Soft Feature Selection by Using a Histogram-Based Classifier

Hiroshi Tenmoto[1] and Mineichi Kudo[2]

[1] Kushiro National College of Technology, Otanoshike Nishi 2-32-1, Kushiro, Hokkaido 084-0916, Japan
tenmo@kushiro-ct.ac.jp
[2] Division of Computer Science, Graduate School of Information Science and Technology, Hokkaido University, Sapporo 060-0814, Japan
mine@main.eng.hokudai.ac.jp

Abstract. Proposed is a histogram approach for feature selection and classification. The axes are divided into equally-spaced intervals, but the division numbers are different among axes. The main difference from similar approaches is that feature selection mechanism is embedded in the method. The optimal division is determined by an MDL criterion, so that the classifier is guaranteed to converge to the Bayes optimal classifier. We also introduce the concept of "soft feature selection" that is carried out by this method as an extension of traditional "feature selection."

Keywords: Histogram classifier, MDL criterion, Soft feature selection.

1 Introduction

Histogram classifiers are one of strong classifiers that are proved their convergence to the Bayes optimal classifier as the number of samples goes to infinity. Indeed, there are many histogram approaches (for example, [1,2,3]). Their main purpose is to speed up the convergence rate to the Bayes error. In this sense, the main attention is paid for large-scale cases. However, what we often have to face up is small- or medium-scale problems. Then one promising approach is to decrease the dimensionality by feature selection. Through feature selection stage, we can reduce a problem at hand to another problem in which a classifier with a high prediction performance can be constructed, e.g., by gaining a better estimation of the parameters of classifiers. Due to this virtue, many algorithms have been proposed for feature selection (for a comparison among them, see [4]). In this paper, we desire to attain both, that is, feature selection for small-scale and medium-scale cases and Bayes optimality for large-scale cases. This is done by thinking different-size division on feature axes. If no division is done, the feature is not useful for classification. On the other hand, for the axis to which the finest division is done, that axis (feature) is most important. In our approach, such an optimal division is obtained from an MDL criterion.

N. da Vitora Lobo et al. (Eds.): SSPR&SPR 2008, LNCS 5342, pp. 572–581, 2008.

2 MDL Coding of a Histogram-Based Classifier

Let us consider a problem to make a classifier under n training samples in m-dimensional Euclidean space $U = R^m$. Then, a sample is given by

$$x = (x_1, x_2, \ldots, x_m) \in R^m,$$

and a training sample sequence z^n is, with the class set $Y = \{1, 2, \ldots, c\}$, denoted by

$$z^n = (x, y)^n = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \in (U \times Y)^n.$$

According to the MDL principle [5], we measure the cost of sending the class-label sequence y^n under the assumption that a receiver knows x^n, c and m. Let $L(\phi|x^n)$ be the bit length needed to send the classifier ϕ (the classification rule). In addition, let $L(y^n|\phi, x^n)$ be the cost of sending the class-label sequence information when ϕ is given. Then, the total cost is written as

$$L(y^n, \phi|x^n) = L(y^n|\phi, x^n) + L(\phi|x^n).$$

Of course, we would like to have $L(y^n|\phi, x^n)$ smaller than $n \log_2 c$ (the naive bit length of y^n). In our case, $L(\phi|x^n)$ is the bit length to send information of the histogram, that is, the division information of each axis, and $L(S|\phi, x^n)$ is the sum of the bit lengths of the class-label sequences in the cells forming the histogram. We assume that $x_1, x_2, \ldots, x_n$, as well as cells, can be ordered in some way, e.g., a dictionary order with numerical order.

What we think of as a classifier is a histogram. We divide ith axis into 2^{d_i} equally-spaced intervals. The ends of each axis are determined by the minimum and maximum values over training samples. Thus, a partition is expressed by m-tuple

$$\boldsymbol{d} = (d_1, d_2, \ldots, d_m).$$

By d we denote the sum of division indexes as $d = \sum_{i=1}^{m} d_i$. Then there are $\prod_{i=1}^{m} 2^{d_i} = 2^d$ cells.

We want to find the optimal division $\boldsymbol{d}$ in some sense. In our case, we use an MDL criterion for this goal. In the MDL criterion, a shorter length means a better partition for classification. We will describe in order the code length $L(y^n, \phi|x^n)$.

1. (Encoding the label sequence of samples in a cell). Let us consider a cell and let n be the number of samples falling in the cell. Let us denote the class-label sequence as

   $$y_1, y_2, \ldots, y_n.$$

   Then, for encoding this sequence, we need

   $$\log_2 \binom{n+c-1}{c-1} + \log_2 \binom{n}{n_1, n_2, \ldots, n_c} \text{ bits.}$$

Here, we do not need to inform the receiver the value of n. This is because once a partition was known, he/she can count the number of samples falling in each cell. By rewriting n by n^r for the rth cell, we have

$$L(y^n|\phi, x^n) = \sum_{r=1}^{R} \left\{ \log_2 \binom{n^r + c - 1}{c - 1} + \log_2 \binom{n^r}{n_1^r, n_2^r, \ldots, n_c^r} \right\} \text{bits},$$

where R is the number of non-empty cells.

It can happen that some cells have samples only from a single class. To think about this case, let us divide all non-empty R cells into R^P class-pure cells and R^M class-mixture cells. Therefore, for sending the value of R^M and the corresponding R^M positions, we need

$$\log_2 R + \log_2 \binom{R}{R^M} \text{ bits}.$$

In addition, we should send the class label of each class-pure cell. For this, we need

$$R^P \log_2 c \,\text{bits}.$$

As a result, we have

$$L(y^n|\phi, x^n) = \log_2 R + \log_2 \binom{R}{R^M} + R^P \log_2 c + \sum_{r=1}^{R^M} \left\{ \log_2 \binom{n^r + c - 1}{c - 1} + \log_2 \binom{n^r}{n_1^r, n_2^r, \ldots, n_c^r} \right\}. \quad (1)$$

2. (Encoding the partition). We have to let the receiver know $\boldsymbol{d}$ too. First, we encode the number d and then $d_1, \ldots, d_m (d = \sum d_i)$. Thus, we need

$$L(\phi|x^n) = \log_2 \binom{d + m - 1}{m - 1} + \log_2^* d \,\text{bits}.$$

However, in many pattern recognition problems, only a few features are important for classification. In such a case, we can expect to economize the bit length as

$$L(\phi|x^n) = \log_2 m + \log_2 \binom{m}{m - m^+} + \log_2^* d + \log_2 \binom{d}{d_1, d_2, \ldots, d_m} + \log_2 d + \sum_{i=1}^{m^+ - 2} \log_2 d_i, \quad (2)$$

where m^+ of m elements are non-zero d_i's.

In total, from (1) and (2), the bit length is evaluated by

$$L(y^n, \phi|x^n) = L(y^n|\phi, x^n) + L(\phi|x^n)$$

$$
\begin{aligned}
&= \log_2 R + \log_2 \binom{R}{R^M} + R^P \log_2 c \\
&+ \sum_{r=1}^{R^M} \left\{ \log_2 \binom{n^r + c - 1}{c-1} + \log_2 \binom{n^r}{n_1^r, n_2^r, \ldots, n_c^r} \right\} \\
&+ \log_2 m + \log_2 \binom{m}{m - m^+} \\
&+ \log_*^2 d + \log_2 \binom{d}{d_1, d_2, \ldots, d_m} + \log_2 d + \sum_{i=1}^{m^+ - 2} \log_2 d_i
\end{aligned}
$$

We may use an entropy evaluation of this equation for large n. For this, we use the following two approximations

$$
\log_2 \binom{n + c - 1}{c - 1} \simeq (c-1) \log_2 n,
$$

$$
\log_2 \binom{n}{n_1, n_2, \ldots, n_c} \simeq nH\left(\frac{n_1}{n}, \frac{n_2}{n}, \ldots, \frac{n_c}{n}\right) - \frac{c-1}{2} \log_2 n, \tag{3}
$$

where

$$
H\left(\frac{n_1}{n}, \frac{n_2}{n}, \ldots, \frac{n_c}{n}\right) = -\sum_{i=1}^{c} \frac{n_i}{n} \log_2 \frac{n_i}{n}.
$$

Then, $L(y^n, \phi | x^n)$ is rewritten as

$$
\begin{aligned}
L(y^n, \phi | x^n) \simeq &\left(RH\left(\frac{R^P}{R}, \frac{R^M}{R}\right) + \frac{1}{2} \log_2 R + R^P \log_2 c \right) \\
&+ \left(n \sum_{r=1}^{R^M} \frac{n^r}{n} H\left(\frac{n_1^r}{n^r}, \frac{n_2^r}{n^r}, \ldots, \frac{n_c^r}{n^r}\right) + \frac{c-1}{2} \sum_{r=1}^{R^M} \log_2 n^r \right) \\
&+ \left(mH\left(\frac{m^+}{m}, \frac{m - m^+}{m}\right) + \frac{1}{2} \log_2 m \right) \\
&+ \left(\log_2^* d + dH\left(\frac{d_1}{d}, \frac{d_2}{d}, \cdots, \frac{d_m}{d}\right) + \frac{m-1}{2} \log_2 d \right) \\
&= \mathrm{I} + \mathrm{II} + \mathrm{III} + \mathrm{IV}
\end{aligned} \tag{4}
$$

3 Evaluation of the MDL Coding

Let us examine how our criterion (4) works. First of all, it is noted that with this criterion our classification approaches to Bayes optimal classifier as n goes to infinity. In criterion (4), we can see more. First of all, the dominant terms are I, II and IV. When, the perfect classification on training samples is done by a certain $\boldsymbol{d}$, II vanishes because of $R^M = 0$ and $R^P = R$. Then, the problem reduces to minimize terms I and IV. Thus, what should be done is firstly to

minimize the value of d and then to minimize the entropy of $\{d_i/d\}$. That is, this enhances feature selection in which some d_i's are expected to be zero to decrease the entropy. This is the biggest difference from previous similar MDL approaches [1,2,3]. It works often when the number n of training samples is small even when the classification on training samples is not perfect. This is because in such a case terms I and IV are dominant. In summary, our criterion works for feature selection when the sample number is small.

4 Classification of Unknown Samples by Subclass Method

Let us describe how to classify unknown samples in our approach. In a high-dimensional feature space, the regions including training samples may be very sparse. Therefore, we use the subclass method [6] with the histogram approach in order to interpolate the sample-containing regions. This method aims to find a set of hyper-rectangles in a class as its class region. Each hyper-rectangles is called "subclass" and satisfies the following two conditions:

1. Exclusiveness: The hyper-rectangle does not include any sample from negative class.
2. Maximalness: The hyper-rectangle is maximal with respect to the inclusion relation in the family of all subsets holding the above exclusiveness.

Since this problem requires a combinatorial examination with respect to the number of samples, we resort to a randomized algorithm [6] in order to obtain a suboptimal solution.

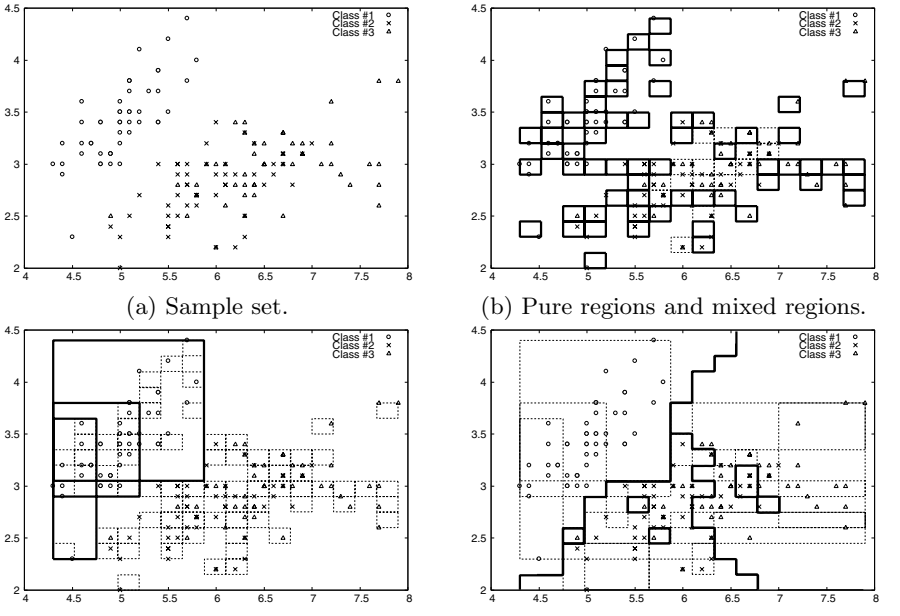
A concrete procedure for the classification is as follows:

1. Find a region including the unknown sample.
2. If the region contains training samples, determine the class label by applying majority rule.
3. If the region does not contain any training samples, determine the class label by using subclasses such that:
   (a) Find subclasses that include the sample-falling region.
   (b) If one or more subclasses are found, adopt the deepest subclass and determine the class label by that subclass.
   (c) If no subclass is found, adopt the nearest subclass and determine the class label by that subclass.

Example results on the first two features of Iris dataset are shown in Fig.1. The training samples are roughly separated by the regions, and subclasses form only on the pure regions. The resultant classification boundary appears based on the regions and subclasses.

5 Experiments

To solve the optimization/minimization problem of (4), we employed a GA. We encoded each d_i in gene and an m-tuple $(d_1, d_2, \ldots, d_m)$ in a chromosome, so that a chromosome shows $\boldsymbol{d}$.



(a) Sample set. (b) Pure regions and mixed regions.

(c) Subclasses on class ω_1 regions. (d) Classification boundary.

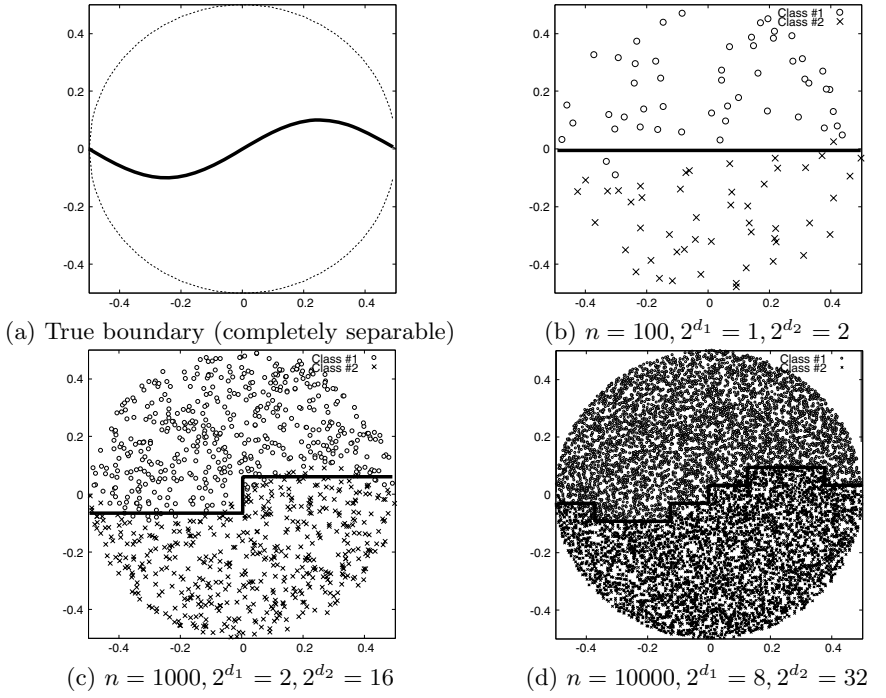
Fig. 1. Classification boundary construction example for the first two features on Iris dataset

In the following experiments, the population size (the number of chromosomes) is 100, the crossover probability is 0.6, and the mutation probability is $1/m$. We used a roulette selection and elite strategy, and terminated the iteration with 100 generations.

5.1 Two-Dimensional Artificial Dataset

At first, we dealt with a 2-dimensional problem with zero Bayes error. In this dataset, the samples distribute uniformly within a circle with radius 0.5 centered at the origin. The two classes are separated by the curve of $y = 0.1\sin(x\pi/0.5)$.

The true boundary, and the classification boundaries obtained by the proposed method are shown in Fig.2(a)–(d), varying the number n of samples. The change of the training and test error is also shown in Fig.3. The test errors are estimated by hold-out method. From these results, we can see: 1) the determined boundary approaches to the true one as n grows, 2) the division becomes finer as n increases and the ratio d_2/d_1 reflects faithfully the ratio of importance between the corresponding two features, and 3) feature selection ($2^{d_i} = 1$) is done when n is small. Especially, we can see that "soft feature selection" was done according to the increase of n. We can regard ith feature with a low ratio d_i/d as being not so important for classification. This is "soft feature selection" in which the importance of each feature is evaluated by this ratio and is an extension of



(a) True boundary (completely separable) (b) $n = 100, 2^{d_1} = 1, 2^{d_2} = 2$

(c) $n = 1000, 2^{d_1} = 2, 2^{d_2} = 16$ (d) $n = 10000, 2^{d_1} = 8, 2^{d_2} = 32$

Fig. 2. Refinement of classification boundary by the increase of samples. The symbols and lines denote the training samples and classification boundaries, respectively.

traditional use-or-not type feature selection. We might remove features with low ratios if we hope. It is reasonable to think of every feature as having at least some amount of contribution for classification. However, we cannot know this amount from a limited number of training samples. On the contrary, when an enough number of training samples is available, this amount would appear so that the value of d_i for such a feature takes a small but positive number. In the light of soft feature selection, we can deal with this amount properly.

To examine the performance as (soft) feature selection, we rotated the boundary so as to change the importance of individual features. The results are shown in Fig.5(a)–(d). From these results, we can see that the difference of the importance of features are captured well by the change of the ratio d_i/d.

5.2 Ten-Dimensional Artificial Dataset (Friedman Dataset)

To examine the performance of this method in high-dimensional cases, we used a famous two-class ten-dimensional artificial dataset "Friedman."

In this dataset, the discriminative features are leading four features, and the trailing six features are non-discriminative. The samples of class ω_1 simply distribute according to the 10-dimensional multivariate normal distribution with zero mean and the unit covariance matrix. The distribution of the samples of

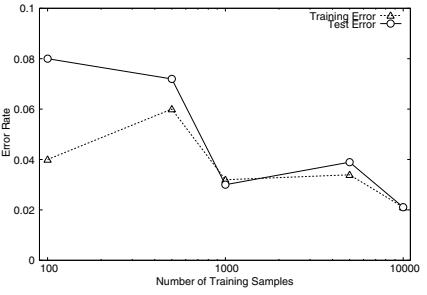


Fig. 3. Training and test errors for two-dimensional artificial dataset

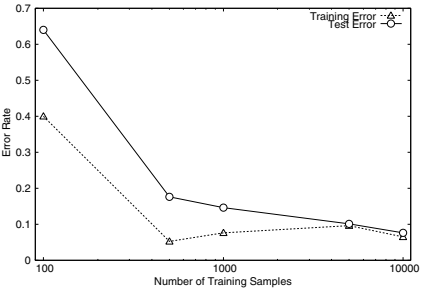
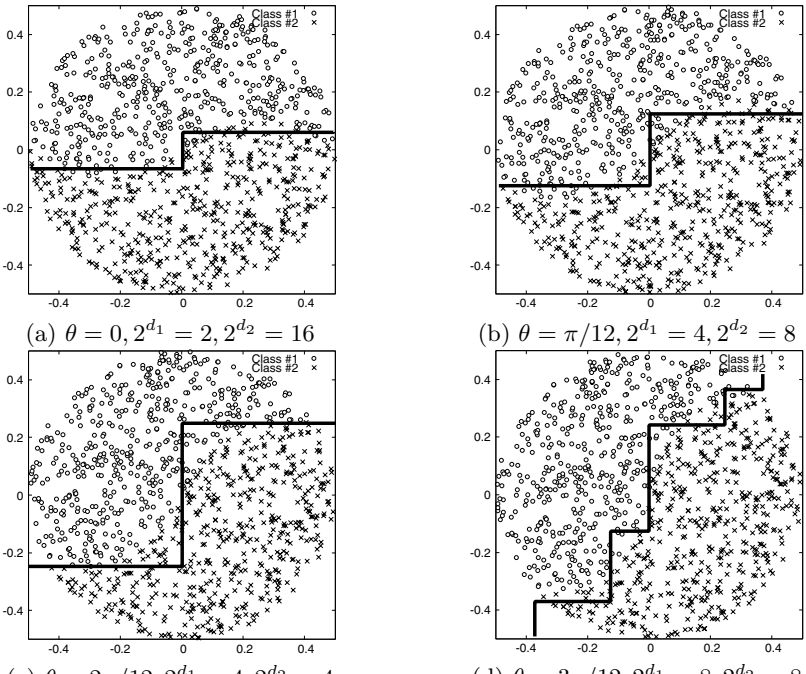


Fig. 4. Training and test errors for ten-dimensional artificial dataset (Friedman dataset)



(a) $\theta = 0, 2^{d_1} = 2, 2^{d_2} = 16$ (b) $\theta = \pi/12, 2^{d_1} = 4, 2^{d_2} = 8$

(c) $\theta = 2\pi/12, 2^{d_1} = 4, 2^{d_2} = 4$ (d) $\theta = 3\pi/12, 2^{d_1} = 8, 2^{d_2} = 8$

Fig. 5. Change of division for the different rotations of true boundary. The number of samples n is fixed at 1000.

class ω_2 is as follows. For the first four features, the samples distribute uniformly between two hyper-spheres with radius 3.5 and 4.0, and for the last six features, the distribution is the same as that of ω_1.

The experimental result is shown in Table 1 and Fig. 4. When the number of samples is small, the training error is high, therefore the proposed method removes too many features. While, in medium- or large-size samples, only

Table 1. Division for Friedman data

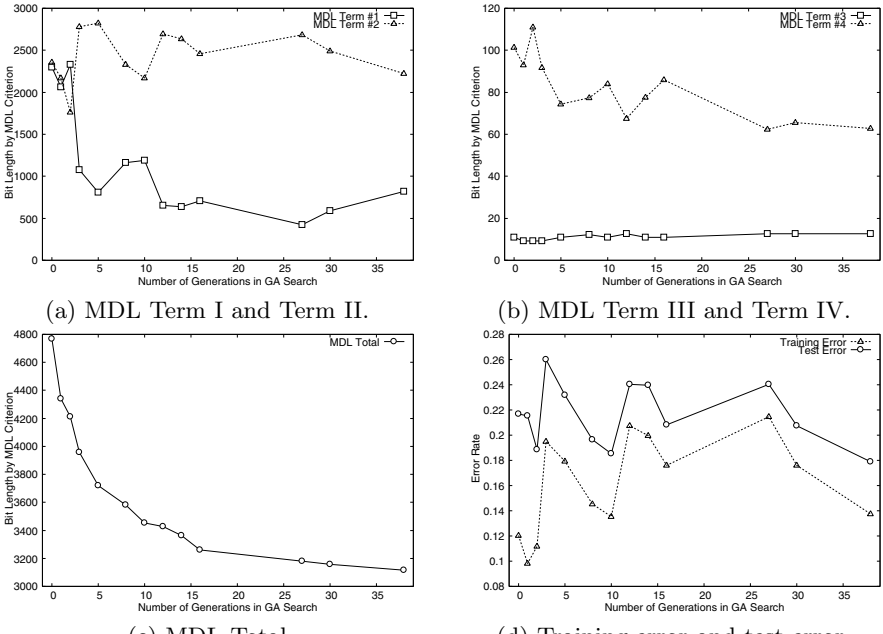
n	2^{d_1}	2^{d_2}	2^{d_3}	2^{d_4}	2^{d_5}	2^{d_6}	2^{d_7}	2^{d_8}	2^{d_9}	$2^{d_{10}}$
100	1	1	4	1	1	1	1	1	1	1
500	4	4	4	4	1	1	1	1	1	1
1000	4	4	4	4	1	1	1	1	1	1
5000	4	4	4	4	1	1	1	1	1	1
10000	4	4	8	8	1	1	1	1	1	1
True	Necessary				Garbage					

necessary four features are chosen properly. Thus, feature selection were performed successfully.

5.3 Real Dataset (Ship Dataset)

We also tested the proposed method on real dataset called "Ship dataset" [7] in which 8 types of military ships are distinguished by 11 features, and the number of samples is 2545.

We can see that the proposed method works also well for the real dataset. As the generation proceeds, the MDL term IV decreases, so the feature selection is done. We can observe that the errors are not increasing while the number of features are getting reduced.



(a) MDL Term I and Term II.

(b) MDL Term III and Term IV.

(c) MDL Total

(d) Training error and test error.

Fig. 6. MDL value and error rate for Ship dataset

6 Conclusion

An MDL-based histogram classifier was proposed. In this method, feature selection mechanism is embedded in the optimization function which determines the optimal division of feature axes. This is a big difference from the previous similar approaches. It is shown that feature selection is done for small-scale (the number of training sample is small) problems and a quasi Bayes classification is done for large-scale problems. In addition, (soft) feature selection is done for medium-scale problems in which the degree of importance of each feature is evaluated by the ratio of the division number of the feature axis to the total number of division numbers. This means that use-or-not feature selection is not always appropriate when a rational number of training samples is available.

Although its convergence to the Bayes optimal classifier is guaranteed, the detailed analysis is needed for the convergence rate of this approach. In addition, we will reconsider our MDL criterion so that the convergence rate would be improved.

Acknowledgment

This research was partly supported by the Grant-in-Aid for Young Scientists (B) No. 20700203 of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1. Rissanen, J., Yu, B.: MDL Learning. In: Kueker, D.W., Smith, C.H. (eds.) Learning and Geometry: Computational Approaches, pp. 3–19. Birkhauser, Basel (1998)
2. Yamanishi, K.: Learning Non-Parametric Smooth Rules Using Stochastic Rules with Finite Partitioning. In: Proceedings of the Computational Learning Theory: Euro-COLT 1993, pp. 217–227 (1993)
3. Tsuchiya, H., Itoh, S., Mashimoto, T.: An Algorithm for Designing a Pattern Classifier by Using MDL Criterion. IEICE Transactions on Fundamentals, E79-A 6, 910–920 (1996)
4. Kudo, M., Sklansky, J.: Comparison of Algorithms that Select Features for Pattern Classifiers. Pattern Recognition 33(1), 25–41 (2000)
5. Rissanen, J.: Stochastic Complexity in Statistical Inquiry. Series in Computer Science, vol. 15. World Scientific, Singapore (1989)
6. Kudo, M., et al.: Construction of Class Regions by a Randomized Algorithm: A Randomized Subclass Method. Pattern Recognition 29, 581–588 (1996)
7. Park, Y., Sklansky, J.: Automated Design of Multiple-Class Piecewise Linear Classifiers. Journal of Classification 6, 195–222 (1989)