesian algorithm produces different results from the *sTAN* method described by Keogh and Pazzani [6]. In the *sTAN* model, they only select from those edges that appear in the basic *TAN* tree structure. The

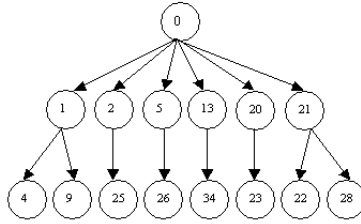


Fig. 1. The Bayesian network of *TAN* on dataset Soybean

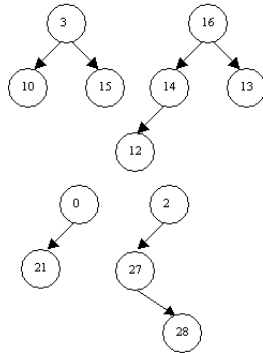


Fig. 2. The Bayesian network of selective augmented *TAN* on dataset Soybean

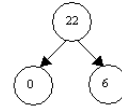


Fig. 3. The Bayesian network of super parent *TAN* on dataset Soybean

result of the *SuperParent* algorithm is shown in figure 3. The *SuperParent* algorithm only uses leave-one-out cross validation to determine the directions of arcs, but it can not always obtain better performance. For example, on dataset *PID*, the Bayesian networks built by both the *SuperParent* algorithm and the selective augmented Bayesian algorithm have only one arc, but the directions are different. The tree structure built by the *SuperParent* algorithm is only the arc: $\langle 4, 1 \rangle$, but the selective augmented Bayesian algorithm produces one arc with reverse direction.

6 Conclusion

In this paper we have investigated three issues that affect the quality of *TAN* classifiers learned from data. The first issue is how to estimate the base probabilities. We conclude that it is important to use a consistent estimation to compare with each other. The second is how to measure dependence relations between two attributes with directions. Based on the definition of dependence degree in the basic rough set theory, we propose a definition of dependence measurement given class variable for building a classification model. Thirdly, we mainly present a selective augmented Bayesian network classifier that reduces error relative to the

original *TAN*, and with similar computational overheads, but much lower computational overheads than the *SuperParent*, which is a state-of-the-art variant of the basic *TAN* classifier. Experimental results show that it can deliver some improvements on performance, while requiring substantially less computation.

References

1. P. Domingos and M. Pazzani.: Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. In: Proceedings of the Thirteenth ICML, Morgan Kaufmann Publishers, San Francisco, CA(1996)105–112,
2. U. Fayyad and K. Irani.: Multi-Interval Discretization of Continuous-valued Attributes for Classification Learning. In: Proceedings of the 13th IJCAI, Seattle, WA: Morgan Kaufmann Publishers, (1993), 1022-1027
3. N. Friedman, D. Geiger and M. Goldszmidt.: Bayesian Network Classifiers. Machine Learning, Kluwer Academic Publishers, Boston, 29 (1997) 131–163
4. N. Friedman and M. Goldszmidt: Building Classifiers Using Bayesian Networks. In: Proc. of the 13h National Conf. on AI, Menlo Park, 1996 (1277-1284)
5. E. J. Keogh and M. J. Pazzani.: Learning Augmented Bayesian Classifiers: A Comparison of Distribution-Based and Classification-Based Approaches. In: Proceedings of the 7th International Workshop on AI and Statistics. (1999) 225-230
6. E. J. Keogh and M. J. Pazzani.: Learning the Structure Augmented Bayesian Classifiers. Int. J. on AI Tools, World Scientific Publishing Company, 11 (2002), 587-601
7. R. Kohavi.: Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid, In: Simoudis, E., Han, J. W., Fayyad, U. M.(eds.): Proceedings of the 2nd International Conference on KDD, The AAAI Press, Menlo Park, (1996)202-207
8. R. Kohavi, B. Becker and D. Sommerfeld.: Improving Simple Bayes. In: Proceedings of the ECML, 1997
9. P. Langley and S. Sage.: Induction of Selective Bayesian Classifiers. In: Proceedings of the 10th Conf. on UAI, Morgan Kaufmann Publishers, Seattle, 1994 (339–406)
10. E. Minieka.: Optimization Algorithms for Networks and Graphs. New York: Marcel Dekker, Inc., (1978) 26-36
11. Z. Pawlak.: Rough Sets: Theoretical Aspects of Reasoning about Data, Dordrecht: Kluwer Academic Publishers, 1991
12. M. Sahami.: Learning Limited Dependence Bayesian Classifiers. In: Proceedings of the 2nd Int. Conf.on KDD, AAAI Press, Portland, OR, 1996, 335–338
13. D. Slezak.: Searching for Frequent Reduts in Decision Tables with Uncertain Objects. In: Proc.of the First Int.Conf.on RSCTC, Springer, 1998, 52-59
14. Wang, Z., Webb, G. I.: Comparison of Lazy Bayesian Rule and Tree-Augmented Bayesian Learning. In: Proc. of ICDM2002, IEEE Computer Society, (2002) 490-497
15. Webb, G. I., Boughton, J., Wang, Z: Averaged One-Dependence Estimators: Preliminary Results. In: Proc. of ADM 2002, Canberra, Australia: UTS, (2002) 65-73
16. Webb, G. I., Pazzani, M. J.: Adjusted Probability Naive Bayesian Induction. In: Proceedings of the 11th AJCAI, Springer-verlag, Brisbane, 1998 (285–295)
17. Witten, I. H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Seattle, WA: Morgan Kaufmann. (2000)
18. Zheng, Z., Webb, G. I.: Lazy Learning of Bayesian Rules. Machine Learning, Kluwer Academic Publishers, Boston, 41(2000) 53-84