Pattern Recognition Using Average Patterns of Categorical k-Nearest Neighbors

Seiji Hotta, Senya Kiyasu, and Sueharu Miyahara
Department of Computer and Information Sciences, Nagasaki University, Japan
{hotta, kiyasu, miyahara}@cis.nagasaki-u.ac.jp

Abstract

The typical nonparametric method of pattern recognition "k-nearest neighbor rule (kNN)" is carried out by counting the labels of k-nearest training samples to a test sample. This method collects the k-nearest neighbors without taking into account a class, and it outputs the class of the test sample by using only the labels of neighborhoods. 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 the k-nearest neighbors belonging to individual classes. A kernel method can be applied to this classifier for improving recognition rates. The performance of the proposed method is verified by experiments with benchmark data sets.

1. Introduction

The nonparametric method of pattern recognition k-*nearest neighbor rule* (kNN) has been implemented on pattern recognition systems because of its good performance and simple algorithm. In kNN, test samples are classified by counting the labels of k-closest training samples [1, 2]. This approach includes the following features: 1) It has been proved that the error rate of kNN approaches the Bayes error when both the number of training samples and the value of k are infinite. 2) We can design the classifier by kNN even if training samples are few. 3) We can implement kNN when classes are 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 dimensionality of feature vectors is large [3]. For example, Figure 1 shows the example of a test sample from the MNIST dataset and the five nearest training samples, which are evaluated using the Euclidean distance. Because the selected five training samples include the three samples of class 8, so the test sample has been misclassified to '8'.

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 classes such as Linear Subspace Method (LSM) [4]. In LSM, the data distribution of each class is represented by subspaces. The class of a test sample is determined by computing the norm of the projected test sample on the subspace. The weakness of this method is that it cannot represent the local distribution of patterns, so recognition rates decrease when the data distribution is not normal distribution.

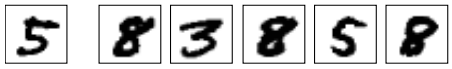

Figure 1. A test sample (leftmost) and its five nearest training samples.

This paper presents an alternative approach similar to kNN 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 of each class. This approach can be easily implemented due to its simple algorithm and can overcome the difficulty of kNN that recognition rates deteriorate when the dimensionality of feature vectors increases. In addition, we show how to apply kernel methods to the proposed method. The performance of the proposed method is verified by experiments with handwritten digit patterns and the benchmark data sets of binary classification problems.

2. Classification using average patterns of categorical k-nearest neighbors

In this section, we observe the nature of the k-nearest neighbors of a test sample for overcoming the difficulties found in kNN and LSM. Figure 2 illustrates the five nearest training samples of each class (only classes 3, 5 and 8 are shown). They consist of various size and line-thickness images. Note that the training samples of the classes 3 and 8 contain the patterns that are not similar to the test sample. To evaluate the rela-

Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)
1051-4651/04 $ 20.00 IEEE

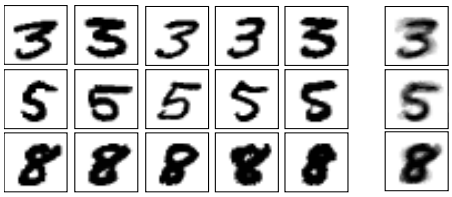

Figure 2. The five nearest training samples of each class. At the right column are the average samples of them.

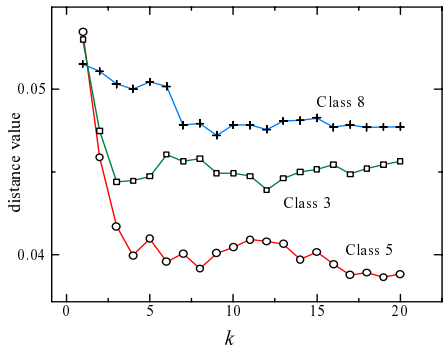

Figure 3. Relation between k and distance.

tionship of the test sample and its neighboring ones, we computed the average patterns of the k-nearest neighbors for 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. Hence, we measured the distance between the test sample and the 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 values of distance for 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.

According to the above observation and discussion, 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 each class.

2.1. Formulation

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

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

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

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

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. [5, 6]), to which CAP described above can be applied. When we apply the kernel method to CAP, the class of the test sample is determined by

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

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 $\boldsymbol{x}_i^j$ in the high-dimensional space is written as

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

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

$$\frac{1}{k^2} \sum_{l,m \in X_j} K(\boldsymbol{x}_l^j, \boldsymbol{x}_m^j) - \frac{2}{k} \sum_{i \in X_j} K(\boldsymbol{x}_i^j, \boldsymbol{q}) + K(\boldsymbol{q}, \boldsymbol{q}).$$

In this equation and the equation (4), the factor $K(\boldsymbol{q}, \boldsymbol{q})$ can be ignored, because it is the common term in all classes. In short, CAP that uses kernel methods is conducted in the following manner: First,

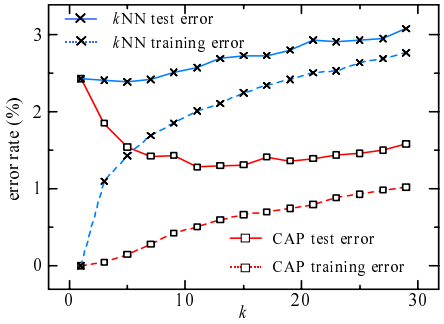

Figure 4. Relation between k and errors.

Table 1. Results on MNIST.

method	test [%]	training [%]	time [s]
kNN ($k = 5$)	2.39	1.43	0.73
DWkNN ($k = 4$)	2.16	0	0.78
CLAFIC ($k = 30$)	3.68	3.87	0.002
CAP ($k = 11$)	1.28	0.50	0.71
KCAP ($k = 11$)	1.27	0.37	0.87

$d_i^j = K(\boldsymbol{x}_i^j, \boldsymbol{x}_i^j) - 2K(\boldsymbol{x}_i^j, \boldsymbol{q})$ is calculated for each class, and the k-nearest training samples $\boldsymbol{x}_i^j (i = 1, \ldots, k)$ 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 of each class in a high-dimensional space: $\sum_{l,m \in X_j} K(\boldsymbol{x}_l^j, \boldsymbol{x}_m^j)/k^2 - 2\sum_{i \in X_j} K(\boldsymbol{x}_i^j, \boldsymbol{q})/k$. In this paper, we call this method *KCAP* (Kernel CAP). Throughout this paper we use the Gaussian kernel with width parameter α: $K(\boldsymbol{x}_i^j, \boldsymbol{q}) = \exp(-\alpha \|\boldsymbol{x}_i^j - \boldsymbol{q}\|^2)$.

3. Experiments

We tested the proposed method on the handwritten digit datasets MNIST and USPS. Firstly, the properties of the proposed method were examined using the MNIST dataset. The MNIST dataset consists of 60,000 training and 10,000 test images. For feature extraction, we use for peripheral direction contributivity feature [7, 8] with 256 dimensions.

3.1 Influence of parameter k on error rates

First, the relationship between k and error rates was examined. 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 the figure, the error rates of kNN against test and training samples increase as the k increases. In contrast, the test error of CAP decreases while k is less than or equal to about 10. In addition, the increasing rate of the training error of CAP is smaller than that of kNN. Hence, selection of k on CAP is easier than that on kNN.

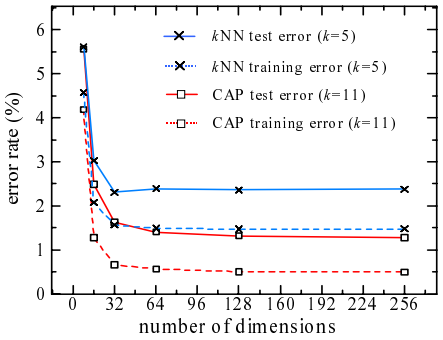

Figure 5. Relationship between the number of dimensions and error rates.

Table 2. Results on USPS.

method	test [%]	training [%]	time [s]
kNN ($k = 1$)	5.2	0	0.08
DWkNN ($k = 3$)	5.13	0	0.09
CLAFIC ($k = 15$)	4.73	1.71	0.001
CAP ($k = 12$)	3.54	0.59	0.08
KCAP ($k = 12$)	3.44	0.43	0.13

Table 1 lists the lowest error rates and average query-time (i.e., execution time per one test sample) of *each method*: kNN, Distance-Weighted kNN (DWkNN), the basic LSM method *CLAFIC* [4], CAP and KCAP. For CLAFIC, the k indicates the dimensionality of subspaces. For KCAP, $\alpha = 70$ was used for the parameter of the Gaussian kernel. The average query-time was obtained using a Pentium 3.06 GHz with 1.0 GB of RAM memory. From the above table, we conclude that the computational costs of CAP and KCAP are high, but the error rates of them significantly are lower than those of other methods.

3.2. Influence of dimensionality on error rates

Next, the relationship between the dimensionality of features and error rates was examined. In this experiment, dimension reduction was applied to the training set (60000) using the Karhunen-Loéve expansion technique. The variation in error rates was examined with the dimensionality ranging from 8 to 256. Figure 5 shows the results. As shown in the figure, CAP achieved lower error rates than kNN across all the range. Also note that the test error rate of kNN reaches its minimum when the number of dimensions is 32, while that of CAP decreases after it. This empirical analysis showed that CAP is effective for processing high-dimensional patterns.

3.3. Experimental results on USPS

Secondly, we tested the proposed method on the USPS dataset. This dataset is more difficult to recog-

Table 3. Error rates [%].

dataset	kNN	CAP	KCAP
Banana	11.3 ± 0.6	11.8 ± 0.5	$\mathbf{10.7 \pm 0.5}$
B.Cancer	$\mathbf{25.3 \pm 4.0}$	26.5 ± 4.5	25.9 ± 4.4
Diabetes	25.1 ± 1.7	24.5 ± 1.8	$\mathbf{23.7 \pm 1.9}$
German	25.2 ± 2.3	24.6 ± 2.3	$\mathbf{24.4 \pm 2.5}$
Heart	$\mathbf{15.7 \pm 3.3}$	15.9 ± 3.4	16.1 ± 3.5
Image	3.4 ± 0.5	$\mathbf{3.3 \pm 0.6}$	$\mathbf{3.3 \pm 0.6}$
Ringnorm	35.0 ± 1.4	12.0 ± 0.8	$\mathbf{1.4 \pm 0.1}$
F.Sonar	34.8 ± 1.9	$\mathbf{34.4 \pm 1.7}$	$\mathbf{34.4 \pm 1.7}$
Splice	26.2 ± 2.1	13.5 ± 0.8	$\mathbf{12.9 \pm 0.7}$
Thyroid	4.4 ± 2.2	4.4 ± 2.2	$\mathbf{4.2 \pm 2.1}$
Titanic	$\mathbf{22.8 \pm 1.1}$	23.1 ± 1.9	22.8 ± 1.5
Twonorm	2.5 ± 0.2	$\mathbf{2.4 \pm 0.1}$	$\mathbf{2.4 \pm 0.1}$
Waveform	10.7 ± 1.0	10.2 ± 0.5	$\mathbf{9.9 \pm 0.6}$

nize than MNIST. The USPS dataset consists of 7,291 training and 2,007 test images. Table 2 lists the lowest error rates and average query-time of each method. For KCAP, $\alpha = 130$ was used for the parameter of the Gaussian kernel. The result showed that the proposed method outperformed all the other investigated techniques. Furthermore, the errors of KCAP were lower than those of CAP, i.e., CAP with kernel methods improved its classification abilities.

3.4. Experimental results on other benchmarks

Finally, we tested the proposed method on the benchmark data of binary classification problems (see [9, 10] for more details). Table 3 lists the lowest average test error rates and its standard deviations. Due to lack of space, we showed the results of kNN, CAP and KCAP only (for comparison to other methods cf. [9, 10]). The best values of each set are depicted in boldface type. This table showed that the results of CAP and KCAP are better than those of kNN in many cases. In addition, the error rates of KCAP were lower than those of CAP, i.e., the use of kernel methods helped improve the recognition performance of CAP.

4. Conclusions

This paper has presented an algorithm that outputs the class of a test sample by measuring the distance between the test sample and the average patterns of the k-nearest neighbors of each class. It was verified by the experiments using benchmark data sets that the proposed method achieved higher recognition rates than other nonparametric method such as kNN and LSM.

The computational cost of the proposed method is high as well as that of kNN. For instance, the computational costs of kNN and CAP are approximately $O((cn_j)^2 d)$ and $O(cn_j{}^2 d)$ respectively, where c is the number of classes. However, the proposed method can compute the distance between test samples and the average patterns of individual classes independently. Hence, it is able to reduce the number of candidate classes by performing rough classification. In addition, when training samples are added, LSM and neural networks such as Support Vector Machine require recalculating the subspace and re-learning support vectors respectively, but the proposed method only needs to add them. In other words, there is no need to reconstruct systems when training samples are added.

In short, the proposed method includes the following advantages: 1) Our methods can achieve lower error rates than other nonparametric methods such as kNN and LSM. 2) The proposed method can achieve low error rates even if the dimensionality of feature vectors is large. Hence, it is possible to improve recognition rates by employing kernel methods to CAP. 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.

Acknowledgments This research was supported by the Telecommunications Advancement Foundation.

References

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

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

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

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

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

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

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

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

[9] S. Mika *et al.* Fisher discriminant analysis with kernels. *Neural Networks for Signal Processing*, 36(10):41–48, 1999.

[10] G. Rätsch *et al.* Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, Mar. 2001.