Feature and Classifier Selection in Class Decision Trees

Kazuaki Aoki and Mineichi Kudo

Division of Computer Science
Graduate School of Information Science and Technology
Hokkaido University
Kita-14, Nishi-9, Kita-ku, Sapporo 060-0814, Japan
{kazu,mine}@main.ist.hokudai.ac.jp
http://prml.main.ist.hokudai.ac.jp

Abstract. Feature selection is an important technique in pattern recognition. By removing features that have little or no discriminative information, it is possible to improve the predictive performance of classifiers and to reduce the measuring cost of features. In general, feature selection algorithms choose a common feature subset useful for all classes. However, in general, the most contributory feature subsets vary depending on classes relatively to the other classes. In this study, we propose a classifier as a decision tree in which each leaf corresponds to one class and an internal node classifies a sample to one of two class subsets. We also discuss classifier selection in each node.

1 Introduction

In pattern recognition, feature selection which chooses an effective feature subset for classification is an important technique. Generally, we tend to think that the greater the number of features is, the higher the recognition rate will be. However, when the number of features is large but the number of training samples is small, features that have little or no discriminative information weaken the performance of classifiers. This situation is typically called the *curse of dimensionality*. In that case, it is beneficial to choose a feature subset deriving the highest performance.

The benefits of feature selection are that it enables 1) improvement in the performance of classifiers, 2) reduction in the measurement cost of features and 3) reduction in the computational cost of classifiers in both the training phase and testing phase. Many feature selection methods have been proposed [1,2,3,4,5,6,7,8]. In all of these approaches, the same feature subset in common to classes is chosen. However, it is reasonable to assume that each feature subset has different discriminative information for different classes. For instance, a feature subset that has the most effective information in discriminating class ω_1 from classes ω_2 and ω_3 does not always work for discriminating ω_2 from ω_1 and ω_3. Therefore, in classification problems with many (at least three) classes, such as character recognition, feature selection depending on groups of classes

N. da Vitora Lobo et al. (Eds.): SSPR&SPR 2008, LNCS 5342, pp. 562–571, 2008.

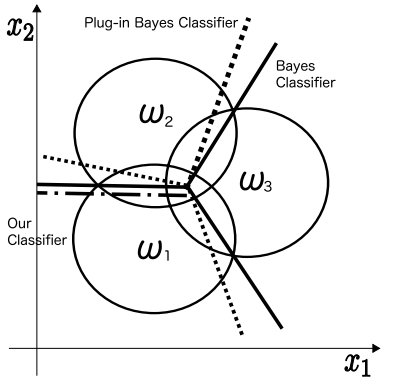

Fig. 1. Example

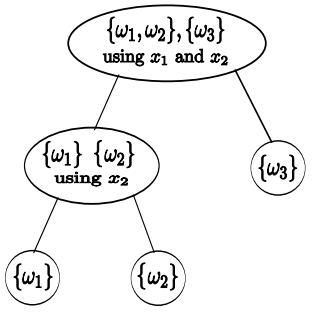

Fig. 2. Class decision tree for Fig.1

is a useful technique. We call such a feature subset a *"class-dependent feature subset."*

According to this idea, we have already proposed a class decision tree classifier using class-dependent feature subsets [9] and we have succeeded in showing the effectiveness in a Chinese character recognition problem of 30 classes. In [9], the classification rate was improved at 1.39% (on the average of four classifiers), and the validity of selected subsets was visually confirmed. In this paper, we investigate how degree such a trial is effective in a variety of problems. Especially, we make clear for which problems this way works most, with respect to the number of classes and the number of features. In addition, we try classifier selection in each internal node of the tree in addition to feature selection. It is well known that the best classifier depends on the problem, so that classifier selection, in each sub-problem corresponding to one internal node, is also promising for improving the performance of the decision tree classifier.

2 Class-Dependent Feature Subset

Our concept is explained by a simple example shown in Fig. 1. In Fig. 1, there are three classes according to normal distributions with the same covariance matrix and different means. The Bayes boundary, therefore, becomes linear. As long as a given training sample is finite, we cannot avoid misclassification in the plug-in Bayes classifier estimated from the training sample. However, we can reduce such misclassification using class-dependent feature subsets. Indeed, in this problem, only feature x_2 has discriminative information between ω_1 and ω_2. Thus, a classifier using only x_2 is expected to perform better than that using x_1 and x_2 in this case. A decision tree is designed naturally shown in Fig. 2.

3 Construction of Decision Tree

Here, we give a summary of the construction way of a class decision tree [9].

3.1 Class Decision Tree

In a class decision tree, the task of each internal node is to classify one group of classes from another group of classes. Here, let two groups of classes be Ω^1 and Ω^2. An internal node is identified by the following information:

1. (Ω^1, Ω^2): two groups of classes to be classified
2. F: feature subset
3. ϕ: classifier

Thus, an internal node t is denoted by four elements as

$$Node^t = \{\Omega_t^1, \Omega_t^2, F_t, \phi_t\}.$$

Our approach is different from conventional decision tree approaches [10,11] in the following two points:

1. In conventional decision tree approaches, each node is split in terms of *impurity* of two divided sample subsets in a feature, whereas in our approach, each node is split in terms of the *separability* between two groups of classes.
2. In conventional decision tree approaches, classification in each node is simple, and usually only one feature is used. In other words, the performance of classification in each node is not so good. Instead, those approaches complement the simplicity with splitting of data many times. In our approach, we split each node by a class-dependent feature subset. Thus, it is expected that the classification performance is improved in individual sub-problems.

3.2 Construction of Decision Tree

A decision tree is constructed in a bottom-up way like hierarchical clustering. The algorithm is as follows.

1. Initialization step: Set $\Omega_i = \{\omega_i\}, (i = 1, 2, \cdots, C)$, $c = C$,$t = 1$. Attach an *unprocessed* mark to all Ω_i. These Ω_i correspond to leaves.
2. Calculate the separability $S_{ij} = S(\Omega_i, \Omega_j)$ of pair (Ω_i, Ω_j) for all unprocessed nodes Ω_i and Ω_j, $(i, j = 1, \cdots, c)$.
3. Choose the pair $(\Omega_{i^*}, \Omega_{j^*})$ with the smallest separability $S_{i^*j^*}$. Let Ω_{i^*} be Ω_{c+1}^1 and Ω_{j^*} be Ω_{c+1}^2. Mark Ω_{i^*} and Ω_{j^*} as *processed*. Select a feature subset F_{c+1} that is effective for discrimination between Ω_{c+1}^1 and Ω_{c+1}^2.
4. Construct a classifier ϕ_{c+1} to classify Ω_{c+1}^1 and Ω_{c+1}^2 with feature subset F_{c+1}. In this step, we have a new node, $Node^{c+1} = \{\Omega_{c+1}^1, \Omega_{c+1}^2, F_{c+1}, \phi_{c+1}\}$.
5. $\Omega_{c+1} = \Omega_{c+1}^1 \bigcup \Omega_{c+2}^1$ and $c \leftarrow c + 1$, $t \leftarrow t + 2$.
6. Repeat steps 2-5 until $t = c$.

Here, choice of a similarity measure $S(\cdot, \cdot)$ between two class subsets is arbitrary, but some kind of estimated classification rate is preferred. In addition, any feature selection algorithm can be used for feature selection.

Table 1. Datasets

Name	#Class	#Feature	#Sample
waveform	3	40	337, 341, 322
glass	6	9	70, 17, 76, 13, 9, 29
mfeat-mor	10	6	200 per class
mfeat-zer	10	47	200 per class
mfeat-kar	10	64	200 per class
mfeat-fou	10	76	200 per class
mfeat-fac	10	216	200 per class
mfeat-pix	10	240	200 per class
letter-recognition	26	16	about 800 per class

In each node with (Ω_1, Ω_2), a sub-problem to be solved is now a two-class problem, but there exist more than two classes essentially. Therefore, the decision boundary is not so simple. For example, we cannot expect a linear boundary to work. Therefore, for classification, we solve the sub-problem as a multi-class problem for all classes included in either Ω_1 or Ω_2. Then we reassign the predicted class to Ω_1 or Ω_2. In this way, a linear classifier behaves like a piecewise linear classifier.

4 Experiments

We dealt with nine datasets from UCI Machine Learning Repository [12]. A summary of the datasets is shown in Table 1.

The separability measure, classifiers, and feature selection method used are as follows:

1. Separability: recognition rate estimated by the leave-one-out technique with 1-NN (the nearest neighbor) classifier (LOO_{1-NN}).
2. Feature selection method: an approach using normal vectors on the estimated Bayes discrimination boundary [8] (kNFS).
3. Classifiers: five classifiers of plug-in Bayes linear classifiers (L), plug-in Bayes quadratic classifier (Q), 1-nearest neighbor classifier (1NN), 3-nearest neighbor classifier (3NN), and support vector machine (S).

In the support vector machine, we used a RBF kernel function with soft margin parameter $C = 100$ and standard deviation $s = 10.0$. The recognition rate was calculated by 10-fold cross validation (5-fold CV only for the small "glass" dataset). Class decision trees were constructed from all training samples. In this first experiment, one classifier was used in common to all nodes, but feature subsets were chosen differently.

4.1 Plane vs. Tree

First, for examining basic performance, we compared two cases: the case in which a classifier is constructed in the usual way (which we call a "plane" approach)

Table 2. Comparison of plane and tree approaches using all features. The numbers are recognition rates in percentage.

Dataset		Classifier							
Name	#Feature	linear		quad		3-nn		svm	
		Plane	Tree	Plane	Tree	Plane	Tree	Plane	Tree
waveform	40	90.81	83.00	**94.75**	88.30	93.43	93.90	94.64	94.30
glass	9	55.50	53.75	52.50	54.23	56.00	63.55	54.00	**72.47**
mfeat-mor	6	46.15	**65.65**	55.25	58.15	44.50	44.80	46.75	43.95
mfeat-zer	47	81.40	78.85	80.60	78.55	79.00	79.10	**81.45**	80.85
mfeat-kar	64	95.10	93.65	96.65	93.65	97.75	97.90	**97.95**	97.45
mfeat-fou	76	80.60	81.70	82.15	78.00	83.75	**84.30**	81.40	82.00
mfeat-fac	216	96.94	97.15	95.90	68.75	95.25	96.85	**97.80**	97.70
mfeat-pix	240	95.65	92.25	94.80	52.45	97.75	**98.05**	84.95	97.90
letter	16	63.92	45.41	79.78	60.46	86.75	91.52	89.00	**91.66**
average	-	78.45	76.83	81.38	70.28	81.52	86.13	80.88	**87.53**

and the case in which the classifier is the same but a decision tree is used for classification (which we call "tree" approach). In this comparison, feature selection is not made. We omitted 1-NN from the comparison because the recognition rates for the plane classifier and tree classifier are the same. The results are shown in Table 2.

In Table 2, the recognition rate increases in some cases and decreases in other cases when the tree approach is used. The reason for the decrease in recognition rate is clear. This is because, in a class decision tree, an error occurred in an upper node cannot be recovered in the lower nodes. While, the improvement cases are quite interesting. The recognition rate was increased in all datasets when the 3-nearest neighbor was used and in 4 of the 9 datasets when SVM was used. In class decision trees, classification is easier in upper nodes and more difficult in lower nodes, so that classification rate is dominated in the lower nodes. It is obvious that, in a lower node, only a few classes and their training samples are concerned for classification. On the other hand, in the plane situation, all classes and all training samples are concerned with classification. In general, a flexible classifier such as k-NN or SVM is strongly affected by a small change in training samples, that is, they have a high variance. As a result, for these sensitive classifiers, class decision trees often work to reduce the sensitivity/variance. In other words, removing unrelated classes/samples is effective for these classifiers.

4.2 Class-Dependent Feature Subsets

Next, we examined the effectiveness of using class-dependent feature subsets. We used the kNFS [8] algorithm for feature selection. For comparison, we used the same feature selection in the corresponding plane approach. That is, a feature subset was chosen commonly to all classes by kNFS. The value of threshold θ needed for kNFS was set to 10.0%. The results are shown in Table 3.

Table 3. Comparison of plane and tree approaches with feature subsets selected by kNFS. The numbers are recognition rates in percentage.

Dataset			Classifier									
Name	#Feature		linear		quad		1-nn		3-nn		svm	
	Plane	Tree	Plane	Tree	Plane	Tree	Plane	Tree	Plane	Tree	Plane	Tree
waveform	34	35.0	91.01	90.60	94.95	94.90	92.83	93.70	94.55	94.60	**95.86**	94.70
glass	7	6.0	55.50	54.20	47.00	52.82	56.50	**64.96**	55.00	64.04	53.00	63.06
mfeat-mor	4	2.8	46.15	36.00	36.00	**47.05**	42.65	35.20	43.65	39.20	43.05	46.35
mfeat-zer	40	40.9	78.50	81.75	**82.50**	82.15	77.55	79.05	79.15	79.20	81.85	80.95
mfeat-kar	52	52.1	95.05	95.60	96.90	96.86	97.30	97.30	97.45	97.45	97.85	**98.05**
mfeat-fou	64	64.9	80.10	82.65	82.80	82.20	82.85	82.95	83.60	**84.15**	80.55	82.95
mfeat-fac	188	182.7	95.80	97.60	92.65	88.95	92.50	96.25	92.80	96.65	97.70	**97.85**
mfeat-pix	203	185.8	95.65	95.65	90.55	61.05	97.70	97.90	97.85	97.70	90.65	**98.40**
letter	13	14.0	61.72	69.47	76.23	80.44	83.91	88.21	83.63	90.70	88.98	**92.31**
average	67.22	64.91	79.93	79.49	81.33	76.27	80.42	81.25	80.85	82.63	81.05	**84.32**

Let us compare the plane approach using selected features (Plane_FS) and the tree approach using class-dependent features (Tree_FS). By using class-dependent features, recognition rates increased on average by 0.83% for 1NN, 1.78% for 3NN, and 3.27% for S but decreased by −0.44% for L and −5.06% for Q. Thus, class-dependent feature selection in class decision trees tends to be more effective for flexible classifiers, or classifiers with a low bias, than for stable classifiers, or classifiers with a low variance.

Fig. 3 shows the relationship between rate of improvement and number of classes. This is the ratio of recognition rate of class-dependent feature subsets to all features both in the class decision trees. Only the **mfeat-fou** dataset was used as a representative dataset of six datasets of the same size (10). From Fig. 3, the advantage of class-dependent feature subsets for a large number of classes is confirmed. The number of selected features is shown in Fig. 4. It is noted

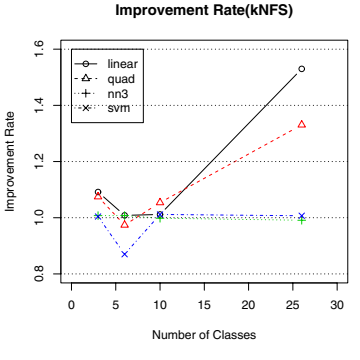

Fig. 3. Rates of Improvement by decision tree with kNFS

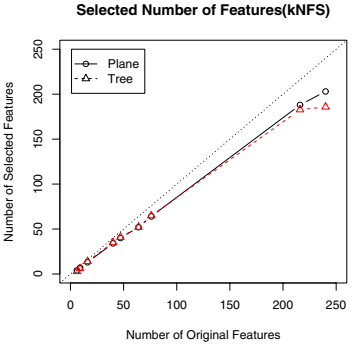

Fig. 4. Comparison of numbers of selected features in plane and tree approaches

that the number of class-dependent features on average is smaller than that of commonly selected features.

It should be noted that trees with class-dependent feature subsets are not always best among the four approaches of Plane, Plane_FS, Tree, and Tree_FS. It depends on datasets. Plane_FS was best for 2 of the 9 datasets, Tree was best for 3 datasets, and Tree_FS was best for 4 datasets with the best classifiers. However, for larger datasets in class, sample or dimensionality, Tree_FS is almost best.

5 Class-Dependent Classifier

Next, we examined the effectiveness of *class-dependent classifier selection*. We selected a *class-dependent classifier* in each internal node as the classifier to solve the corresponding sub-problem at the best recognition rate.

In this experiment, a class-dependent classifier was chosen from five previous classifiers abbreviated as L, Q, 1(NN), 3(NN) and S. In each node, a class-dependent feature subset was first chosen, and then a class-dependent classifier was chosen. Classifier selection is made by 10-fold CV in each node. The results are shown in Table 4 with the best single classifier for each approach.

A comparison with Tree_FS_CS and Tree_FS shows that the recognition rate was increased by using class-dependent classifier in 6 datasets and was decreased in 3 datasets. For degraded cases, it must be the case that CV did not work for classifier selection.

Table 4. Recognition rates with class-dependent classifiers. The characters in parentheses show the best single classifiers among L: linear, Q: quadratic, 1: 1-nearest neighbor, 3: 3-nearest neighbor, S: SVM. Approaches are named Plane: plane with all features, Tree: tree with all features, Plane_FS: plane with feature selection, Tree_FS: tree with feature selection, and Tree_FS_CS: tree with feature selection and classifier selection. In Tree_FS_CS, the characters in parentheses show the set of classifiers used in at least one node.

Dataset			Recognition rate (%)				
	#Class	#Feature	Plane	Tree	Plane_FS	Tree_FS	Tree_FS_CS
waveform	3	40	94.75(Q)	94.30(S)	**95.86(S)**	94.70(S)	95.50(Q,S)
glass	6	9	56.00(L)	**72.47(S)**	56.50(1)	64.96(1)	65.37(L,1,S)
mfeat-mor	10	6	55.25(Q)	**65.65(L)**	46.15(L)	47.05(Q)	65.60(L,Q,1,S)
mfeat-zer	10	47	81.45(S)	80.85(S)	**82.50(Q)**	82.15(Q)	79.75(L,Q,1,3)
mfeat-kar	10	64	97.95(S)	97.90(3)	97.85(S)	**98.05(S)**	94.70(L,Q,3,S)
mfeat-fou	10	76	83.75(3)	84.30(3)	83.60(3)	84.15(3)	**84.40(L,Q,1,3)**
mfeat-fac	10	216	97.80(S)	97.70(S)	97.70(S)	**97.85(S)**	97.30(L,Q,1,3,S)
mfeat-pix	10	240	97.75(3)	98.05(3)	97.85(3)	97.90(1)	**98.15(Q,1,3,S)**
letter	26	16	89.00(S)	91.66(S)	88.98(S)	92.31(S)	**95.05(L,Q,1,3,S)**
average	-	-	83.74	86.99	82.99	84.35	86.20
improvement	-	-	1.00	1.09	0.99	1.01	1.03

6 Total Evaluation

The recognition rates attained by five approaches are summarized in Fig. 5. The recognition rate was averaged over all datasets and all classifiers. From Fig. 5, we can conclude that both class-dependent feature selection and class-dependent classifier selection work well with class decision trees.

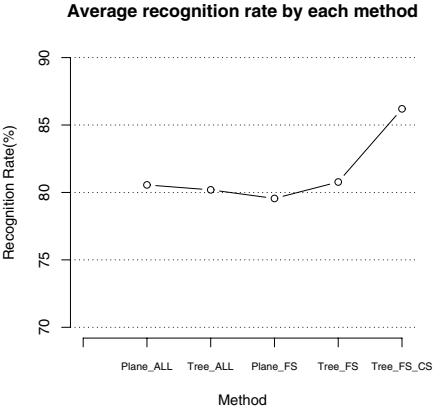

Fig. 5. Average recognition rate attained by each approach. The recognition rate was averaged over all datasets and all classifiers.

7 Discussion

The benefit of the proposed method is not only improvement of the recognition rate but also visualization of problem at hand. The decision tree constructed for **letter** dataset of 26 capital letters is shown in Fig. 6.

The number of selected features is almost 14 and the number of classifiers used is 12 (L or Q) : 4 (nearest neighbors) : 9 (SVM) on 25 internal nodes. It can be seen that k-NN and SVM are selected in lower nodes. Such flexible classifiers are suitable for these difficult sub-problems. In contrast, the linear and quadratic classifiers are selected in upper nodes. These observations imply that easier problems with many training samples should be solved by stable and low-variance classifiers and that more difficult problems with less samples should be solved by unstable but low-bias classifiers.

The features are 16 statistical moments and edge counts (Table 5). The selected features are almost the same in all nodes. Feature No.1 (horizontal position of the box) was removed from all nodes. This is obviously useless for classification. In the node classifying "M" and "W", No.1 and No.3 are removed. No.3 is the width of the box, and it does not contribute to this sub-problem because "M" and "W" have almost the same widths. We also notice that almost all features are invariant for upside down. A similar interpretation is possible for "H" and "K". The removed No.14 is not useful for classifying them.

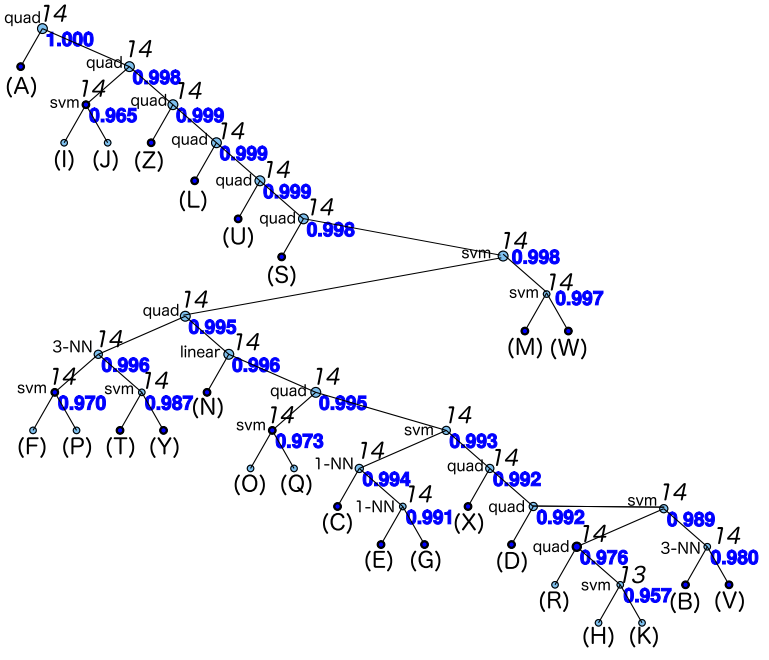

Fig. 6. Decision tree for **letter** dataset. As for a node, clockwise from top, the gray integer denotes the number of selected features, the decimal number denotes the separability, and the string denotes the classifier. In a leaf, the class is shown by the corresponding letter. The original number of features is 16.

Table 5. Details of features in **letter** dataset

No.	Name	Content	Type
1	x-box	horizontal position of box	integer
2	y-box	vertical position of box	integer
3	width	width of box	integer
4	high	height of box	integer
5	onpix	total # on pixels	integer
6	x-bar	mean x of pixels in box	integer
7	y-bar	mean y of pixels in box	integer
8	x2bar	mean x variance	integer
9	y2bar	mean y variance	integer
10	xybar	mean x y correlation	integer
11	x2ybr	mean of x * x * y	integer
12	xy2br	mean of x * y * y	integer
13	x-ege	mean edge count left to right	integer
14	xegvy	correlation of x-edge with y	integer
15	y-ege	mean edge count from bottom to top	integer
16	yegvx	correlation of y-edge with x	integer

8 Conclusion

We have experimentally studied the effectiveness of class-dependent feature subsets and class-dependent classifiers. Using class-dependent feature subsets and

class-dependent classifiers at the same time resulted in significant improvement in class decision trees. In the future, better selection methods should be studied. Other ways to construct trees such as top-down construction and methods for choosing feature subsets should also be studied. If we determine a classifier first in each node, we can use classifier-specific feature selection algorithm, which is known to be superior to classifier-independent feature selection, as adopted in this paper.

References

1. Devijver, P.A., Kittler, J.: Pattern Recognition: A Statistical Approach. Prentice-Hall, Englewood Cliffs (1982)
2. Pudil, P., Novovičová, J., Kittler, J.: Floating search methods in feature selection. Pattern Recognition Letters 15, 1119–1125 (1998)
3. Somol, P., Novovičová, J., Paclík, P.: Adaptive floating search methods in feature selection. Pattern Recognition Letters 15, 1157–1163 (1998)
4. Ferri, F.J., Pudil, P., Hatef, M., Kittler, J.: Comparative study of techniques for large-scale feature selection. Pattern Recognition in Practice IV, 403–413 (1994)
5. Kudo, M., Sklansky, J.: A comparative evaluation of medium- and large-scale feature selection for pattern classifiers. In: 1st International Workshop on Statistical Techniques in Pattern Recognition, pp. 91–96 (1997)
6. Kudo, M., Sklansky, J.: Classifier-independent feature selection for two-stage feature selection. Advances in Pattern Recognition 1451, 548–554 (1998)
7. Zongker, D., Jain, A.: Algorithms for feature selection: An evaluation. In: 13th International Conference on Pattern Recognition, vol. 2, pp. 18–22 (1996)
8. Abe, N., Kudo, M.: Non-parametric classifier-independent feature selection. Pattern Recognition 39, 737–746 (2006)
9. Aoki, K., Kudo, M.: Decision tree using class-dependent feature subsets. In: Caelli, T.M., Amin, A., Duin, R.P.W., Kamel, M.S., de Ridder, D. (eds.) SPR 2002 and SSPR 2002. LNCS, vol. 2396, pp. 761–769. Springer, Heidelberg (2002)
10. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification: Second Edition. John Wiley & Sons, Chichester (2000)
11. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regresion Trees. Wadsworth & Brooks Cole Advanced Books & Software (1984)
12. Asuncion, A., Newman, D.: UCI machine learning repository (2007)