A Classifier Based on Distance Between Test Samples and Average Patterns of Categorical Nearest Neighbors

Seiji HOTTA, Senya KIYASU, and Sueharu MIYAHARA
Department of Computer and Information Sciences, Nagasaki University
Bunkyo-machi 1–14, Nagasaki-shi, Nagasaki 852-8521, Japan
{hotta, kiyasu, miyahara}@cis.nagasaki-u.ac.jp

Abstract

The recognition rate of the typical nonparametric method "k-Nearest Neighbor rule (kNN)" is degraded when the dimensionality of feature vectors is large. Another nonparametric method "linear subspace methods" cannot represent the local distribution of patterns, so recognition rates decrease when pattern distribution is not normal distribution. This paper presents a classifier that outputs the class of a test sample by measuring the distance between the test sample and the average patterns, which are calculated using nearest neighbors belonging to individual categories. A kernel method can be applied to this classifier for improving its recognition rates. The performance of those methods is verified by experiments with handwritten digit patterns and two class artificial ones.

1. Introduction

Nonparametric methods can be used with arbitrary data distributions and without the assumptions that the forms of the underlying pattern densities are known [1]. There are several types of nonparametric methods in pattern recognition. The typical one is the k-nearest neighbor (kNN) rule. The kNN rule has been implemented on pattern recognition systems because of its good performance and simple algorithm. In the kNN rule, the class of a test sample is chosen as the class of the majority of its k-nearest neighbors [1, 2]. This approach includes the following features: 1) It has been proved that the error rate of kNN is close to the Bayes error when both the number of training samples and the value of parameter k are infinite. 2) We can design the classifier by kNN even if the number of training samples is few. 3) We can implement kNN when the distribution of classes is overlapped with each other. 4) kNN can be implemented easily due to its simple algorithm. The main drawback to kNN is that recognition rates deteriorate when the number of dimensions of a feature vector is large [3]. For example, Figure 1 shows the example of a test sample from the MNIST dataset [4], and its five nearest training samples, which are evaluated using the Euclidean distance. In this example, the test sample is misclassified to '8' because the selected five training samples include the three samples of the class 8.

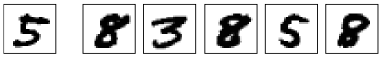

Figure 1. An example of test sample (leftmost) and its five nearest training samples.

For reducing this type of misclassification, it is effective to use the classification method based on comparison between the test sample and the global data distribution of individual categories such as a linear subspace method [5, 6]. In the linear subspace method, data distribution of each class is represented by individual subspaces. The class of a test sample is determined by computing the norm of the projected test sample on the individual subspaces. This approach cannot represent the local data distribution, so the recognition rate decreases when data distribution is not normal distribution.

In order to overcome the difficulties of kNN and the linear subspace method, we propose a new classification method that classifies a test sample by measuring the distance between the test sample and the average patterns, which are calculated using the k-nearest neighbors belonging to individual categories. Furthermore, we show how to apply kernel methods to the proposed method. The performance of the proposed method is verified by experiments with the real-life problem of handwritten digit recognition.

2 Classification by distance between test samples and average patterns of categorical nearest neighbors

In this section, we observe the nature of the k-nearest neighbors of test samples for overcoming the difficulties

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 **IEEE**

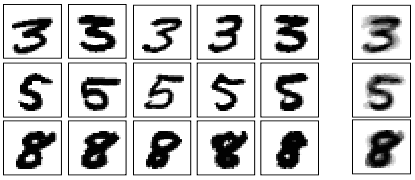

Figure 2. The five training samples closest to the test sample. Only the classes 3, 5 and 8 are shown. From top to bottom, the class 3, 5 and 8. At the right column are the average patterns of each class.

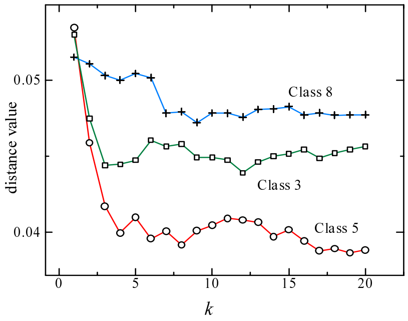

Figure 3. Relationship between the value of k and distance.

found in kNN and the linear subspace method. Figure 2 illustrates the five nearest training samples of each class to the test sample depicted in Figure 1. As shown in Figure 2, they consist of various size and line-thickness images. Note that the training samples for the classes 3 and 8 contain the neighbors that are not similar to the test sample. To evaluate the relationship of the test sample and its neighboring ones, we computed the average patterns using the k-nearest neighbors of each class (see the rightmost in Figure 2). It seems that the average pattern for the class 5 is similar to the test sample, but other average patterns are not. So we measured the distance between the test sample and these average patterns. Figure 3 shows the relationship between the number of k-nearest neighbors and the distance from the test sample to the average patterns. This figure indicates that the average pattern for the class 5 is closest to the test sample, and the distance values of the classes 3 and 8 never drop as low as that of the class 5. In other words, the distance between the test sample and the average pattern of the class 5 becomes smaller than the other classes because the training samples belonging to the class 5 are uniformly distributed around the test sample. In addition, the dissimilar training samples to the test sample are more and more added to the average patterns for classes other than class 5 by increasing of k, so the distance values of these classes never drop as low as that of the class 5.

According to the above observation and discussion, it is expected that a high recognition rate can be achieved by measuring the distance between the test sample and the average patterns instead of counting the number of the labels of k-nearest neighbors such as kNN. Hence, we propose a classifier that outputs the class of a test sample by measuring the distance between the test sample and the average patterns, which are calculated using the k-nearest neighbors of individual categories.

2.1 Formulation

Let $\boldsymbol{x}_i = [x_{i1}, ..., x_{id}]^T \, (i = 1, ..., n)$ be a d-dimensional training sample belonging to the class j, where n is the number of the training samples belonging to the class j. When a test sample $\boldsymbol{q} = [q_1, ..., q_d]^T$ is given, the class of the test sample (denoted by ω) is chosen as

$$\omega = \arg\min_j \left\{ \left\| \frac{1}{k} \sum_{i \in X_j} \boldsymbol{x}_i - \boldsymbol{q} \right\|^2 \right\}, \qquad (1)$$

where X_j is the set of the k-nearest training samples which belong to the class j. The following relationship is established between the individual samples of X_j:

$$||\boldsymbol{x}_1 - \boldsymbol{q}||^2 \leq ||\boldsymbol{x}_2 - \boldsymbol{q}||^2 \leq ... \leq ||\boldsymbol{x}_k - \boldsymbol{q}||^2. \qquad (2)$$

This classification approach employs k as a parameter. In this paper, we call this method *CAP* (classification using Categorical Average Patterns). When $k = 1$, CAP coincides with the nearest neighbor rule (1-NN).

2.2 Kernel CAP

In recent years much research has been conducted on kernel methods (e.g. [7, 8]), to which CAP described above can be applied. When we apply the kernel method to CAP, the class of the test sample is chosen as

$$\omega = \arg\min_j \left\{ \left\| \frac{1}{k} \sum_{i \in X_j} \Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{q}) \right\|^2 \right\}, \qquad (3)$$

where $\Phi(\cdot)$ is a mapping function that maps samples from an input space to a high-dimensional space. We represent an inner product in the high-dimensional space $\langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle$ by an appropriate Mercer kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Hence, the square of the Euclidean distance between the test sample $\boldsymbol{q}$ and the training sample $\boldsymbol{x}_i$ in the high-dimensional space is written as

$$\begin{aligned} d_i &= ||\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{q})||^2 \\ &= \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_i) \rangle - 2\langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{q}) \rangle + \langle \Phi(\boldsymbol{q}), \Phi(\boldsymbol{q}) \rangle \\ &= K(\boldsymbol{x}_i, \boldsymbol{x}_i) - 2K(\boldsymbol{x}_i, \boldsymbol{q}) + K(\boldsymbol{q}, \boldsymbol{q}). \qquad (4) \end{aligned}$$

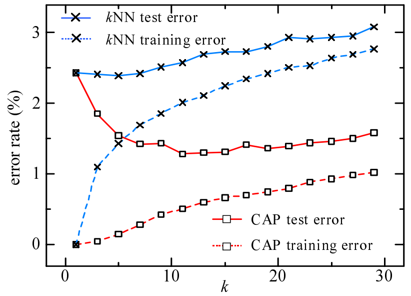

Figure 4. Relationship between the value of k and error rates.

In the same way, the equation (3) can be expanded as

$$\left\| \frac{1}{k}\sum_{i\in X_j}\Phi(\boldsymbol{x}_i)-\Phi(\boldsymbol{q})\right\|^2 \tag{5}$$
$$= \frac{1}{k^2}\sum_{l,m\in X_j}K(\boldsymbol{x}_l,\boldsymbol{x}_m)-\frac{2}{k}\sum_{i\in X_j}K(\boldsymbol{x}_i,\boldsymbol{q})+K(\boldsymbol{q},\boldsymbol{q}).$$

Note that in the equations (4) and (5), the factor $K(\boldsymbol{q},\boldsymbol{q})$ can be ignored, because it is the common term in all classes.

In short, CAP that uses the kernel method is conducted in the following manner. First, $d_i = K(\boldsymbol{x}_i,\boldsymbol{x}_i)-2K(\boldsymbol{x}_i,\boldsymbol{q})$ is calculated for each class, and the k-nearest training samples $\boldsymbol{x}_i(i = 1,...,k)$ are selected for each class. Second, the class of the test sample is determined by measuring the distance between the test sample and the average patterns in the high-dimensional space: $\sum_{l,m\in X_j} K(\boldsymbol{x}_l,\boldsymbol{x}_m)/k^2 - 2\sum_{i\in X_j} K(\boldsymbol{x}_i,\boldsymbol{q})/k$. In this paper, we call this method *KCAP* (Kernel CAP). Throughout this paper we use the Gaussian kernel with width parameter α:

$$K(\boldsymbol{x}_i,\boldsymbol{q}) = e^{-\alpha\|\boldsymbol{x}_i-\boldsymbol{q}\|^2}. \tag{6}$$

3 Experiments

3.1 Experimental results on MNIST

In this section, we show the property of the proposed method using the MNIST dataset. The MNIST dataset consists of 60,000 training and 10,000 test images. For the feature extraction, we use for *peripheral direction contributivity feature* (P-DC) [9, 10]. This feature set represents each image as a 256 dimensional vector.

3.1.1 Influence of the parameter k on error rates

In a first experiment, we examined the relationship between the parameter k and error rates. Figure 4 shows the results of kNN and CAP. The result of KCAP was not included in this figure, because it was almost same as that of CAP. As shown in this figure, the increase of k leads to the increasing of the test and training error of kNN. In contrast, the test error rate of CAP decreases while k is less than or equal to about 10. In addition, the increasing rate against training samples is smaller than that of kNN. Hence, selection of k on CAP is easier than that on kNN.

Table 1. Error rates on MNIST [%].

method	test	training
kNN ($k = 5$)	2.39	1.43
DWkNN ($k = 11$)	2.12	0.12
CLAFIC ($k = 30$)	3.68	3.87
PFM ($\alpha = 5000$)	2.26	0
SVM ($\alpha = 180, C = 10$)	0.92	0.01
CAP ($k = 11$)	1.28	0.50
KCAP ($k = 11$, $\alpha = 70$)	1.27	0.37

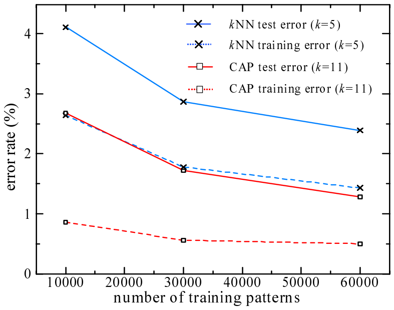

Figure 5. Relationship between the number of training samples and error rates.

Table 1 lists the lowest error rates with the parameter values of *each classifier*: kNN, the Distance-Weighted kNN (DWkNN) [11], the basic linear subspace method *CLAFIC* [5, 6], Potential Function Method (PFM) [12] with the Gaussian potential function (see the equation (6)), Support Vector Machine (SVM) [7], CAP and KCAP. In CLAFIC, the parameter k indicates the dimensionality of subspaces. In SVM, α and C indicate the scale parameter of the Gaussian kernel and the soft margin constant, respectively. For SVM, we used the SVM package, *LIBSVM* [13].

From the above table, we conclude that SVM outperforms all the other investigated techniques and the test error rates of CAP and KCAP significantly are lower than those of kNN, CLAFIC and PFM. Incidentally, some techniques that select the appropriate prototypes by clustering such as LVQ-kNN [14] have a property of a trade-off between recognition rates and the number of prototypes [14]. That is, if we desire a high-accuracy prediction, we should select a large number of prototypes (clusters). Consequently, clustering-based classifiers with a large number of prototypes will approach to the kNN rule, so we did not compare them to the proposed method.

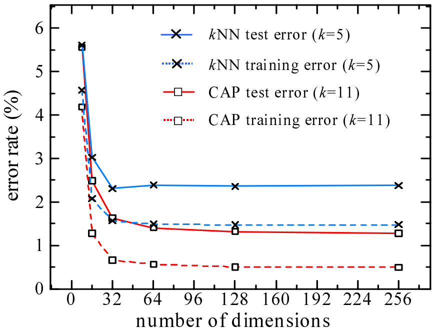

Figure 6. Relationship between the dimensionality of feature vectors and error rates.

3.1.2 Relationship between the number of training samples and error rates

In a next experiment, we examined the influence of the number of training samples on error rates. Figure 5 indicates the changes in error rates when the number of training samples increases from 10,000, 30,000 to 60,000. This experiment showed that the error rates of CAP were lower than those of kNN in all range of the number of training samples. However, no significant difference in the decreasing rate of errors was found between CAP and kNN.

3.1.3 Influence of the dimensionality of feature vectors on error rates

Next, we examined the relationship between the dimensionality of feature vectors and the error rates. In experiments, dimension reduction was applied to the 60000 training samples using the Karhunen-Loéve expansion technique. The changes in error rates were examined with the dimensionality ranging from 8 to 256. Figure 6 shows the results. The result of KCAP was similar to that of CAP, so we did not depict it in this figure. As shown in this figure, CAP achieved lower error rates than kNN across all range. Also note that the test error rate of kNN reached its minimum when the number of dimensions was 32. On the other hand, the test error rate of CAP reached its minimum when the dimensionality was 256. This empirical analysis showed that CAP is effective for processing high-dimensional patterns.

3.2 Experimental results on USPS

In this section, we test the proposed method on the USPS dataset [15]. The USPS dataset consists of fewer training samples than MNIST. In addition, this dataset is more difficult to recognize than MNIST. The USPS consists of 7,291 training and 2,007 test images. In experiments, we used the 256-dimensional P-DC feature vector.

Table 2. Error rates on USPS [%].

method	test	training
kNN ($k=1$)	5.2	0
DWkNN ($k=5$)	4.73	0
CLAFIC ($k=15$)	4.73	1.71
PFM ($\alpha=7000$)	4.98	0
SVM ($\alpha=200, C=1$)	3.84	0.26
CAP ($k=12$)	3.54	0.59
KCAP ($k=12, \alpha=130$)	3.44	0.43

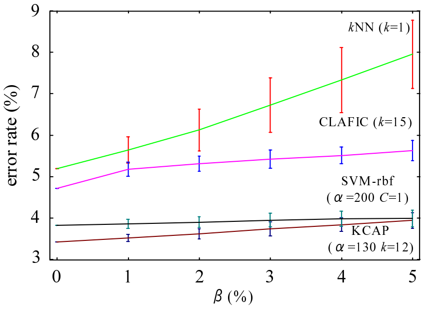

Figure 7. Relationship between error rates and the rate of outliers β.

Table 2 lists the lowest error rates with the parameter values of each classifier. The result showed that the proposed method outperformed all the other investigated techniques. Furthermore, the error rates of KCAP were lower than those of CAP. That is, the use of kernel methods helped improve the recognition performance of CAP.

3.2.1 Robustness Against Outliers

Next, we examined the nature of robustness against outliers of each classifier by randomly replacing the class labels of training samples. The averaged misclassification rates and standard errors of each method by 100 trials are displayed with error bars plot in Figure 7. The horizontal axis denotes the rate of outliers (i.e., the rate of the number of the replaced training samples). The result of CAP and PFM were similar to those of KCAP and CLAFIC respectively, so we did not display them in this figure. The increase of outliers led to the deterioration of averaged misclassification rates and standard errors of kNN, but it did not lead to those of SVM. The increasing rate of the averaged error of KCAP was not so different from that of CLAFIC, but the standard error of KCAP was small as well as that of SVM. This empirical analysis showed that CAP and KCAP were robust to outliers more than kNN and CLAFIC.

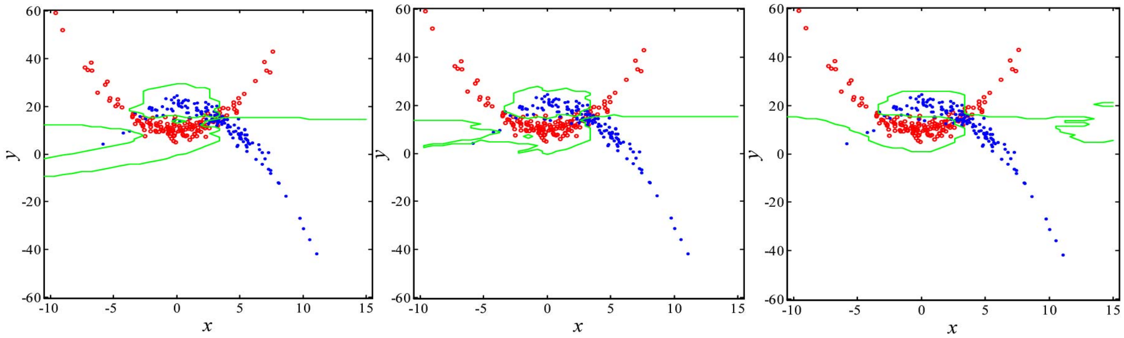

Figure 8. An example of two-dimensional artificial samples and decision boundaries for each method. Left: kNN ($k=5$). Middle: CAP ($k=9$). Right: KCAP ($k=11$, $\alpha=0.08$).

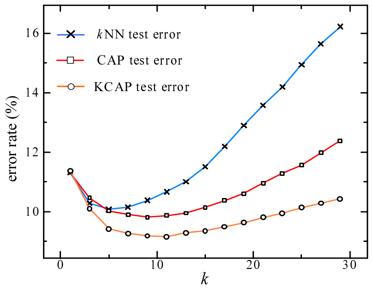

Figure 9. Relationship between test error and the value of k.

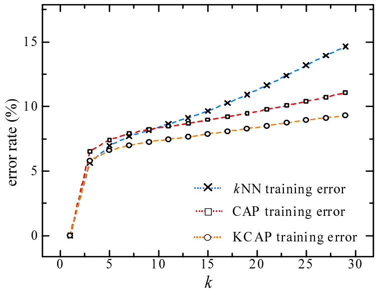

Figure 10. Relationship between training error and the value of k.

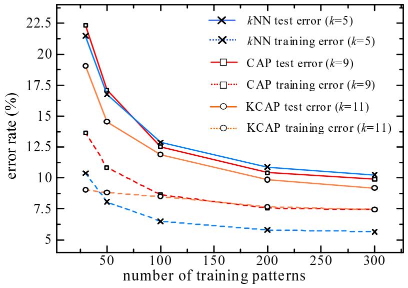

Figure 11. Relationship between the number of training samples and error rates.

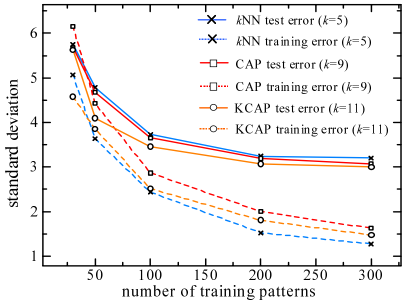

Figure 12. Standard deviations of error rates.

3.3 Experimental results on artificial patterns

Generally, the kNN rule can be applied to patterns of which distribution is nonlinear, but it is not suitable for high-dimensional patterns (e.g., handwritten characters). So CAP and KCAP are compared with kNN using a low-dimensional artificial pattern with nonlinear distribution. First, we created two classes that consist of two-dimensional patterns by the following manner: Let x_1, y_1, x_2 and y_2 be the values randomly sampled from normal distributions which have the mean and variance (μ, σ^2) of $(0, 10)$, $(10, 5)$, $(3, 10)$ and $(20, 5)$. An equal number of patterns were generated for the two-dimensional patterns belonging to Class 1 (ω_1) and Class 2 (ω_2) given by $(x_1, 0.5x_1^2 + y_1) \in \omega_2$ and $(x_2, -0.5x_2^2 + y_2) \in \omega_2$ [16]. Figure 8 shows 300 training samples and the decision boundaries obtained by kNN, CAP and KCAP. The two-class samples are indicated by dots, and the decision boundaries are indicated by lines. As seen in this figure, the decision boundary of KCAP is the smoothest one.

Next, the relationship between the parameter k and error rates was conducted with the number of test and training samples fixed at 100 and 300 respectively. Each test was independently repeated 100 times under each test condition, and the mean and variance of classification error rates were calculated. Figure 9 and 10 show the averages of test error rates and training ones for each value of k. As shown in those figures, the large value of k led to increase of the error rates for kNN. In contrast, the error rates of CAP and KCAP did not increase such as kNN. Hence, selection of k on CAP and KCAP is easier than that on kNN.

Finally, classification tests were conducted with the number of test samples fixed at 100 and the number of training samples varying from 30 to 300. The parameters of each method were selected based on the above experiment (kNN: k=5, CAP: k=9, KCAP: $k = 11$, $\alpha = 0.08$). Each test was independently repeated 100 times under each test condition, and the mean and standard deviation of error rates were calculated. Figure 11 shows the mean of error rates for each number of training samples and for each method. Two curved lines having the same dot indicate the mean of test error rate (solid lines) and training error rate (dashed lines) in the same method. This experiment showed that the error rates of CAP and KCAP were lower than those of kNN in any range of the number of training samples. It should be noted that the test error rate of KCAP was lower than those of kNN and CAP in all range of the number of training samples. That is, the use of kernel methods helped improve the classification performance of CAP. Figure 12 shows the standard deviation of error rates for each number of training samples. This experiment showed that standard deviations of the test error rates of CAP and KCAP was lower than of kNN in all range of the number of training samples. Hence, we can conclude that CAP and KCAP outperform the kNN rule and have a high-generalization performance.

4 Conclusions

This paper has presented the algorithm that outputs the class of a test sample by measuring the distance between the test sample and the average patterns, which are calculated using the k-nearest neighbors belonging to individual classes. It was verified by the experiments using handwritten digit patterns and a low-dimensional artificial one that the proposed method achieved higher recognition rates than other nonparametric methods such as kNN and linear subspace methods.

The computational cost of the proposed method is high as well as that of kNN, because they are the methods of complete storage (i.e., the method that stores all training samples in systems). For instance, the computational costs of kNN and CAP are approximately $O((cn)^2d)$ and $O(cn^2d)$ respectively, where c is the number of classes. However, the proposed method can measure the distance between test samples and the average patterns on individual classes independently. Hence, it is able to reduce the number of candidate classes by performing a rough classification. In addition, subspace methods and neural networks such as SVM may require recalculating subspaces and relearning support vectors when training samples are added, but the proposed method only needs to add them. That is, there is no need to reconstruct systems when training samples are added.

In short, the proposed method includes the following advantages: 1) CAP and KCAP can achieve lower error rates than other nonparametric methods such as kNN and subspace methods. 2) The proposed method can be applied to high-dimensional patterns. Hence, the recognition rate of CAP can be improved by employing kernel methods. 3) We can implement CAP and KCAP easily because of its simple algorithms. 4) There is no need to reconstruct systems when samples are added.

References

[1] R.O. Duda *et al*. *Pattern classification (2nd edition)*. John Wiley and Sons, 2001.

[2] K. Fukunaga. *Introduction to statistical pattern recognition*. Academic press, Boston, 2 edition, 1990.

[3] K. Fukunaga. Bias of nearest neighbor error estimation. *IEEE Trans. PAMI*, 9(1):103–112, 1987.

[4] Y. LeCun *et al*. Gradient-based learning applied to document recognition. *Intell. Signal Process.*, 306–351, 2001.

[5] S. Watanabe and N. Pakvasa. Subspace method in pattern recognition. *Proc. 1st Int. J. Conf on Pattern Recognition*, 2–32, 1973.

[6] E. Oja. *Subspace methods of pattern recognition*. Research Studies Press, 1983.

[7] V. Vapnik. *Statistical learning theory*. John Wiley and Sons, 1998.

[8] K.-R. Müller *et al*. An introduction to kernel-based learning algorithms. *IEEE Trans. on Neural Networks*, 12(2):181–201, Mar. 2001.

[9] N. Hagita *et al*. Handprinted chinese characters recognition by peripheral direction contributivity feature. *Trans. IEICE*, J66-D-II(10):1185–1192, Oct. 1983.

[10] C.L. Liu *et al*. Handwritten digit recognition: Benchmarking of state-of-the-art techniques. *Pattern Recognition*, 36(10):2271–2285, 2003.

[11] S.A. Dudani. The distance-weighted k-nearest neighbor rule. *IEEE trans. on systems, man and cybernetics*, SMC-8(4):311–313, 1978.

[12] M.A. Aizerman *et al*. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

[13] C.C. Chang and C. J. Lin. LIBSVM: A library for support vector machines. 2001. Available from `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[14] Q. Xie, C.A. Laszlo and R.K. Ward. Vector quantization technique for nonparametric classifier design. *IEEE Trans. PAMI*, 15(12):1326–1330, 1993.

[15] Y. LeCun *et al*. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[16] E. Maeda and H. Murase. Kernel-based nonlinear subspace method for pttern recognition. *Systems and Computers in Japan*, 33(1):38–52, 2002.