

- [7] A. Guezic and N. Ayache, "Smoothing and Matching of 3D Spatial Curves, *Int'l J. Computer Vision*, vol. 12, no. 1, pp. 79-104, 1994.
- [8] B.K.P. Horn, *Robot Vision*, Cambridge, Mass.: MIT Press, 1986.
- [9] E. Kishon, T. Hastie and H. Wolfson, "3D Curve Matching Using Splines, *J. Robotic Systems*, vol. 8, no. 6, pp. 723-743, 1991.
- [10] P. Maillot, "Using Quaternions For Coding 3D Transformations," *Graphic Gems*, A.S. Glassner, ed. New York: Academic Press, pp. 498-515, 1990.
- [11] F. Mokhtarian and A. Mackworth, "Scale-Based Description and Recognition of Planar Curves and Two Dimensional Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 34-43, 1986.
- [12] T. Pajdala and L. Van Gool, "Matching of 3D Curves Using Semi-Differential Invariants, *Proc. Fifth Int'l Conf. Computer Vision*, pp. 390-395, Cambridge, Mass., June, 1995.
- [13] B. Parsi, A. Margalit, and A. Rosenfeld, "Matching General Polygonal Arcs," *Computer Vision, Graphics and Image Processing: Image Understanding*, vol. 53, pp. 227-234, 1991.
- [14] J.T. Schwartz and M. Sharir, "Identification of Partially Obscured Objects in Two and Three Dimensions by Matching Noisy Characteristic Curves," *Int'l J. Robotics Res.*, vol. 6, no. 2, pp. 29-44, 1987.
- [15] K. Shoemake, "Quaternions and  $4 \times 4$  Matrices," *Graphic Gems II*, J. Arvo, ed. New York: Academic Press, pp 351-354, 1991.

## A Bootstrap Technique for Nearest Neighbor Classifier Design

Yoshihiko Hamamoto, *Member, IEEE Computer Society*,  
Shunji Uchimura, and Shingo Tomita

**Abstract**—A bootstrap technique for nearest neighbor classifier design is proposed. Our primary interest in designing a classifier is in small training sample size situations. Conventional bootstrapping techniques sample the training samples with replacement. On the other hand, our technique generates bootstrap samples by locally combining original training samples. The nearest neighbor classifier is designed on the bootstrap samples and is tested on the test samples independent of training samples. The performance of the proposed classifier is demonstrated on three artificial data sets and one real data set. Experimental results show that the nearest neighbor classifier designed on the bootstrap samples outperforms the conventional  $k$ -NN classifiers as well as the edited 1-NN classifiers, particularly in high dimensions.

**Index Terms**—Bootstrap, nearest neighbor classifier, error rate, peaking phenomenon, small training sample size, high dimensions, outlier.

### 1 INTRODUCTION

IN the nonparametric framework the nearest neighbor (1-NN) approach was first introduced by Fix and Hodges [12], [13] and later studied by Cover and Hart [1]. Cover and Hart [1] show that the error for the 1-NN classifier is bounded by twice the Bayes error when an infinite number of samples is available. However, in practice, we never have an infinite number of samples, and, due to the finite sample size, the 1-NN estimates have large biases and variances. Our primary interest in designing a classifier is in small training sample size situations. Fukunaga and Hummels [2] show that the 1-NN estimates may be severely biased even for the large sample size if the dimensionality of the data is large. This is a major obstacle in many practical situations where the ratio of the training sample size to the dimensionality is small. Both Fukunaga and Hummels [7] and Chandrasekaran and Jain [22] recommend a decision threshold,  $t$ , to take into account the bias in density estimation. Then, the decision rule can be given by

Classify  $x$  into class  $\omega_k$  if

$$\hat{p}(x|\omega_k) > \hat{p}(x|\omega_j) + t \text{ for all } j = 1, \dots, m; j \neq k$$

where  $\hat{p}(x|\cdot)$  denotes the estimated density and  $m$  is the number of classes. Fukunaga and Hummels [7] show that the proper selection of the decision threshold is an extremely important procedure for  $k$ -NN error estimates. However, it is difficult to determine the optimal value of the threshold because of its complexity. We wish to learn as much as possible about the pattern distributions that a given training set may have. One possible way of doing this is to generate an artificial training set from the original training set. This technique is called the bootstrap method, and the artificial samples are called the bootstrap samples [3]. The bootstrap method has been successfully applied to error estimation [4], [5], [6], [21]. In this paper, we

- Y. Hamamoto and S. Tomita are with the Faculty of Engineering, Yamaguchi University, Ube, 755 Japan.  
E-mail: hamamoto@csse.yamaguchi-u.ac.jp.
- S. Uchimura is with the Oshima National College of Maritime Technology, Oshima-cho, 742-21, Japan

Manuscript received Jan. 20, 1995; revised Aug. 23, 1996. Recommended for acceptance by R. Duin.

For information on obtaining reprints of this article, please send e-mail to: transpami@computer.org, and reference IEEECS Log Number P96090..

propose a bootstrap technique for the 1-NN classifier design. Experimental results show that the 1-NN classifier based on our bootstrap technique outperforms the conventional  $k$ -NN classifiers as well as edited 1-NN classifiers, particularly in high-dimensional spaces.

## 2 BOOTSTRAP TECHNIQUES

Bootstrapping is similar to other resampling schemes such as cross-validation and jackknifing. There could be many possible ways to generate bootstrap samples. We will focus on the generation of the bootstrap samples. Let  $X_{N_i} = \{x_{i1}^i, x_{i2}^i, \dots, x_{iN_i}^i\}$  be an original training set from class  $\omega_i$ . We now consider four bootstrap techniques of generating a bootstrap set  $X_{N_i}^B = \{x_{i1}^b, x_{i2}^b, \dots, x_{iN_i}^b\}$  of size  $N_i$ .

### 2.1 Bootstrapping I

- 1) Select a sample  $x_{k_0}^i$  from  $X_{N_i}$  randomly.
- 2) Find the  $r$  nearest neighbor samples  $x_{k_1}^i, x_{k_2}^i, \dots, x_{k_r}^i$  of  $x_{k_0}^i$ , using the Euclidean distance.
- 3) Compute a bootstrap sample  $x_{i1}^b = \sum_{j=0}^r w_j x_{k_j}^i$  where  $w_j$  is a weight. The weight  $w_j$  is given by

$$w_j = \frac{\Delta_j}{\sum_{c=0}^r \Delta_c}, \quad 0 \leq j \leq r$$

where  $\Delta_j$  is chosen from a uniform distribution on  $[0, 1]$ . Note that  $\sum_{j=0}^r w_j = 1$ .

- 4) Repeat steps 1, 2, and 3,  $N_i$  times.

In the ordinary bootstrap presented by Efron [3], the bootstrap set of size  $N_i$  is a set  $\{x_{i1}^*, x_{i2}^*, \dots, x_{iN_i}^*\}$  randomly selected from the original training set  $X_{N_i}$  with replacement. In Bootstrapping I, Efron's bootstrap set is needed. In order to verify the necessity of Efron's bootstrap set, the following bootstrapping method is considered:

### 2.2 Bootstrapping II

- 1) Select one sample  $x_{k_0}^i$  from  $X_{N_i}$ .
- 2) Find the  $r$  nearest neighbor samples  $x_{k_1}^i, x_{k_2}^i, \dots, x_{k_r}^i$  of  $x_{k_0}^i$ , using the Euclidean distance.
- 3) Compute a bootstrap sample  $x_{i1}^b = \sum_{j=0}^r w_j x_{k_j}^i$ , where  $w_j$  is a weight. The weight  $w_j$  is given by

$$w_j = \frac{\Delta_j}{\sum_{c=0}^r \Delta_c}, \quad 0 \leq j \leq r$$

where  $\Delta_j$  is chosen from a uniform distribution on  $[0, 1]$  and  $\sum_{j=0}^r w_j = 1$ .

- 4) Repeat steps 1, 2, and 3, until all  $N_i$  samples are selected.

In Step 1 of Bootstrapping II, the samples are chosen so that no sample is selected more than once. In both Bootstrappings I and II, the original training samples are linearly combined by using random weights. Next, in order to verify the necessity of the random weights, the following bootstrapping method is considered:

### 2.3 Bootstrapping III

- 1) Select a sample  $x_{k_0}^i$  from  $X_{N_i}$  randomly.
- 2) Find the  $r$  nearest neighbor samples  $x_{k_1}^i, x_{k_2}^i, \dots, x_{k_r}^i$  of  $x_{k_0}^i$  using the Euclidean distance.
- 3) Compute a bootstrap sample  $x_{i1}^b = \frac{1}{r+1} \sum_{j=0}^r x_{k_j}^i$ .
- 4) Repeat steps 1, 2, and 3,  $N_i$  times.

In Step 3 of Bootstrapping III, a bootstrap sample is given as a local sample mean. Finally, a bootstrapping method without either Efron's bootstrap set or random weights is considered.

### 2.4 Bootstrapping IV

- 1) Select one sample  $x_{k_0}^i$  from  $X_{N_i}$ .
- 2) Find the  $r$  nearest neighbor samples  $x_{k_1}^i, x_{k_2}^i, \dots, x_{k_r}^i$  of  $x_{k_0}^i$ , using the Euclidean distance.
- 3) Compute a bootstrap sample  $x_{i1}^b = \frac{1}{r+1} \sum_{j=0}^r x_{k_j}^i$ .
- 4) Repeat Steps 1, 2, and 3, until all  $N_i$  samples are selected.

In Step 1 of Bootstrapping IV, the samples are chosen so that no sample is selected more than once.

The most important difference between our bootstrapping and the ordinary bootstrapping presented by Efron [3] lies in the manner in which bootstrap samples are generated. In our bootstrapping, the bootstrap samples are created (not selected) by locally combining the original training samples. Chernick et al. [5] point out that one of the problems with the ordinary bootstrapping is the discreteness of the empirical distribution function. The use of our bootstrap samples  $x_{ij}^b$ s may give a smoothing of the distribution of the training samples. In our bootstrap approach, the problem is in determining the number of near neighbors,  $r$ , which will be studied in Section 4.

Now we describe the 1-NN classifier with  $X_{N_i}^B$  ( $1 \leq i \leq m$ ). Let  $x_{kj}^b \in X_{N_i}^B \cup \dots \cup X_{N_m}^B$  be the nearest neighbor of a test pattern  $x$ .

Then the test pattern  $x$  is classified to the class  $\omega_k$  represented by the nearest neighbor  $x_{kj}^b$ . That is,

Classify  $x$  into class  $\omega_k$  if

$$\min_{x_{kj}^b \in X_{N_k}^B} \{d(x, x_{kj}^b)\} \leq \min_{x_{ij}^b \in X_{N_i}^B} \{d(x, x_{ij}^b)\} \text{ for all } i \neq k \quad (1)$$

where  $d(a, b)$  denotes the Euclidean distance between  $a$  and  $b$ . On the other hand, the conventional 1-NN classifier is given as follows:

Classify  $x$  into class  $\omega_k$  if

$$\min_{x_j^i \in X_{N_k}} \{d(x, x_j^i)\} \leq \min_{x_j^i \in X_{N_i}} \{d(x, x_j^i)\} \text{ for all } i \neq k. \quad (2)$$

The error rate is the most effective measure of the performance of a classifier. In order to predict the future performance of a classifier accurately, the independence between the training and test sets must be maintained. In addition, a large test sample should be used. When using artificial data, we estimated the error of classifiers by using 5,000 test samples per class independently generated from distributions of the training samples. It follows that the test samples are also independent of the bootstrap samples. Note that the estimated error rate is a random variable. Thus, it is preferable to repeat the experiment several times independently. Each experiment with artificial data involves 100 independent trials.

## 2.5 Experiment 1

The purpose of Experiment 1 is to compare the performance of four bootstrap techniques in small training sample size situations. This experiment is based on two data sets (*I-I* and nonnormal data sets). The *I-I* and nonnormal data sets were used in [7], [8] to study the behavior of the *k*-NN classifier in finite sample conditions. We briefly describe the above artificial data sets.

### 2.5.1 The *I-I* Data Set

First, we describe the *I-I* data set. The available samples were independently generated from *n*-dimensional normal distributions  $N(\mu_i, \Sigma_i)$  with the following parameters:

$$\mu_1 = [0, \dots, 0]^T, \mu_2 = [\mu, 0, \dots, 0]^T, \Sigma_1 = I_n, \Sigma_2 = I_n$$

where  $\mu_1$  is the *n*-dimensional zero vector and  $I_n$  is the  $n \times n$  identity matrix. In the data set, both  $\Sigma_1$  and  $\Sigma_2$  are  $I_n$ . The value of  $\mu$  controls the overlap between the two distributions. We used  $\mu = 2.56$ , which gives the Bayes error of 10%. Even when the dimensionality of the data changes, the Bayes error stays the same for a fixed  $\mu$ .

### 2.5.2 The Nonnormal Data Set

Next, we describe a nonnormal data set, which is from Experiment 7 on page 335 of [8]. Each class consists of two normal distributions as follows:

$$p_1(x) = \frac{1}{2} N(\mu_{11}, I_n) + \frac{1}{2} N(\mu_{12}, I_n)$$

$$p_2(x) = \frac{1}{2} N(\mu_{21}, I_n) + \frac{1}{2} N(\mu_{22}, I_n)$$

where

$$\mu_{11} = [0, 0, \dots, 0]^T, \mu_{12} = [6.58, 0, \dots, 0]^T$$

$$\mu_{21} = [3.29, 0, \dots, 0]^T, \mu_{22} = [9.87, 0, \dots, 0]^T$$

Even when the dimensionality of the data changes, the Bayes error of this data set is 7.5%. The following experiment was conducted.

Data:	<i>I-I</i> and nonnormal
Dimensionality:	$n = 8$
No. of training samples:	$N_1 = N_2 = 8, 16, 32$
No. of test samples:	5,000 per class
Classifiers:	the 1-NN classifier
Bootstrappings:	I, II, III, IV and Efron
Values of $r$ :	$N/16, N/8, N/4, N-1$

For comparison, the performance of the 1-NN classifier with Efron's bootstrap set is presented. In general, it is recommended that the ratio of the training sample size to the dimensionality is large. Jain and Chandrasekaran [10] point out that the number of training samples per class should be at least five to ten times the dimensionality. As previously mentioned, we are mainly interested in the practical situations where the ratio of the training sample size to the dimensionality is small. Therefore, in Experiment 1, the values of the ratio ranged from one to four. All experimental results which were derived from a Monte Carlo analysis are shown in Table 1 and Table 2. Experimental results showed no clear preference between four bootstrap techniques as used here. All techniques yielded very comparable error rates for each  $r$ . From Table 2, we see that the value of  $r$  can have a strong effect on results.

## 2.6 Experiment 2

The purpose of Experiment 2 is to compare the performance of four bootstrap techniques in high-dimensional spaces. In this experiment, we used the Ness data set [9]. This data set was used in [9] to study the performance of the Parzen, linear and quadratic classifiers in high dimensions.

### 2.6.1 The Ness Data Set

The available samples were independently generated from *n*-dimensional normal distributions  $N(\mu_i, \Sigma_i)$  with the following parameters:

$$\mu_1 = [0, \dots, 0]^T, \mu_2 = [\Delta/2, 0, \dots, 0, \Delta/2]^T$$

$$\Sigma_1 = I_n, \Sigma_2 = \begin{pmatrix} I_{n/2} & O \\ O & \frac{1}{2} I_{n/2} \end{pmatrix}$$

where  $\Delta$  is the Mahalanobis distance between class  $\omega_1$  and class  $\omega_2$ . Note that in this data set, both the mean vectors and covariance matrices differ. The Bayes error varies depending on the value of  $\Delta$  as well as  $n$ . The following experiment was conducted.

Data:	Ness ( $\Delta = 2, 4, \text{ and } 6$ )
Dimensionality:	$n = 50$
No. of training samples:	$N_1 = N_2 = 100$
No. of test samples:	5,000 per class
Classifiers:	the 1-NN classifier
Bootstrappings:	I, II, III, IV, and Efron
Values of $r$ :	$N/16, N/8, N/4, N-1$

Note that by varying the value of  $\Delta$ , we can control the Bayes error. That is, the Bayes error decreases with increase in  $\Delta$ . By varying the value of  $\Delta$ , we can observe how bootstrap techniques perform as a function of the Bayes error. Hence, we used this data set. The results are shown in Table 3. Again, Bootstrappings I, II, III, and IV gave very similar results for each  $r$ , regardless of the Bayes error. The 1-NN classifier with Efron's bootstrap set performed poorly.

In Experiments 1 and 2, no improvement was observed by using Efron's bootstrap set and random weights. Thus, we think that the use of Efron's bootstrap set and random weights is not essential. Rather, the idea of locally combining the original training samples is important. That is, local sample means, which act as representatives, play an important role in applying a bootstrap technique to the 1-NN classifier design. Therefore, we will address only Bootstrapping IV, which is the simplest one.

In Bootstrapping IV, when  $r = 0$ , the proposed classifier corresponds to the conventional 1-NN classifier. By increasing the value of  $r$ , the nonparametric classifier is transformed into a parametric one. That is, when  $r = N_i - 1$ , our classifier becomes the Euclidean distance classifier. By varying the value of  $r$ , the parametric and nonparametric classifiers may be considered in a unified framework.

## 3 COMPARATIVE STUDY

In this section, we evaluate the performance of the 1-NN classifier with our bootstrap samples, compared to those of *k*-NN classifiers and edited 1-NN classifiers.

### 3.1 Experiment 3

In general, when a fixed number of samples is used to design a classifier, the error of the classifier tends to increase as the dimensionality of the data gets large. This topic is called the peaking phenomenon [20]. In particular, the peaking phenomenon of the 1-3 NN classifier is known to be more severe than other parametric classifiers such as Fisher's linear and quadratic classifiers [7], [8]. Thus, it is widely believed that the value of  $N_i$  needed to achieve a given recognition accuracy would be prohibitively large, when the dimensionality of the data is high.

The purpose of Experiment 3 is to compare the proposed classifier with *k*-NN classifiers in high-dimensional spaces. First, the following experiment was performed.

TABLE 1  
COMPARISON OF BOOTSTRAP TECHNIQUES  
ON THE I-I DATA SET IN TERMS OF THE ERROR RATE(%)

		No. of training samples per class		
		8	16	32
Bootstrapping I	$r = N - 1$	14.55 (14.02, 15.08)	12.16 (11.92, 12.40)	11.06 (10.96, 11.17)
	$r = N/4$	18.80 (17.99, 19.62)	15.50 (15.01, 15.98)	13.34 (13.15, 13.53)
	$r = N/8$	20.77 (19.73, 21.82)	17.92 (17.24, 18.60)	15.14 (14.85, 15.43)
	$r = N/16$	23.80 (22.41, 25.19)	19.26 (18.43, 20.09)	16.61 (16.19, 17.03)
Bootstrapping II	$r = N - 1$	14.57 (14.02, 15.12)	12.17 (11.93, 12.41)	11.07 (10.96, 11.18)
	$r = N/4$	18.47 (17.71, 19.24)	15.27 (14.79, 15.75)	13.25 (13.04, 13.45)
	$r = N/8$	19.89 (19.03, 20.76)	17.39 (16.81, 17.97)	14.86 (14.59, 15.13)
	$r = N/16$	22.55 (21.32, 23.77)	18.68 (17.97, 19.38)	16.24 (15.89, 16.59)
Bootstrapping III	$r = N - 1$	13.74 (13.37, 14.11)	11.82 (11.62, 12.02)	11.04 (10.92, 11.16)
	$r = N/4$	17.97 (17.12, 18.82)	15.09 (14.60, 15.59)	13.03 (12.83, 13.24)
	$r = N/8$	20.79 (19.72, 21.86)	17.24 (16.63, 17.84)	14.81 (14.53, 15.10)
	$r = N/16$	23.80 (22.45, 25.15)	18.57 (17.83, 19.32)	16.13 (15.77, 16.49)
Bootstrapping IV	$r = N - 1$	13.74 (13.37, 14.11)	11.82 (11.62, 12.02)	11.04 (10.92, 11.16)
	$r = N/4$	17.38 (16.78, 17.98)	14.66 (14.31, 15.00)	12.72 (12.52, 12.91)
	$r = N/8$	19.41 (18.65, 20.16)	16.47 (16.04, 16.89)	14.11 (13.85, 14.37)
	$r = N/16$	22.55 (21.32, 23.77)	18.68 (17.52, 18.86)	16.24 (15.51, 16.15)
Efron's bootstrapping		23.65 (22.33, 24.98)	21.68 (20.78, 22.57)	19.99 (19.44, 20.54)

The first and second lines of the table are the mean and 95% confidence interval from the 100 trials of experiment, respectively

Note that the ratio of the training sample size to the dimensionality is small. In all of the results reported in Tables 4-6, the proposed classifier with a proper  $r$  outperforms  $k$ -NN classifiers. It is important properly to select the value of  $r$ , particularly for nonnormal data, as shown in Table 6. It should be stressed that when  $\Delta = 6$ , the error of the 1-NN classifier with bootstrap samples hardly increases as the dimensionality increases. This contradicts the common belief that the 1-NN classifier needs a very large number of samples for high-dimensional data. Experimental results suggest that the peaking phenomenon of the 1-NN classifier with bootstrap samples will at least be not as severe as that of the conventional  $k$ -NN classifier.

Next, we compared the proposed classifier with  $k$ -NN classifiers on a real data set. In this data set, each class represents one of 10 handwritten numerals. This data set contains 1,400 256-dimensional feature vectors per class. In feature extraction, 256 Gabor filters [14] were applied to a character image. The outputs of Gabor filters produce a 256-dimensional feature vector. Gabor filters tend to detect line and edge segments, which seem to be good discriminating features [15]. We call this the Gabor data set. For additional details refer to [16]. We need to assure the independence between training and test sets. Thus, the Gabor data set was randomly partitioned into two equally sized sets. One set was used to train a classifier and the other set was used to evaluate its performance. This procedure was repeated four times independently and the average of the error rate and its standard deviation were calculated. Results are shown in Table 7. For small values of  $r$ , the proposed classifier worked well, even in high-dimensional spaces. Results show that our technique can be utilized to solve real world problems. The effectiveness of small values of  $r$  suggests that Gabor data may be nonnormal.

TABLE 2  
COMPARISON OF BOOTSTRAP TECHNIQUES  
ON THE NONNORMAL DATA SET IN TERMS OF THE ERROR RATE(%)

		No. of training samples per class		
		8	16	32
Bootstrapping I	$r = N - 1$	43.16 (42.24, 44.08)	44.53 (44.05, 45.01)	46.23 (45.97, 46.48)
	$r = N/4$	24.85 (23.05, 26.64)	15.07 (14.15, 16.00)	11.95 (11.56, 12.34)
	$r = N/8$	23.99 (22.52, 25.46)	16.16 (15.41, 16.91)	12.80 (12.44, 13.15)
	$r = N/16$	25.95 (24.70, 27.20)	18.92 (18.16, 19.68)	14.84 (14.43, 15.25)
Bootstrapping II	$r = N - 1$	43.15 (42.18, 44.13)	44.72 (44.26, 45.18)	46.29 (46.05, 46.53)
	$r = N/4$	23.14 (21.56, 24.73)	14.86 (13.97, 15.75)	11.84 (11.46, 12.21)
	$r = N/8$	21.51 (20.32, 22.71)	15.99 (15.38, 16.59)	12.64 (12.32, 12.96)
	$r = N/16$	22.02 (21.22, 22.83)	17.76 (17.04, 18.49)	14.35 (13.98, 14.72)
Bootstrapping III	$r = N - 1$	44.53 (43.68, 45.39)	45.97 (45.51, 46.44)	46.59 (46.40, 46.77)
	$r = N/4$	24.07 (22.21, 25.94)	14.82 (13.95, 15.69)	11.60 (11.24, 11.95)
	$r = N/8$	22.38 (20.84, 23.91)	15.72 (15.01, 16.44)	12.40 (12.08, 12.72)
	$r = N/16$	27.13 (25.87, 28.39)	18.00 (17.27, 18.73)	14.22 (13.83, 14.61)
Bootstrapping IV	$r = N - 1$	44.53 (43.68, 45.39)	45.97 (45.51, 46.44)	46.59 (46.40, 46.77)
	$r = N/4$	21.53 (20.03, 23.03)	13.99 (13.34, 14.65)	11.39 (11.01, 11.77)
	$r = N/8$	19.96 (18.78, 21.13)	14.70 (14.13, 15.26)	12.10 (11.78, 12.42)
	$r = N/16$	22.02 (21.22, 22.83)	16.94 (16.28, 17.60)	13.82 (13.48, 14.15)
Efron's bootstrapping		26.09 (24.70, 27.49)	22.03 (21.10, 22.95)	18.76 (18.15, 19.37)

### 3.2 Experiment 4

It is believed in the pattern recognition field that classifier performance can be improved by removing outliers, which seem to distort the distribution. As previously mentioned, our bootstrapping method can act as a smoother of the empirical distribution. Thus, the main advantage of our technique seems to derive from the ability to remove outliers. On the other hand, in the NN classifier design, editing algorithms such as CNN [17], RNN [18], and VQ-NN [19] have been proposed for removing outliers from the training set. The purpose of Experiment 4 is to compare the proposed classifier with edited 1-NN classifiers in high-dimensional spaces. So far, we have assumed that the bootstrap sample size (or the number of representatives) and training sample size are the same. On the other hand, the performance of an edited 1-NN classifier must be evaluated in terms of both the error rate and the reduction rate. Hence, one should compare the error of edited 1-NN classifiers when operating at the same range of the reduction rate. In order to reduce the bootstrap sample size, our bootstrapping method is now modified as follows:

- 1) Select a sample  $x_{k_0}^i$  from  $X_{N_i}$  randomly.
- 2) Find the  $r$  nearest neighbor samples  $x_{k_1}^i, x_{k_2}^i, \dots, x_{k_r}^i$  of  $x_{k_0}^i$
- 3) Compute a bootstrap sample  $\tilde{x}_{k_0}^i = \frac{1}{r+1} (x_{k_0}^i + x_{k_1}^i + \dots + x_{k_r}^i)$
- 4) Repeat Steps 1-3  $N_i^*$  ( $N_i^* \leq N_i$ ) times, under a condition that no sample is selected more than once.

Data: Ness ( $\Delta = 2, 6$ ) and nonnormal  
 Dimensionality:  $n=50$   
 No. of training samples:  $N_1 = N_2 = 100$   
 No. of test samples: 5000 per class  
 Classifiers: Proposed ( $r = N/4$ ), CNN, RNN,  
 and VQ-NN classifiers

TABLE 3  
 COMPARISON OF BOOTSTRAP TECHNIQUES ON THE NESS DATA SET  
 IN TERMS OF THE ERROR RATE(%)

		$\Delta$		
		2	4	6
Bootstrapping I	$r = N - 1$	26.05 (25.82, 26.27)	7.54 (7.46, 7.62)	1.35 (1.32, 1.38)
	$r = N/4$	28.16 (27.91, 28.42)	8.67 (8.58, 8.77)	1.63 (1.60, 1.67)
	$r = N/8$	30.07 (29.80, 30.33)	10.17 (10.05, 10.30)	2.01 (1.96, 2.05)
	$r = N/16$	31.69 (31.44, 31.95)	11.62 (11.46, 11.78)	2.49 (2.43, 2.54)
	$r = N - 1$	26.06 (25.83, 26.28)	7.43 (7.37, 7.50)	1.34 (1.31, 1.36)
Bootstrapping II	$r = N/4$	27.92 (27.66, 28.17)	8.53 (8.44, 8.62)	1.58 (1.54, 1.61)
	$r = N/8$	29.87 (29.62, 30.13)	9.84 (9.73, 9.96)	1.92 (1.88, 1.96)
	$r = N/16$	31.98 (31.74, 32.22)	11.32 (11.17, 11.47)	2.64 (2.57, 2.71)
	$r = N - 1$	26.29 (26.10, 26.47)	7.61 (7.53, 7.68)	1.38 (1.35, 1.41)
Bootstrapping III	$r = N/4$	27.88 (27.63, 28.12)	8.44 (8.35, 8.53)	1.56 (1.53, 1.59)
	$r = N/8$	29.65 (29.39, 29.90)	9.65 (9.54, 9.76)	1.87 (1.83, 1.91)
	$r = N/16$	32.43 (32.17, 32.70)	12.03 (11.87, 12.18)	2.73 (2.66, 2.80)
	$r = N - 1$	26.29 (26.10, 26.47)	7.61 (7.53, 7.68)	1.38 (1.35, 1.41)
Bootstrapping IV	$r = N/4$	27.69 (27.45, 27.93)	8.38 (8.29, 8.48)	1.54 (1.51, 1.57)
	$r = N/8$	29.42 (29.14, 29.70)	9.55 (9.43, 9.67)	1.83 (1.79, 1.87)
	$r = N/16$	31.23 (30.99, 31.48)	12.28 (12.11, 12.45)	2.40 (2.34, 2.46)
	$r = N - 1$	26.29 (26.10, 26.47)	7.61 (7.53, 7.68)	1.38 (1.35, 1.41)
Efron's bootstrapping		42.24 (41.98, 42.50)	28.55 (28.02, 29.08)	12.90 (12.46, 13.33)

The first and second lines of the tables are the mean and 95% confidence interval from the 100 trials of experiment, respectively.

TABLE 4  
 COMPARISON OF THE PROPOSED CLASSIFIER AND K-NN CLASSIFIERS  
 ON THE NESS DATA SET ( $\Delta = 2$ ) IN TERMS OF THE ERROR RATE(%).

		Dimensionality $n$		
		10	30	50
The proposed classifier	$r = N - 1$	23.34 (23.22, 23.46)	25.02 (24.86, 25.18)	26.29 (26.10, 26.47)
	$r = N/4$	23.75 (23.59, 23.91)	25.99 (25.78, 26.21)	27.69 (27.45, 27.93)
	$r = N/8$	25.10 (24.91, 25.29)	27.47 (27.23, 27.71)	29.42 (29.14, 29.70)
	$r = N/16$	26.76 (26.50, 27.03)	29.60 (29.37, 29.83)	31.23 (30.99, 31.48)
	1-NN classifier	32.08 (31.77, 32.39)	38.12 (37.84, 38.39)	41.69 (41.46, 41.93)
3-NN classifier	29.00 (28.71, 29.30)	37.63 (37.30, 37.97)	43.12 (42.83, 43.40)	
5-NN classifier	27.76 (27.48, 28.04)	38.05 (37.67, 38.43)	44.38 (44.07, 44.68)	

The first and second lines of the tables are the mean and 95% confidence interval from the 100 trials of experiment, respectively.

TABLE 5  
 COMPARISON OF THE PROPOSED CLASSIFIER AND K-NN CLASSIFIERS  
 ON THE NESS DATA SET ( $\Delta = 6$ ) IN TERMS OF THE ERROR RATE (%)

		Dimensionality $n$		
		10	30	50
The proposed classifier	$r = N - 1$	1.25 (1.23, 1.28)	1.31 (1.28, 1.34)	1.38 (1.35, 1.41)
	$r = N/4$	1.25 (1.22, 1.28)	1.41 (1.37, 1.45)	1.54 (1.51, 1.57)
	$r = N/8$	1.33 (1.31, 1.36)	1.61 (1.56, 1.65)	1.83 (1.79, 1.87)
	$r = N/16$	1.44 (1.41, 1.48)	1.91 (1.87, 1.96)	2.40 (2.34, 2.46)
	1-NN classifier	2.60 (2.52, 2.68)	6.67 (6.43, 6.91)	12.33 (11.97, 12.69)
3-NN classifier	1.89 (1.83, 1.94)	5.10 (4.90, 5.30)	10.83 (10.46, 11.20)	
5-NN classifier	1.73 (1.67, 1.78)	4.75 (4.55, 4.96)	10.45 (10.07, 10.83)	

The first and second lines of Table are the mean and 95% confidence interval from the 100 trials of experiment, respectively.

TABLE 6  
 COMPARISON OF THE PROPOSED CLASSIFIER AND K-NN CLASSIFIERS  
 ON THE NONNORMAL DATA SET IN TERMS OF THE ERROR RATE(%).

		Dimensionality $n$		
		10	30	50
The proposed classifier	$r = N - 1$	47.15 (47.04, 47.26)	47.03 (46.91, 47.15)	46.88 (46.75, 47.00)
	$r = N/4$	10.19 (9.95, 10.42)	14.34 (14.00, 14.68)	17.70 (17.32, 18.07)
	$r = N/8$	10.16 (10.01, 10.31)	13.52 (13.27, 13.77)	16.28 (16.01, 16.55)
	$r = N/16$	10.95 (10.81, 11.09)	14.78 (14.59, 14.98)	17.84 (17.63, 18.04)
	1-NN classifier	16.60 (16.38, 16.81)	24.61 (24.34, 24.88)	29.15 (28.86, 29.43)
3-NN classifier	13.05 (12.89, 13.21)	20.54 (20.26, 20.83)	25.33 (24.86, 25.83)	
5-NN classifier	11.72 (11.58, 11.86)	18.82 (18.52, 19.12)	23.61 (23.35, 23.86)	

The first and second lines of the tables are, respectively, the mean and 95% confidence interval from 100 trials of experiment.

TABLE 7  
 COMPARISON OF THE PROPOSED CLASSIFIER AND K-NN  
 CLASSIFIERS ON THE 256-DIMENSIONAL GABOR DATA SET  
 IN TERMS OF THE ERROR RATE (%).

		Mean	Standard deviation
The proposed classifier	$r = N - 1$	8.47	0.49
	$r = N/2$	4.52	0.27
	$r = N/4$	3.16	0.39
	$r = N/8$	2.49	0.39
	$r = N/16$	2.07	0.39
	$r = N/32$	1.85	0.40
	$r = N/64$	1.76	0.49
	$r = N/128$	1.77	0.54
1-NN classifier		2.89	0.38
3-NN classifier		2.86	0.19
5-NN classifier		2.98	0.19

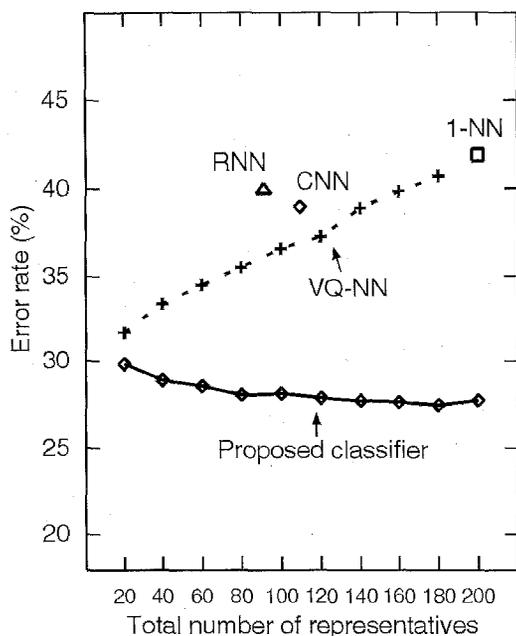


Fig. 1. Comparison of the proposed classifier ( $r = N/4$ ) and edited classifiers on the Ness data set ( $\Delta = 2$ ).

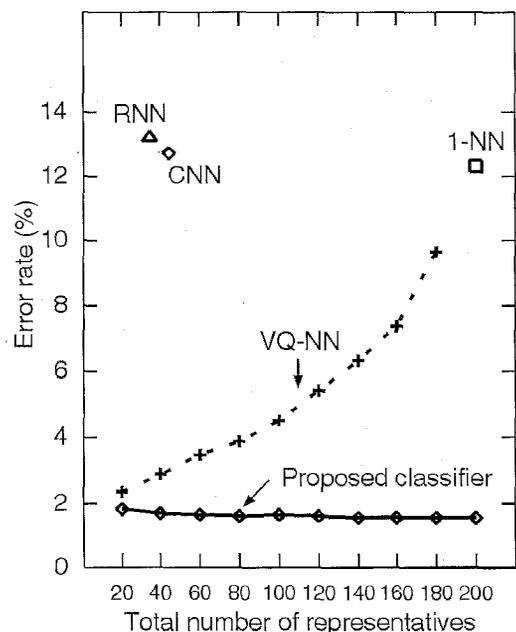


Fig. 2. Comparison of the proposed classifier ( $r = N/4$ ) and edited classifiers on the Ness data set ( $\Delta = 6$ ).

By using the modified bootstrapping method, we can control the bootstrap sample size (i.e., the number of representatives). On the other hand, in CNN and RNN techniques, the number of representatives is itself a random variable and not under the control of the algorithms. Note that when  $N_i^* = N_i$ , the modified bootstrapping method corresponds to Bootstrapping IV. The following experiment was conducted.

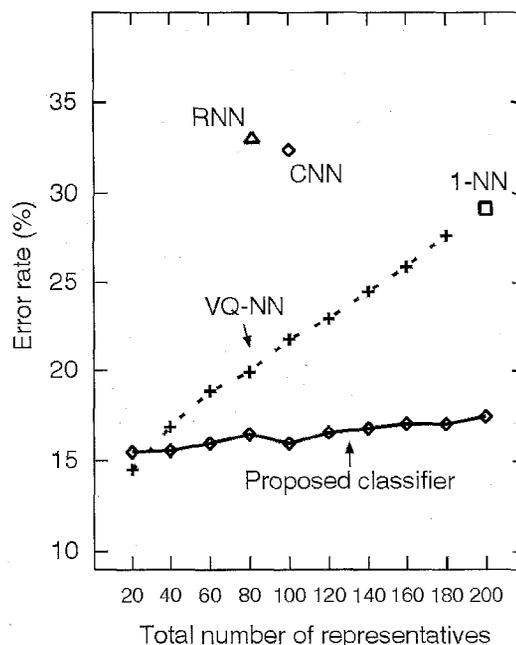


Fig. 3. Comparison of the proposed classifier ( $r = N/4$ ) and edited classifiers on the nonnormal data set.

The results are shown in Figs. 1-3. The error curve is the average of 100 trials. Our main concern is to maintain the error rate while reducing the number of samples. In our bootstrapping method, selection of representatives was done randomly. Nevertheless, Fig. 1 and Fig. 2 show that the best performances were given by the proposed classifier at all the reduction rates. Fig. 3 also shows that at all almost reduction rates, our classifier outperformed other edited classifiers. Thus, this technique can be effectively utilized to perform data reduction in the design of the 1-NN classifier. It is interesting to note that the error curve of the proposed classifier in Fig. 3 tends to decrease rather slowly as the total number of representatives decreases. A similar tendency was observed for the VQ-NN classifier. It has been believed that as the number of representatives decreases, the error increases. However, the results of Experiment 4 contradict the above common belief, and suggest that we may need only a relatively small number of representatives.

Data: Ness ( $\Delta = 2, 6$ ) and nonnormal  
 Dimensionality:  $n = 10, 30, 50$   
 No. of training samples:  $N_1 = N_2 = 100$   
 No. of test samples: 5,000 per class  
 Classifiers: the proposed classifier,  
 1-NN, 3-NN, and 5-NN classifiers

#### 4 OPTIMIZATION OF $r$

Up till now, we have studied the performance of the 1-NN classifier based on our bootstrapping method, assuming that the value of  $r$  is given. From experimental results, the selection of  $r$  is important and strongly affects the classifier performance, especially for nonnormal data. In practice, its proper value must be selected by using only a finite number of training samples. In this section, we study an algorithm for selecting the value of  $r$  which minimizes the error rate estimated by the leave one out method [11].

##### 4.1 Experiment 5

We assume that the bootstrap sample size and training sample size are the same. An algorithm for optimizing the value of  $r$  is as follows:

- 1) Select a candidate  $r$ .
- 2) One sample is excluded.

- 3) By using the  $r$  selected and remaining samples, generate the bootstrap samples.
- 4) Test the excluded sample by the 1-NN classifier with the bootstrap samples.
- 5) Repeat Steps 2-4 until all samples are tested and estimate the error rate.
- 6) If all candidates are evaluated, go to Step 7. Otherwise go to Step 1.
- 7) Find the value of  $r$  which minimizes the estimated error rate.

In order to demonstrate the effectiveness of the above algorithm, the following experiment was performed.

Data: Ness ( $\Delta = 2,6$ ) and nonnormal  
 Dimensionality:  $n = 50$   
 No. of training samples:  $N_1 = N_2 = 100$   
 No. of test samples: 5,000 per class  
 Values of  $r$ :  $N/16, N/8, N/4, N/2, N-1$

For comparison, the performance obtained by the exhaustive method with test samples is presented. In the exhaustive method, we found the optimal value of  $r$  which minimizes the error rate estimated by using test samples. Thus, this performance is viewed as the best one of our classifier. The results are shown in Table 8. The two methods gave similar results. Therefore, the effectiveness of using the leave-one-out method is verified and our technique has practical utility.

TABLE 8  
 COMPARISON OF OPTIMIZATION METHODS  
 IN TERMS OF THE ERROR RATE(%).

	Data set		
	Ness( $\Delta = 2$ )	Ness( $\Delta = 6$ )	non-normal
Leave-one-out method	26.81 (26.51,27.11)	1.49 (1.44,1.54)	16.75 (16.46,17.05)
Exhaustive method with test samples	26.29 (26.10,26.47)	1.38 (1.35,1.41)	16.37 (16.11,16.62)

The first and second lines of Table are the mean and 95% confidence interval from the 100 trials of experiment, respectively.

## 5 CONCLUSIONS

We have proposed a bootstrap technique for the 1-NN classifier design. The performance of the 1-NN classifier based on bootstrap samples was demonstrated on several data sets. It was shown that the proposed classifier outperforms the conventional  $k$ -NN classifiers as well as the edited 1-NN classifiers. Experimental results suggest that the use of our bootstrap samples is an effective means of reducing the bias of the 1-NN error. Moreover, we have discussed the optimization of the parameter  $r$ , which depends strongly on the degree of the nonnormality. We feel that since our bootstrap technique acts as a smoother of the distribution of training samples, this advantage comes from removing the outliers. Therefore, the bootstrap technique should be considered in the design of the 1-NN classifier, particularly in high-dimensional spaces.

## ACKNOWLEDGMENTS

We are very grateful to the anonymous reviewers for their helpful comments. We acknowledge special thanks to Y. Tsuneta who carried out the experiments.

## REFERENCES

- [1] T.M. Cover and P.E. Hart, "Nearest Neighbor Pattern Classification," *IEEE Trans. Information Theory*, vol. 13, no. 1, pp. 21-27, Jan. 1967.
- [2] K. Fukunaga and D.M. Hummels, "Bias of Nearest Neighbor Error Estimates," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, pp. 103-112, Jan. 1987.
- [3] B. Efron, "Bootstrap Methods: Another Look at the Jackknife," *Annual Statistics*, vol. 7, pp. 1-26, 1979.
- [4] A.K. Jain, R.C. Dubes, and C.-C. Chen, "Bootstrap Techniques for Error Estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, pp. 628-633, 1987.
- [5] M.C. Chernick, V.K. Murthy, and C.D. Nealy, "Application of Bootstrap and Other Resampling Techniques: Evaluation of Classifier Performance," *Pattern Recognition Letters*, vol. 3, pp. 167-178, 1985.
- [6] S.M. Weiss, "Small Sample Error Rate Estimation for  $k$ -NN Classifiers," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, pp. 285-289, 1991.
- [7] K. Fukunaga and D.M. Hummels, "Bayes Error Estimation Using Parzen and  $k$ -NN Procedures," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, pp. 634-643, 1987.
- [8] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, second ed. Academic Press, 1990.
- [9] J. Van Ness, "On the Dominance of Non-Parametric Bayes Rule Discriminant Algorithms in High Dimensions," *Pattern Recognition*, vol. 12, pp. 355-368, 1980.
- [10] A.K. Jain and B. Chandrasekaran, "Dimensionality and Sample Size Considerations in Pattern Recognition Practice," *Handbook of Statistics*, P.R. Krishnaiah and L.N. Kanal, eds., vol. 2. North Holland, 1982, pp. 835-855.
- [11] P. Lachenbruch and M. Mickey, "Estimation of Error Rates in Discriminant Analysis," *Technometrics*, vol. 10, pp. 1-11, 1968.
- [12] E. Fix and J.L. Hodges, Jr., "Discriminatory Analysis: Non-parametric Discrimination: Consistency Properties," Report No. 4, USAF School of Aviation Medicine, Randolph Field, Texas, Feb. 1951.
- [13] E. Fix and J.L. Hodges, Jr., "Discriminatory Analysis: Non-parametric Discrimination: Small Sample Performance," Report No. 11, USAF School of Aviation Medicine, Randolph Field, Texas, Aug. 1952.
- [14] D. Gabor, "Theory of Communication," *J. Inst. Elect. Engr.*, vol. 93, pp. 429-459, 1946.
- [15] Y. Hamamoto, S. Uchimura, K. Masamizu, and S. Tomita, "Recognition of Handprinted Chinese Characters Using Gabor Features," *Proc. Third Int'l Conf. Document Analysis and Recognition*, pp. 819-823, Montreal, Aug. 1995.
- [16] Y. Hamamoto, S. Uchimura, M. Watanabe, T. Yasuda, and S. Tomita, "Recognition of Handwritten Numerals Using Gabor Features," *Proc. 13th Int'l Conf. Pattern Recognition*, vol. 3, pp. 250-253, Vienna, Aug. 1996.
- [17] P.E. Hart, "The Condensed Nearest Neighbor Rule," *IEEE Trans. Information Theory*, vol. 14, pp. 515-516, May 1968.
- [18] G.W. Gates, "The Reduced Nearest Neighbor Rule," *IEEE Trans. Information Theory*, vol. 18, pp. 431-433, May 1972.
- [19] Q. Xie, C.A. Laszlo, and R.K. Ward, "Vector Quantization Technique for Nonparametric Classifier Design," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1,326-1,330, 1993.
- [20] G.F. Hughes, "On the Mean Accuracy of Statistical Pattern Recognizers," *IEEE Trans. Information Theory*, vol. 14, no. 1, pp. 55-63, 1968.
- [21] D.J. Hand, "Recent Advances in Error Rate Estimation," *Pattern Recognition Letters*, vol. 4, pp. 335-346, 1986.
- [22] B. Chandrasekaran and A.K. Jain, "On Balancing Decision Functions," *J. Cybernetics and Information Science*, vol. 2, no. 1, pp. 12-15, 1979.