Induction of hybrid decision tree based on post-discretization strategy*

WANG Limin** and YUAN Senmiao

(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

Received September 25, 2003; revised November 10, 2003

Abstract By redefining test selection measure, we propose in this paper a new algorithm, Flexible NBTree, which induces a hybrid of decision tree and Naive Bayes. Flexible NBTree mitigates the negative effect of information loss on test selection by applying post-discretization strategy: at each internal node in the tree, we first select the test which is the most useful for improving classification accuracy, then apply discretization of continuous tests. The finial decision tree nodes contain univariate splits as regular decision trees, but the leaves contain Naive Bayesian classifiers. To evaluate the performance of Flexible NBTree, we compare it with NBTree and C4.5, both applying pre-discretization of continuous attributes. Experimental results on a variety of natural domains indicate that the classification accuracy of Flexible NBTree is substantially improved.

Keywords: machine learning, hybrid decision tree, Naive Bayes.

Decision tree-based methods of supervised learning represent one of the most popular approaches within the AI field for dealing with classification problems. They have been widely used for years in many domains such as pattern recognition, data mining, signal processing, etc. But standard decision tree learning algorithms can handle discrete attributes only[1]. Learning decision tree from data consisting of continuous and discrete variables is a key issue in machine learning.

The decision tree learning algorithms proposed before commonly apply discretization of continuous attributes[2,3], then use the test selection measure for discrete attributes[1,4] to construct decision tree, thus continuous-valued predictive attributes can be incorporated into the learned tree. The effectiveness of pre-discretization has been proved in practice. But from the viewpoint of information theory, the information loss caused by pre-discretization may affect test selection, then in turn degrade the classification accuracy to some extent.

Naive Bayes is known to be optimal if predictive attributes are independent given the class. Although the conditional independence assumption is rarely valid in practical learning problems, experiments on real world data have repeatedly shown it to be competitive with much more sophisticated induction algorithms[5]. Since the leaves of decision tree consist of very few instances, we suppose that the distribution of those instances approximately satisfies the conditional independence assumption. If the leaves are replaced by Naive Bayes, the advantages of both decision tree (i.e. segmentation) and Naive Bayes (evidence accumulation from multiple attributes) can be utilized simultaneously[6].

Based on the above considerations, we redefine the test selection measure to overcome the limitation in handling continuous attributes and then propose a new approach, Flexible NBTree, which induces a hybrid of decision tree and Naive Bayes. Flexible NBTree mitigates the negative effect of information loss on test selection by applying post-discretization strategy[7]: at each internal node in the tree, we first select the test which is the most useful for improving classification accuracy, then apply discretization of continuous tests. The finial decision tree nodes contain univariate splits as regular decision trees, but the leaves contain Naive Bayesian classifiers.

We introduce the post-discretization strategy in Section 1. In Section 2, we describe the hybrid approach—Flexible NBTree. At last, we explain our experimental results and sum up the whole paper in

* Supported by the National Natural Science Foundation of China (Grant No. 60275026)

** To whom correspondence should be addressed. E-mail: wanglimin_74.student@sina.com

Sections 3 and 4, respectively.

1 The post-discretization strategy

1.1 Test selection measure δ

Definition 1. The training set T consists of predictive attributes $\{X_1, \cdots, X_n\}$ and class attribute C. Each predictive attribute X_i is either continuous or discrete.

Definition 2. Let $P(\cdot)$ denote the probability, $p(\cdot)$ refer to the probability density function and Count$(\cdot)$ the size of data set.

The aim of decision tree learning is to construct a tree model which can describe the relationship between predictive attributes $\{X_1, \cdots, X_n\}$ and class attribute C in set T.

$$\text{Tree model: } X_1, \cdots, X_n \rightarrow C.$$

That is, the classification accuracy of the tree model on set T should be the highest. Correspondingly the Bayes measure δ which is introduced in this section as a test selection measure, is also based on this criterion.

Let X represent one of the observable, predictive attributes. If X is discrete, according to Bayes theorem, there will be

$$P(C = c_j \mid X = x_i) = \frac{P(C = c_j, X = x_i)}{P(X = x_i)}, \tag{1}$$

where x_i is the value of attribute X, c_j the class label of testing instance. The aim of Bayesian classification is to decide and choose the class that maximizes the posteriori probability. Since $P(X = x_i)$ in Eq. (1) is the same for all classes, and does not affect the relative values of their probabilities, it can be ignored. When some instances satisfy $X = x_i$, their class labels are most likely to be

$$\begin{aligned} c_i^* &= \arg\max_{c_j \in C} P(C = c_j \mid X = x_i) \\ &= \arg\max_{c_j \in C} P(C = c_j, X = x_i). \end{aligned} \tag{2}$$

Correspondingly, if X is continuous, we will have

$$\begin{aligned} &P(C = c_j \mid X = x_i) \\ &= \frac{p(X = x_i \mid C = c_j) P(C = c_j)}{p(X = x_i)}, \end{aligned} \tag{3}$$

where $p(X = x_i)$ is a constant independent of C and then

$$\begin{aligned} c_i^* &= \arg\max_{c_j \in C} P(C = c_j \mid X = x_i) \\ &= \arg\max_{c_j \in C} p(X = x_i \mid C = c_j) P(C = c_j). \end{aligned} \tag{4}$$

Definition 3. Suppose X_i has m distinct values. We define the Bayes measure δ as:

$$\delta = \frac{\sum_{i=1}^{m} \text{Count}(X = x_i \wedge C = c_i^*)}{N}, \tag{5}$$

where N is the size of set T. Intuitively spoken, δ is the classification accuracy when classifier consists of attribute X only. It describes the extent to which the model constructed by attribute X fits class attribute C. The predictive attribute which maximizes δ is the one that is the most useful for improving classification accuracy.

1.2 Discretization of continuous attributes

The aim of discretization is to partition the continuous attribute values into a discrete set of intervals. According to Eq. (4), we have

$$c_i^* = \arg\max_{c_j \in C} p(X = x_i \mid C = c_j) P(C = c_j),$$

where conditional probability density function $p(X \mid C = c_j)$ is continuous. Given arbitrary values x_i and x_k, when $x_i \rightarrow x_k$, there will be

$$\begin{aligned} &p(X = x_i \mid C = c_j) P(C = c_j) \\ &\quad \rightarrow p(X = x_k \mid C = c_j) P(C = c_j). \end{aligned}$$

So, the class labels inferred from Eq. (4) will not change within a small interval of the values of X. For clarification, suppose the relationship between the distribution of X and C is shown in Fig. 1.

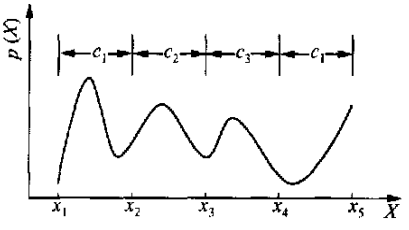

Fig. 1. The relationship between the distribution of X and C.

We can see from Fig. 1 that

$$\begin{cases} C = c_1 \quad (x_1 \leqslant X < x_2 \text{ or } x_4 \leqslant X \leqslant x_5), \\ C = c_2 \quad (x_2 \leqslant X < x_3), \\ C = c_3 \quad (x_3 \leqslant X < x_4). \end{cases} \tag{6}$$

What should be noted is that the attribute values (c_1, c_2, c_3) are inferred from Eq. (4), not the true

class labels of training instances. In the current example, there are three candidate boundaries corresponding to the values of X at which the value of C changes: x_2, x_3, x_4. If we use these boundaries to discretize attribute X, the classification accuracy after discretization will be equal to δ. So, the process of computing δ is also the process of discretization. The Bayes measure δ can also be used to automatically find the most appropriate boundaries for discretization and the number of intervals.

Although this kind of discretization method can retain classification accuracy, it may cause too many intervals. The MDL principle, which is presented by Dougherty et al.[8] to determine a stopping criterion for their recursive discretization strategy, is used in our experimental study to control the number of intervals.

Suppose we have sorted sequence S in ascending order by the values of continuous attribute X. Such a sequence is partitioned by boundary B to two subsets S_1, S_2. The class information entropy of the partition denoted by $E(X, B; S)$ is given by

$$E(X, B; S) = \frac{|S_1|}{|S|}\mathrm{Ent}(S_1) + \frac{|S_2|}{|S|}\mathrm{Ent}(S_2),$$

where $\mathrm{Ent}(\cdot)$ denotes the entropy function,

$$\mathrm{Ent}(S_i) = -\sum_{c_j \in C} P(c_j, S_i)\log_2 P(c_j, S_i)$$

and $P(c_j, S_i)$ stands for the proportion of the instances in S_i that belong to class c_j.

According to MDL principle, the partitioning within S is reasonable iff

$$\mathrm{Gain}(X, B; S) \geqslant \frac{\log_2(N-1)}{N} + \frac{\Delta(X, B; S)}{N},$$

where $\mathrm{Gain}(X, B; S) = \mathrm{Ent}(S) - E(X, B; S)$ is the information gain, which measures the decrease of the weighted average impurity of the partitions S_1, S_2, compared with the impurity of the complete set S. N is the number of instances in set S, $\Delta(X, B; S) = \log_2(3^k - 2) - [k \cdot \mathrm{Ent}(S) - k_1 \cdot \mathrm{Ent}(S_1) - k_2 \cdot \mathrm{Ent}(S_2)]$, k_i is the number of class labels represented in set S_i. This approach can then be applied recursively to all adjacent partitions, thus create the final intervals on attribute X.

1.3 Parameter estimation

Maximum likelihood estimation of the probability and joint probability in Eqs. (2) and (4) is straightforward, then

$$\begin{cases} P(C = c_j) = \dfrac{\mathrm{Count}(C = c_j)}{N}, \\ P(C = c_j, X = x_i) = \dfrac{\mathrm{Count}(C = c_j \wedge X = x_i)}{N}. \end{cases} \tag{7}$$

Kernel-based density estimation is the most widely used non-parametric density estimation technique. Compared with parametric density estimation technique, it does not make any assumption of data distribution. In this paper we choose it to estimate conditional probability density function in Eq. (4):

$$\hat{p}(X = x_i \mid C = c_j) = \frac{1}{nh_j}\sum_{k=1}^{n} K\left(\frac{x_i - x_k}{h_j}\right), \tag{8}$$

where x_k ($k = 1, \cdots, n$) is the corresponding value of attribute X when $C = c_j$, $K(\cdot)$ is a given kernel function $K(t) = (2\pi)^{-1/2}\mathrm{e}^{-t^2/2}$. And h_j is the corresponding kernel width, n is the number of training instances when $C = c_j$.

This estimate converges to the true probability density function if the kernel function obeys certain smoothness properties and the kernel width is chosen appropriately[9]. If h_j chosen is too small then spurious fine structure becomes visible, while if h_j is too large then the bimodal nature of the distribution is obscured. One way of measuring the difference between the true $p(X = x_i \mid C = c_j)$ and the estimated $\hat{p}(X = x_i \mid C = c_j)$ is the expected cross-entropy, an unbiased estimate which can be obtained by leave-one-out cross-validation[10]:

$$CV_{CE} = -\frac{1}{n}\sum_{k=1}^{n}\log\left(\frac{1}{(n-1)h_j}\sum_{i=1, i\neq k}^{n} K\left(\frac{x_i - x_k}{h_j}\right)\right),$$

where $h_j = c_X/\sqrt{n}$ and c_X is chosen to minimize the estimated cross-entropy. In our experiments, we use an exhaustive grid search where grid width is 0.01 and the search is over $c_X \in [0.2, 0.8]$[10].

2 Flexible NBTree: the hybrid learning algorithm

We can now introduce the Flexible NBTree learning algorithm, which is exactly the same as Kohavi's NBTree[6] but in two respects: the method used for discretizing continuous attributes and the Naive Bayesian classifier used for constructing leaf node.

The NBTree learning algorithm pre-discretizes the data by applying an entropy-based algorithm and uses standard Naive Bayes at the leaf node to handle

pre-discretized and discrete attributes. The Flexible NBTree algorithm we propose is shown below. It uses post-discretization strategy to construct decision tree and replaces leaf node with another version of Naive Bayes, Flexible Naive Bayes[11], which can directly handle continuous attributes, thus make discretizaiton unnecessary and the negative effect caused by discretizaiton can be avoided.

Input: a training set S of pre-classified instances.

Output: a hybrid decision tree with Flexible Naive Bayes at the leaves.

1. From the predictive attribute set $X_1, \cdots, X_n$, select test X_i which maximizes δ.

2. If X_i is continuous, partition its value into a discrete set of intervals according to subsection 1.2.

3. Partition S according to the value of X_i. If X_i is continuous, a multi-way split is made for all possible discrete intervals; if X_i is discrete, a multi-way split is made for all possible values.

4. If the descendant node satisfies specific stopping critera, create a Flexible Naive Bayes as the leaf node and return.

5. For each descendant node, the entire process is recursively repeated on the portion of S that matches the test leading to the node.

3 Experimental results and analyses

In order to evaluate the performance of Flexible NBTree, we conducted an empirical study on 12 data sets from the UCI machine learning repository[1)]. Because each data set consists of a set of classified instances described in terms of continuous or discrete attributes, they seemed likely candidates for contrasting the behavior of Flexible NBTree and NBTree. For comparison purpose, the stopping critera in our experiments are the same: the relative reduction in error for a split is less than 5% and there are no more than 30 instances in the node. We also considered another method to provide a reference point: the C4.5 Release 8[1], a well-known algorithm for decision tree induction.

For each domain, we used ten-fold cross validation to evaluate the generalization accuracy of the three induction algorithms. Table 1 summarizes the characteristics of the data sets and compares the experimental results. C4.5 denotes the decision tree that applied a local, MDL inspired penalty to adjust the gain of a binary split to pre-discretize continuous values. NBTree denotes the decision tree that learned from the data sets which were pre-discretized using an entropy-based algorithm. The symbols √(×) denote relatively better (worse) performance of Flexible NBTree to NBTree.

Table 1. Description of data sets and comparison of experimental results

Data Sets	Instances	Continuous attributes	Discrete attributes	C4.5	NBTree	Flexible NBTree
Abalone	4177	7	1	73.8±3.6	75.0±2.1	78.6±6.8√
Anneal	898	6	32	81.3±1.6	85.6±3.5	87.5±5.2√
Australian	690	6	8	65.8±1.9	62.2±3.2	61.0±8.7×
Breast	699	10	0	63.8±4.6	64.3±2.9	65.9±5.8√
Crx	690	6	9	71.6±8.6	75.4±3.9	76.7±2.5√
Diabetes	768	8	0	67.2±2.6	69.8±1.9	71.7±5.7√
German	1000	24	0	66.6±7.3	63.7±1.2	71.8±3.5√
Hypothyroid	2108	7	18	95.5±5.6	97.8±7.0	98.7±1.5√
Letter	20000	16	0	88.8±1.9	93.1±3.6	95.6±5.7√
Optical	5620	64	0	53.8±2.6	55.2±1.8	57.9±3.2√
Sick-enthyroid	2108	7	18	91.2±2.6	95.5±1.3	93.8±7.3×
Vehicle	846	18	0	31.2±6.2	32.5±7.6	36.8±9.5√

The experimental results reveal that Flexible NBTree performed much better than NBTree in ten of the 12 data sets, and not significantly different in the other two cases. We attribute this disparity in accuracy to the effectiveness of post-discretization strategy.

From the viewpoint of information theory, dis-

1) ftp://ftp.ics.uci.edu/pub/machine-learning-databases

cretization will bring about information loss. The more continuous attributes used to predict, the more information to be lost by pre-discretization. We conjecture that the pre-discretization strategy does not take full advantage of the information that continuous attributes supply and it can only partially help the induction process for the data sets we tested. It is the main reason why NBTree and C4.5 performed poorly on data sets Optical and Vehicle. To prove this hypothesis, we sort data sets by the number of continuous attributes. The comparison results are shown in Fig. 2.

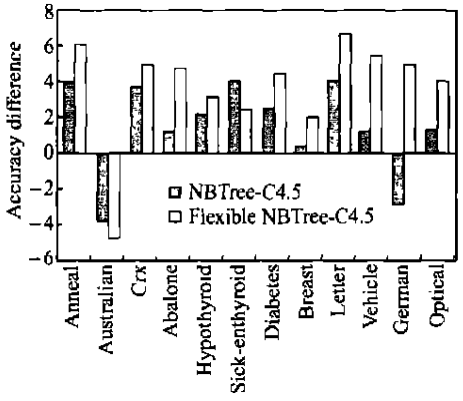

Fig. 2. The accuracy differences. The data sets are sorted by the number of continuous attributes.

We can see from Fig. 2 that Flexible NBTree significantly outperformed NBTree when data sets have many continuous attributes. Especially for data set German, NBTree was relatively worse than C4.5 whereas Flexible NBTree provided a significant increase in accuracy. The experimental results on the natural domains confirm that Flexible NBTree can mitigate the negative effect of information loss by applying post-discretization strategy.

What should be noted is that pre-discretization will improve the efficiency and simplify the procedure of learning algorithm. But Flexible NBTree estimates probability density function based on expected cross-entropy, which will need more cost of computation. The number and the distribution of instances will also affect the experimental results, thus Flexible Tree is more applicable to large data sets, especially when they contain many continuous attributes.

4 Summary

Standard decision tree learning algorithms can not handle continuous attributes. The information loss caused by pre-discretization is one of the main reasons why decision tree performs poorly when data sets consist of many continuous attributes. In this paper, we introduce a novel test selection measure, the Bayes measure, to overcome this limitation. The Bayes measure is based on Bayes theorem to select test, which guarantees the robustness of the performance of the decision tree.

On the basis of this, we propose a hybrid approach, Flexible NBTree, which applies post-discretization strategy to mitigate the negative effect caused by information loss. At the same time, it embodies tradeoff between the accuracy and the complexity of the learned discretization by applying MDL principle.

We present an empirical comparison of different decision tree learning algorithms. Experiments with natural domains showed that Flexible NBTree generalizes much better than NBTree and C4.5, both applying pre-discretization of continuous attributes. Although more work remains to be done, our research to date indicates that Flexible NBTree constitutes a promising addition to the repertoire of induction algorithms.

References

1 Quinlan J. R. Induction of decision trees. Machine Learning, 1986, 81.
2 Quinlan J. R. C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann, 1993.
3 Quinlan, J. R. Improved use of continuous attributes in C4.5. Journal of Artificial Intelligence Research, 1996, 4: 77.
4 Breiman L. et al. Classification and regression trees. Statistics Probability Series, Wadsworth, Belmont, 1984.
5 McCallum, A. K. et al. A comparison of event models for naive bayes text classification. In: Proc. of AAAI-98 Workshop on Learning for Text Categorization, Madison, WI, 1998, 41.
6 Kohavi, R. Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In: Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining, Menlo Park, CA, 1996, 202.
7 Zhou Z. H. et al. Extracting symbolic rules from trained neural network ensemble. AI Communications. 2003, 16(1): 3.
8 Dougherty, J. et al. Supervised and unsupervied discretization of coninuous features. In: Proc. of the 12th International Conference on Machine Learning. San Francisco: Morgan Kaufmann Publishers, 1995, 194.
9 Silverman, B. W. Density estimation for statistics and data analysis. Monographs on Statistics and Applied Probability, 1986.
10 Smyth, P. et al. Retrofitting decision tree classifiers using kernel density estimation. In: Proc. of the 12th International Conference on Machine Learning. San Francisco: Morgan Kaufmann Publishers, 1995, 506.
11 George, H. et al. Estimating continuous distributions in bayesian classifiers. In: Proc. of the 11th Conference on Uncertainty in Artificial Intelligence, Montreal: Morgan Kaufmann Publishers, 1995, 338.