Kybernetika

Hiroshi Tenmoto; Mineichi Kudo; Masaru Shimbo

Piecewise linear classifiers preserving high local recognition rates

Kybernetika, Vol. 34 (1998), No. 4, [479]--484

Persistent URL: http://dml.cz/dmlcz/135235

KYBERNETIKA — VOLUME 34 (1998), NUMBER 4, PAGES 479-484

PIECEWISE LINEAR CLASSIFIERS PRESERVING HIGH LOCAL RECOGNITION RATES

HIROSHI TENMOTO, MINEICHI KUDO AND MASARU SHIMBO

We propose a new method to construct piecewise linear classifiers. This method constructs hyperplanes of a piecewise linear classifier so as to keep the correct recognition rate over a threshold for a training set. The threshold is determined automatically by the MDL (Minimum Description Length) criterion so as to avoid overfitting of the classifier to the training set. The proposed method showed better results in some experiments than a previous method.

1. INTRODUCTION

In pattern recognition, nonparametric classifiers are effective when the assumption of a statistical model cannot be made on the basis of the underlying distribution of samples. A piecewise linear classifier is a typical nonparametric classifier and approximates the true discrimination boundary by a combination of some hyperplanes.

Many methods have been proposed for construction of piecewise linear classifiers [4–7, 9–11]. Park and Sklansky's method [7] is the most effective and least restrictive one. It aims to separate prototypes belonging to different classes, where the prototypes are the cluster centers of the training samples of each class. Therefore, unless the prototypes properly represent the samples around them, the method does not work well. It is especially difficult for prototypes to represent training samples located at class boundaries. Their method, therefore, depends strongly on the result of clustering and often fails to discriminate even the training samples.

In our method, prototypes and training samples are evenly used, and hyperplanes are constructed incrementally so as to keep the local recognition rate over a threshold. However, in general, a high recognition rate for the training samples does not imply the same performance for many unknown samples. Overfitting to the training samples causes the degradation of performance. Therefore, we determine an appropriate value of the threshold on the basis of the MDL criterion [8].

2. CONSTRUCTION OF PIECEWISE LINEAR CLASSIFIERS

2.1. Basic algorithm

Our method is based on Park and Sklansky's method [7]. Both methods are summarized by the following basic algorithm.

Step 1: Using a clustering method (e. g., Forgy's algorithm [2]), find some clusters over the training samples in each class, and let the cluster centers be *prototypes*.

Step 2: Among all links connecting pairs of different-class prototypes, find Tomek links [12], where a link is said to be a *Tomek link* when the hypersphere with the link as the diameter does not include other prototypes. For simplicity, we refer to a Tomek link as a link.

Step 3: Find some hyperplanes so as to cut all the (Tomek) links.

Step 4: Assign a class label to each region surrounded by the hyperplanes by applying the majority rule to the number of training samples that fall in the region. Classify an unknown sample according to the label of the region where the sample falls in. For a region without the label, reject the sample (with-reject mode) or adopt the nearest region to the sample (without-reject mode).

The primary part of the algorithm is Step 3. In Step 3, the previous method finds the nearly minimum number of hyperplanes that is enough for separating all the prototypes. However, there is no guarantee that the hyperplanes can separate all training samples as well. This problem may be solved by increasing the number of prototypes, although it increases the computational cost.

In our method, a hyperplane is first found on the basis of the prototypes, and then the location is corrected on the basis of the training samples. An addition of a link to the same hyperplane is carried out only when the addition keeps the local recognition rate high. The previous method also has a similar correction mechanism, but the cutting of many links has priority over such a correction.

2.2. Incremental construction of hyperplanes

Next, we show our concrete procedure for the construction of hyperplanes.

Step 0: Let the set of links be L. The value of the upper bound $\epsilon_{\max}$ of the local error rate is determined.

Step 1: Repeat the following steps until L becomes empty.

Step 2: Select the longest link $l \in L$ as an initial link, and let the perpendicular bisector be an initial hyperplane h. Let $L_h = \{l\}$ and $L = L - \{l\}$. Let p and n be the prototypes of l located on the positive side and the negative side with respect to h, respectively. Let a *positive prototype set* $P = \{p\}$ and a *negative prototype set* $N = \{n\}$. Make a *local positive set* S_P of the training samples belonging to the same cluster with $p \in P$. In a similar way, make a *local negative set*, S_N (Figure 1(a)). Next, train h locally so as to classify S_P and S_N more correctly by *Window Training Procedure* [10]. Copy L to L'.

Step 3: Find the link $l' \in L'$ nearest to L_h, where the nearness is measured by the distance $D(l', L_h) = \min_{l \in L_h} d(l', l)$, here $d(\cdot, \cdot)$ is the distance between two link centers. Next, let $L' \leftarrow L' - \{l'\}$ and $L_h \leftarrow L_h \cup \{l'\}$. If h also cuts l' simultaneously, try to add both prototypes of l' to P and N according to the signs with respect to h. If both prototypes are located on the negative side, the prototype nearer to h is added to P, and the other is added to N. If both prototypes are located on the positive side, add them to N and P, conversely. Then, reconstruct S_P and S_N by collecting the training samples belonging to the same cluster as at least one prototype of P and N, respectively (Figure 1(b)).

Step 4: Train h locally for S_P and S_N (Figure 1(c)). Calculate the local error rate ϵ.

Step 5: If $\epsilon \leq \epsilon_{\max}$, update L and L_h as $L = L - \{l'\}$ and $L' \leftarrow L$. Furthermore, we limit the addition of the links only when at least one side of h is always of one class. If both the limitations are satisfied, return to Step 3. Otherwise, cancel the addition as $L_h \leftarrow L_h - \{l'\}$ (P and N are returned to the situation they were in before the addition of l'), and return to Step 3 in order to find a second nearest link to L_h. If there is no link to satisfy the limitations, or if the number of canceled links is beyond a given number K, terminate the addition to L_h, and return to Step 2 for finding another hyperplane, h'.

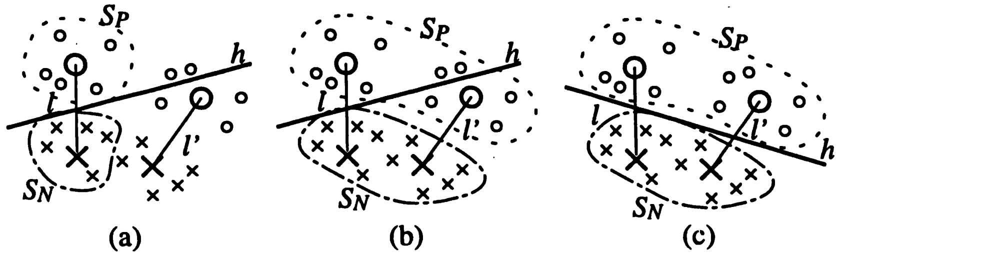


Fig. 1. Construction of hyperplanes by the proposed method. The small and large symbols denote the training samples and the prototypes, respectively.

2.3. Determination of threshold by MDL criterion

In the proposed method, the recognition rate for the training samples is controllable by a threshold $\epsilon_{\max}$. With a small $\epsilon_{\max}$, we can obtain a classifier which can discriminate the training samples well. However, fitting a classifier too close to the samples does not necessarily improve the performance for many unknown samples. There is an appropriate value of $\epsilon_{\max}$ for every given problem.

To estimate the appropriate value, we use the MDL criterion [8], which is one of the probabilistic model selection criteria. The MDL criterion selects a certain model M from a model class $\mathcal{M}$ such that M minimizes the description length of the data and M simultaneously. That is, we require the classifier to be as simple as possible and to classify correctly as many training samples as possible at the same time.

The MDL value is denoted by $L_{\text{MDL}} = L(X^N|\theta) + L(\theta|M) + L(M)$, where X^N denotes given N training samples and θ is a real-valued parametric vector of the model M. The first term is the description length of X^N under a particular θ of M. It is calculated as the log-likelihood of θ with respect to X^N, i.e., $-\log_2 P(X^N|\theta)$.

The second term is the description length of θ, and the third term is the description length of M. By summing up these lengths, we obtain the value of L_{MDL}.

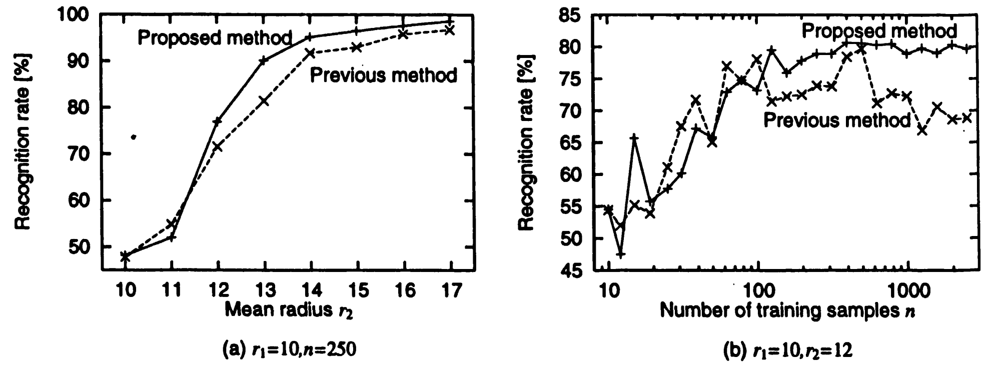
The practical calculation is as follows. We consider a finite partitioning model (for example, see reference [13]). The universal region is assumed to be partitioned into R regions $\{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_R\}$ by the hyperplanes. In region $\mathcal{R}_r$, let N_r, N_r^+ and N_r^- be the number of samples, the number of samples of the most dominant class and the number of samples of the other classes, respectively. Then, a maximum likelihood estimator of a binomial distribution for $\mathcal{R}_r$ is given by $\hat{\theta}_r = N_r^+/N_r$. Thus, the first term is calculated by $\sum_{r=1}^{R} -\log_2 \hat{\theta}_r^{N_r^+}(1-\hat{\theta}_r)^{N_r^-} = \sum_{r=1}^{R} N_r\{-\hat{\theta}_r \log_2 \hat{\theta}_r - (1-\hat{\theta}_r)\log_2(1-\hat{\theta}_r)\}$. For the second term, we use $\frac{1}{2}(D+1)H(\log_2 N + \log_2 e)$, where D and H are the number of features and the number of hyperplanes, respectively. In the last term of L_{MDL}, we identify model M by encoding the number of hyperplanes as $\log_2^* H$, where $\log_2^* H = 1.519 + \log_2 H + \log_2 \log_2 H + \cdots$, and the summation is taken only for positive terms [8]. We choose an appropriate number of hyperplanes where L_{MDL} takes the minimum.

3. EXPERIMENTS

All experiments were performed on an *Intel Pentium* 200MHz machine with *BSD/OS*. Throughout all the experiments, we determined the value of K for the terminal condition by $2D$, i.e., twice the number of features.

A. Artificial data

An experiment was performed using a two-class set of artificial data, in which two distributions form double rings with the center at the origin in a 2-dimensional space. The radius of Class 1 varies according to the normal distribution N(r_1, 1), and that of Class 2 varies according to N(r_2, 1). There is a considerable overlap between the two classes when $|r_1 - r_2|$ is small. For each class, we used n ($10 \leq n \leq 2511$) samples for training and 1000 samples for test.



(a) r_1=10,n=250

(b) r_1=10,r_2=12

Fig. 2. Recognition rates of the proposed method and the previous method with changes in (a) the value of r_2 and (b) the value of n.

We examined the robustness of the proposed method, changing (a) the separability of the two distributions and (b) the number of training samples. In test (a), r_1 is fixed at 10, r_2 is varied from 10 to 17 and n is fixed at 250. In test (b), r_1 is fixed at 10, r_2 is fixed at 12 and n is increased from 10 to 2511 with a log scale step such as $\lfloor 10^k \rfloor$ ($k = 1, 1.1, \ldots, 3.4$). Figure 2 shows the results. The proposed method outperformed the previous method in discrimination, except when the classes were very close ($r_2 < 12$) or the number of the training samples was very small ($n < 100$).

Under conditions of such a small amount of information, the number of hyperplanes in the proposed method was too small. The MDL criterion generally tends to underestimate in such a case. By the proposed method with an optimal number of hyperplanes, we can expect a better result. As an alternative, we may adopt a constant value (e. g., 0.1) for $\epsilon_{\max}$ without MDL estimation.

B. Real data

Experiments were performed on two practical problems: (a) 26-class, 10-feature alphabetical character recognition (*ETL-3* database [1]) and (b) 5-class, 6-feature Japanese vowel recognition (*ETL-WD-I* database [3]). In experiment (a), the number of training samples per class is 100, and that of test samples is 100. In experiment (b), the number of training samples per class is 100, and that of test samples is 500. For all experiments, ten prototypes are used. The values of $\epsilon_{\max}$ estimated by the MDL criterion were 0.12 in (a) and 0.08 in (b), respectively.

The results are shown in Table 1. Especially in with-reject mode, the proposed method was better than the previous method in the recognition rate, while the number of hyperplanes of the proposed method was smaller than that of the previous method. This means that our MDL criterion works sufficiently to avoid overfitting to the training samples. The results without MDL estimation (in this case, $\epsilon_{\max} = 0.1$) were also good. These results indicate the usefulness of a constant value for $\epsilon_{\max}$ when we want to economize the computational cost.

Table 1. Experimental results. Here, M, R_1 and R_2 correspond to the number of hyperplanes, the recognition rate in with-reject mode, and the recognition rate in without-reject mode, respectively.

Dataset	Method	M	L_{MDL}	R_1	R_2
(a) ETL-3	Proposed with MDL ($\epsilon_{\max} = 0.12$)	33	2519	75.04	93.27
	Proposed (fixed $\epsilon_{\max} = 0.1$)	35	2643	74.23	93.04
	Previous	44	3157	51.65	92.23
(b) ETL-WD-I	Proposed with MDL ($\epsilon_{\max} = 0.08$)	10	470.7	77.88	82.44
	Proposed (fixed $\epsilon_{\max} = 0.1$)	11	505.4	77.12	81.52
	Previous	19	721.3	57.08	82.20

4. DISCUSSION AND CONCLUSION

We proposed a new method for constructing piecewise linear classifiers, in which each hyperplane is constructed so as to keep the local error rate under a threshold that is determined by the MDL criterion. The results of experiments showed the effectiveness of the proposed method.

As in the case of the previous method, our method also depends on the result of clustering, i. e., the prototypes. In a future study, we will try to develop a construction method without clustering. The use of computational geometry techniques with probabilistic algorithms may be one possibility. Determination of the optimal number of clusters using the MDL criterion will also be studied.

(Received December 18, 1997.)

REFERENCES

[1] Electrotechnical Laboratory / Japanese Technical Committee for Optical Character Recognition, 1993, ETL3, ETL Character Database. Electrotechnical Laboratory, Japan.

[2] E. W. Forgy: Cluster analysis of multivariate data: Efficiency versus interpretability of classifications. Abstracts Biometrics *21* (1965), 3, 768.

[3] S. Hayamizu et al: Generation of VCV/CVC balanced word sets for speech database. Bull. Electrotechnical Laboratory *49* (1985), 10, 803–834.

[4] O. L. Mangasarian: Multisurface method of pattern separation. IEEE Trans. Inform. Theory *14* (1968), 6, 801–807.

[5] W. S. Meisel: Computer-Oriented Approaches to Pattern Recognition. Academic Press, New York 1972.

[6] N. J. Nilsson: Learning Machines. McGraw-Hill, New York 1965.

[7] Y. Park and J. Sklansky: Automated design of multiple-class piecewise linear classifiers. J. Classification *6* (1989), 195–222.

[8] J. Rissanen: A universal prior for integers and estimation by minimum description length. Ann. Statist. *11* (1983), 416–431.

[9] J. Sklansky and L. Michelotti: Locally trained piecewise linear classifiers. IEEE Trans. Pattern Analysis and Machine Intelligence *2* (1980), 101–111.

[10] J. Sklansky and G. N. Wassel: Pattern Classification and Trainable Machines. Springer-Verlag, New York 1981.

[11] R. Takiyama: A learning procedure for multisurface method of pattern separation. Pattern Recognition *12* (1980), 75–82.

[12] I. Tomek: Two modifications of CNN. IEEE Trans. Systems Man Cybernet. *6* (1976), 11, 769–772.

[13] K. Yamanishi: A learning criterion for stochastic rules. In: Machine Learning, to appear. An extended abstract in Proceedings of the Third Annual Workshop on Computational Learning Theory, 1990, pp. 67–81.

Hiroshi Tenmoto, Mineichi Kudo and Masaru Shimbo, Division of Systems and Information Engineering, Graduate School of Engineering, Hokkaido University, Sapporo 060-8628. Japan.

e-mail: tenmo, mine, shimbo @main.eng.hokudai.ac.jp