Available online at www.sciencedirect.com
ScienceDirect
Knowledge-Based Systems 19 (2006) 511–515
Knowledge-Based SYSTEMS
www.elsevier.com/locate/knosys

Combining decision tree and Naive Bayes for classification

Li-Min Wang [a,*], Xiao-Lin Li [b], Chun-Hong Cao [c], Sen-Miao Yuan [a]

[a] College of Computer Science and Technology, JiLin University, ChangChun 130012, People's Republic of China
[b] National Laboratory for Novel Software Technology, NanJing University, NanJing 210093, People's Republic of China
[c] Department of Computer Science, NorthEast University, ShenYang 230012, People's Republic of China

Received 2 December 2003; accepted 27 October 2005
Available online 15 June 2006

Abstract

Decision tree is useful to obtain a proper set of rules from a large amount of instances. However, it has difficulty in obtaining the relationship between continuous-valued data points. We propose in this paper a novel algorithm, Self-adaptive NBTree, which induces a hybrid of decision tree and Naive Bayes. The Bayes measure, which is used to construct decision tree, can directly handle continuous attributes and automatically find the most appropriate boundaries for discretization and the number of intervals. The Naive Bayes node helps to solve overgeneralization and overspecialization problems which are often seen in decision tree. Experimental results on a variety of natural domains indicate that Self-adaptive NBTree has clear advantages with respect to the generalization ability.

Keywords: Self-adaptive NBTree; Decision tree; Naive Bayes; Bayes measure; Discretization

1. Introduction

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 web mining, data mining, pattern recognition, signal processing, etc. But standard decision tree learning algorithms can handle discrete attributes only [1,2]. It is a key issue in research to learn from data consisting of both continuous and discrete variables.

The learning algorithms proposed before commonly apply pre-discretization to incorporate continuous-valued predictive attributes into the learned trees. Quinlan [3] argued that typical discretization process in C4.5 gives continuous attributes an unfair advantage over discrete attributes during test selection. He introduced a local, MDL based discretization method to penalize continuous attributes that have many values. Dougherty et al. [4] adopted a global discretization procedure introduced by Catlett [5] prior to induction, rather than locally based on the subset of instances at a node. Their procedure recursively performs binary partitions on continuous attributes based on class information entropy to produce a discretized attribute with multiple values, and uses an MDL based stopping criterion. Auer et al. [6] introduced a local discretization method, T2, which produces *n*-ary (multiple) partitions of the attributes. T2 restricts decision trees to two levels of internal nodes, with only the second level nodes using non-binary splits of continuous attributes to reduce complexity. 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 [7,8]. Since the leaves of decision tree consist of very few instances, we suppose that the distribution of those instances approximately

* Corresponding author. Tel.: +86 431 5394135; fax +86 431 5399040.
E-mail address: jeffreywlm@hotmail.com (L.-M. Wang).

0950-7051/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2005.10.013

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 [9].

We propose in this paper a novel algorithm, Self-adaptive NBTree, which induces a hybrid of decision tree and Naive Bayes. Self-adaptive 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 Bayes measure, which is used to construct decision tree, can directly handle continuous attributes and automatically find the most appropriate boundaries for discretization and the number of intervals. The finial decision tree nodes contain univariate splits as regular decision trees, but the leaves contain General Naive Bayes (GNB), which is introduced in this paper as an extension of standard Naive Bayes and can handle both continuous and discrete attributes. For brevity, we use capital letters, such as X, Y, Z, for attribute names and corresponding lower case letters to denote specific values taken by these attributes (for instance, x_i represents the event that $X_i = x_i$).

2. The post-discretization strategy

2.1. Bayes measure δ

Suppose the training set T consists of predictive attributes $\{X_1, \ldots, X_n\}$ and class attribute C. Each predictive attribute X_i is either continuous or discrete.

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

$$\text{Tree Model}: \quad \{X_1, \ldots, X_n\} \to C$$

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_i represent one of the observable, predictive attributes. If X_i is discrete, according to Bayes theorem, there will be:

$$P(c|x_i) = \frac{P(x_i|c)P(c)}{P(x_i)} \tag{1}$$

where $P(\cdot)$ denote the probability.

The aim of Bayesian classification is to decide and choose the class that maximizes the posteriori probability. Since $P(x_i)$ 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 $X_i = x_i$, their class labels are most likely to be:

$$c^* = \arg\max_{c \in C} P(c|x_i) = \arg\max_{c \in C} P(x_i|c)P(c) \tag{2}$$

Correspondingly, if X_i is continuous, we will have:

$$P(c|x_i) = \frac{p(x_i|c)P(c)}{p(x_i)} \tag{3}$$

where $p(\cdot)$ refers to the probability density. Since $p(x_i)$ is a constant independent of C, then:

$$c^* = \arg\max_{c \in C} P(c|x_i) = \arg\max_{c \in C} p(x_i|c)P(c) \tag{4}$$

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

$$\delta = \frac{\sum \text{Count}(X_i = x_i \wedge C = c^*)}{N} \tag{5}$$

where $\text{Count}(\cdot)$ denotes the size of given subset and N is the size of set T. Intuitively spoken, δ is the classification accuracy when classifier consists of attribute X_i only. It describes the extent to which the model constructed by attribute X_i fits class attribute C. The predictive attribute which maximizes δ is also the one that is the most useful for improving classification accuracy.

2.2. 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:

$$c^* = \arg\max_{c \in C} p(x_i|c)P(c)$$

where conditional probability density function $p(x_i|c)$ is continuous. Given arbitrary values α_j and α_k, when $\alpha_j \to \alpha_k$, there will be

$$p(X_i = \alpha_j|c)P(c) \to p(X_i = \alpha_k|c)P(c)$$

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

We can see from Fig. 1 that,

$$C = \begin{cases} c_1 \ (\text{If } \alpha_1 \leqslant X_i < \alpha_2 \text{ or } \alpha_4 \leqslant X_i \leqslant \alpha_5) \\ c_2 \ (\text{If } \alpha_2 \leqslant X_i < \alpha_3) \\ c_3 \ (\text{If } \alpha_3 \leqslant X_i < \alpha_4) \end{cases} \tag{6}$$

Which should be noted is that, the attribute values (c_1, c_2, and c_3) are inferred from Eq. (4), not the true class labels of testing instances. In the current example, there are three

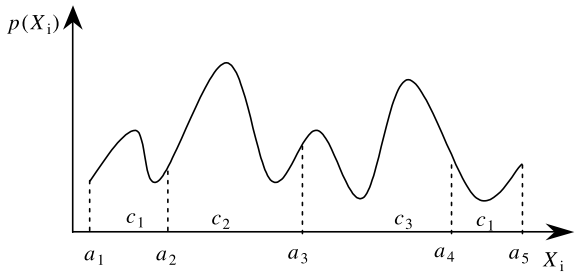

Fig. 1. The relationship between the distribution of X_i and C.

candidate boundaries corresponding to the values of X_i at which the value of C changes: α_2, α_3, and α_4. If we use these boundaries to discretize attribute X_i, 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 [4] 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 into 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.

3. General Naive Bayes (GNB)

Naive Bayes comes originally from work in pattern recognition and is based on one assumption that predictive attributes $X_1, \ldots, X_n$ are conditionally independent given the class attribute C [10], which can be expressed as:

$$P(x_1, \cdots, x_n|c) = \prod_{i=1}^{n} P(x_i|c)$$

But when instance space contains continuous attributes, the situation is different. For clarity, we first just consider two attributes: X_1(continuous) and X_2(discrete). Suppose the values of X_1 have been discretized into a set of intervals, each corresponding to a nominal value. Then the independence assumption should be:

$$\begin{aligned} P(x_1 \leqslant X_1 \leqslant x_1 + \triangle, x_2|c) &= P(x_1 \leqslant X_1 \\ &\leqslant x_1 + \triangle|c)P(x_2|c) \end{aligned} \tag{7}$$

where $[x_1, x_1 + \triangle]$ is arbitrary interval of the values of attribute X_1. This assumption, which is the basis of GNB, supports very efficient algorithms for both classification and learning. Suppose the distribution of X_1 is continuous, by the definition of a derivative,

$$\begin{aligned} &P(c|x_1 \leqslant X_1 \leqslant x_1 + \triangle, x_2) \\ &\quad = \frac{P(c)P(x_1 \leqslant X_1 \leqslant x_1 + \triangle|c)P(x_2|c)}{P(x_1 \leqslant X_1 \leqslant x_1 + \triangle|x_2)P(x_2)} \\ &\quad = \frac{P(c)p(\zeta|c)\triangle P(x_2|c)}{p(\eta|x_2)\triangle P(x_2)} \\ &\quad = \frac{P(c)p(\zeta|c)P(x_2|c)}{p(\eta|x_2)P(x_2)} \end{aligned} \tag{8}$$

where $x_1 \leqslant \zeta, \eta \leqslant x_1 + \triangle$. When $\triangle \to 0$, $\zeta, \eta \to x_1$ and $P(c|x_1 \leqslant X_1 \leqslant x_1 + \triangle, x_2) \to P(c|x_1, x_2)$, hence

$$\begin{aligned} \lim_{\triangle \to 0} P(c|x_1 \leqslant X_1 \leqslant x_1 + \triangle, x_2) &= P(c|x_1, x_2) \\ &= \frac{P(c)p(x_1|c)P(x_2|c)}{p(x_1|x_2)P(x_2)} \end{aligned} \tag{9}$$

We now extend Eq. (9) to handle a much more common situation. Suppose the first k of n attributes are continuous and the remaining attributes are nominal. Similar to the induction process of Eq. (9), we will have

$$\begin{aligned} P(c|x_1, \cdots, x_n) &= \frac{P(c)\prod_{i=1}^{k} p(x_i|c)\prod_{j=k+1}^{n} P(x_j|c)}{p(x_1, \cdots, x_k|x_{k+1}, \cdots, x_n)P(x_{k+1}, \cdots, x_n)} \\ &\propto P(c)\prod_{i=1}^{k} p(x_i|c)\prod_{j=k+1}^{n} P(x_j|c) \end{aligned} \tag{10}$$

The aim of Bayesian classification is to decide and choose the class that maximizes the posteriori probability, then the classification rule of GNB is:

$$\begin{aligned} c^* &= \arg\max_{c \in C} P(c|x_1, \cdots, x_n) \\ &= \arg\max_{c \in C} P(c)\prod_{i=1}^{k} p(x_i|c)\prod_{j=k+1}^{n} P(x_j|c) \end{aligned} \tag{11}$$

4. Parameter estimation

Three kinds of parameters are needed to estimate from training data: $P(c)$, $P(x_j|c)$ and $p(x_i|c)$. Maximum likelihood estimation of the first two parameters is straightforward.

$$\begin{cases} \hat{P}(c) = \frac{\mathrm{Count}(c)}{N} \\ \hat{P}(x_j|c) = \frac{\mathrm{Count}(c \wedge x_j)}{\mathrm{Count}(c)} \end{cases} \tag{12}$$

where N is the number of training instances.

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 $p(x_i|c)$:

$$\hat{p}(x_i|c) = \frac{1}{mh}\sum_{k=1}^{m} K(\frac{x_i - x_{ik}}{h}) \quad (13)$$

where x_{ik} $(k = 1, \ldots, m)$ is the corresponding value of attribute X_i when $C = c$, $K(\cdot)$ is a given kernel function $K(t) = (2\pi)^{-1/2}e^{-t^2/2}$. And h is the corresponding kernel width, m is the number of training instances when $C = c$.

This estimate converges to the true probability density function if the kernel function obeys certain smoothness properties and the kernel width are chosen appropriately [11]. If h is chosen too small then spurious fine structure becomes visible, while if h is too large then the bimodal nature of the distribution is obscured. One way of measuring the difference between the true $p(x_i|c)$ and the estimated $\hat{p}(x_i|c)$ is the expected cross-entropy, an unbiased estimate of which can be obtained by leave-one-out cross-validation [12]:

$$CV_{CE} = -\frac{1}{m}\sum_{k=1}^{m} \log\left(\frac{1}{(m-1)h}\sum_{j=1, j\neq k}^{m} K\left(\frac{x_{ij} - x_{ik}}{h}\right)\right)$$

where $h = c_X/\sqrt{m}$ 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 $c_X \in [0.2, 0.8]$.

5. Self-adaptive NBTree

Self-adaptive NBTree learning algorithm is exactly the same as Kohavi's NBTree [9] but in two respects: the method used for discretizing continuous attributes and the Naive Bayes 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 Self-adaptive NBTree algorithm we propose is shown in Fig. 2. It applies post-discretization strategy to construct decision tree and replaces leaf node with another version of Naive Bayes, GNB, which can directly handle continuous attributes, thus make discretizaiton unnecessary and the negative effect caused by discretizaiton can be minimized.

Input: a training set S of pre-classified instances.
Output: a hybrid decision tree with GNB 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 2.2.
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 criterions, create a GNB as the leaf node and return. If the descendant node belongs to the same class, create a class label 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.

Fig. 2. The induction procedure of Self-adaptive NBTree.

6. Experimental results and analyses

In order to evaluate the performance of Self-adaptive NBTree, we conducted an empirical study on 12 data sets from the UCI machine learning repository [13]. In data sets with missing value, we considered the most frequent attribute value as a candidate. Our experiments compared Self-adaptive NBTree with another state-of-art method, NBTree, for classification. The stopping criterions 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 [3], a well-known algorithm for decision tree induction.

For each domain, we used 10-fold cross validation to evaluate the generalization accuracy of the three induction algorithms. The accuracy of each classifier is based on the percentage of successful predictions on the test sets of each data set. Table 1 shows classification accuracy and standard deviation. 'S-NBTree' denotes the hybrid decision tree proposed in this paper. The symbols $\surd, \times$ denote relatively better (worse) performance of Self-adaptive NBTree to NBTree.

The experimental results reveal that Self-adaptive NBTree performed much better than NBTree in 10 of 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, discretization 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

Table 1
Comparison of experimental results

Data set	C4.5	NBTree	S-NBTree
Abalone	73.82 ± 3.61	75.06 ± 2.13	$78.62 \pm 6.85\surd$
Anneal	81.37 ± 1.63	85.62 ± 3.53	$87.58 \pm 5.24\surd$
Anneal	81.37 ± 1.63	85.62 ± 3.53	$87.58 \pm 5.24\surd$
Australian	65.86 ± 1.97	62.25 ± 3.27	$61.04 \pm 8.78\times$
Breast	63.83 ± 4.65	64.32 ± 2.90	$65.93 \pm 5.82\surd$
Crx	71.60 ± 8.66	75.42 ± 3.94	$76.76 \pm 2.54\surd$
Diabetes	67.26 ± 2.66	69.87 ± 1.97	$71.77 \pm 5.74\surd$
German	66.68 ± 7.32	63.76 ± 1.20	$71.82 \pm 3.56\surd$
Hypothyroid	95.58 ± 5.65	97.82 ± 7.01	$98.73 \pm 1.55\surd$
Letter	88.83 ± 1.97	93.13 ± 3.68	$95.61 \pm 5.74\surd$
Optical	53.86 ± 2.68	55.22 ± 1.88	$57.90 \pm 3.21\surd$
Sick-enthyroid	91.23 ± 2.64	95.56 ± 1.38	$93.81 \pm 7.34\times$
Vehicle	31.23 ± 6.24	32.55 ± 7.66	$36.83 \pm 9.55\surd$

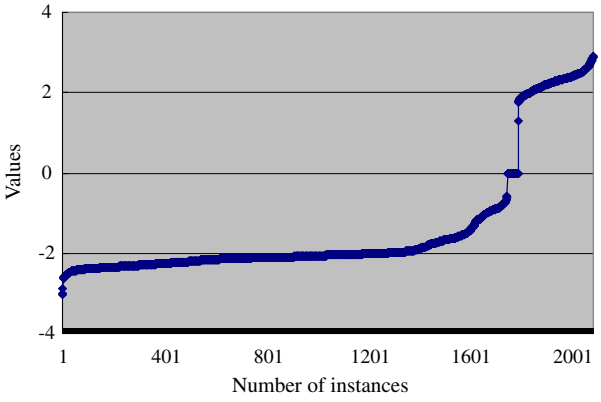

Fig. 3. The distribution of attribute HUE-MEAN.

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.

But Self-adaptive NBTree can mitigate the negative effect of information loss by applying post-discretization strategy, thus its superiority lies in dealing with continuous attributes. We can see from Table 1 that, Self-adaptive NBTree outperformed NBTree when data sets have many continuous attributes. Especially for data set German, NBTree was relatively worse than C4.5 whereas Self-adaptive NBTree provided a significant increase in accuracy.

Furthermore, a common assumption, which is often made in learning algorithms, is that the values of continuous attributes are normally distributed. One can represents such a distribution in terms of its mean and standard deviation, and one can efficiently compute the probability of an observed value from such estimates. However, such an assumption may not be satisfied in real data sets. For example, Fig. 3 shows the distribution of continuous attribute HUE-MEAN in Image data set. On visual inspection, we can clearly see that the distribution was definitely different from normal. Thus, we can conclude from this, the kernel estimator is another reason for Self-adaptive NBTree to improve classification accuracy.

7. 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 guarantee the robustness of the performance of the decision tree.

On the basis of this, we propose a hybrid approach, Self-adaptive 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 Self-adaptive 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 Self-adaptive NBTree constitutes a promising addition to the repertoire of induction algorithms.

Acknowledgement

Supported by the National Natural Science Foundation of People's Republic of China (Grant No. 60275026).

References

[1] J.R. Quinlan, Induction of decision trees, Machine Learning (1986) 81–106.
[2] J.R. Quinlan, C.45: Programs for Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993.
[3] J.R. Quinlan, Improved use of continuous attributes in C4.5, Artificial Intelligence in Research 4 (1996) 77–90.
[4] J. Dougherty, R. Kohavi, M. Sahami, Supervised and unsupervised discretization of continuous features, in: Proceedings of the 12th International Conference on Machine Learning, Morgan Kaufman Publishers, San Francisco, CA, 1995, pp. 191–202.
[5] J. Carlerr, On changing continuous attributes into ordered discrete attributes, Proceedings of the Fifth European Working Session on Learning (1991) 164–178.
[6] P. Auer, R.C. Holte, W. Maass, Theory and applications of agnostic PAClearning with small decision trees, Proceedings of the 12th International Conference on Machine Learning (1995) 21–29.
[7] W. Iba, K. Thompson, An analysis of bayesian classifiers, Proceedings of the 10th Conference on Artificial Intelligence (1992) 223–228.
[8] A. McCallum, K. Nigam, A comparison of event models for naive bayes text classification, Proceedings of the AAAI-98 Workshop Learning for Text Categorization (1998) 41–48.
[9] R. Kohavi, Scaling up the accuracy of naive-bayes classifiers: a decisiontree hybrid, Proceedings of the Second International conference on Knowledge Discovery and Data Mining (1996) 202–207.
[10] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[11] B.W. Silverman, Density estimation for statistics and data analysis, Monographs on Statistics and Applied Probability (1986).
[12] P. Smyth, A. Gray, U. Fayyad, Retrofitting decision tree classifiers using kernel density estimation, Proceedings of the 12th International Conference on Machine Learning (1995) 506–514.
[13] P.M. Murphy, D.W. Aha, UCI repository of machine learning databases, <http://www.ics.uci.edu/~mlearn/>, 1996.