IJDAR (2002) 4: 191–204

International Journal on IJDAR
Document Analysis and Recognition

Performance evaluation of pattern classifiers for handwritten character recognition

Cheng-Lin Liu, Hiroshi Sako, Hiromichi Fujisawa

Central Research Laboratory, Hitachi, 1-280 Higashi-koigakubo, Kokubunji-shi, Tokyo 185-8601, Japan;
e-mail: {liucl, sakou, fujisawa}@crl.hitachi.co.jp

Received: July 18, 2001 / Accepted: September 28, 2001

Abstract. This paper describes a performance evaluation study in which some efficient classifiers are tested in handwritten digit recognition. The evaluated classifiers include a statistical classifier (modified quadratic discriminant function, MQDF), three neural classifiers, and an LVQ (learning vector quantization) classifier. They are efficient in that high accuracies can be achieved at moderate memory space and computation cost. The performance is measured in terms of classification accuracy, sensitivity to training sample size, ambiguity rejection, and outlier resistance. The outlier resistance of neural classifiers is enhanced by training with synthesized outlier data. The classifiers are tested on a large data set extracted from NIST SD19. As results, the test accuracies of the evaluated classifiers are comparable to or higher than those of the nearest neighbor (1-NN) rule and regularized discriminant analysis (RDA). It is shown that neural classifiers are more susceptible to small sample size than MQDF, although they yield higher accuracies on large sample size. As a neural classifier, the polynomial classifier (PC) gives the highest accuracy and performs best in ambiguity rejection. On the other hand, MQDF is superior in outlier rejection even though it is not trained with outlier data. The results indicate that pattern classifiers have complementary advantages and they should be appropriately combined to achieve higher performance.

Keywords: Handwritten character recognition – Pattern classification – Outlier rejection – Statistical classifiers – Neural networks – Discriminative learning – Handwritten digit recognition

1 Introduction

In optical character recognition (OCR), statistical classifiers and neural networks are prevalently used for classification due to their learning flexibility and cheap computation. Statistical classifiers can be divided into parametric classifiers and non-parametric classifiers [1,2]. Parametric classifiers include the linear discriminant function (LDF), the quadratic discriminant function (QDF), the Gaussian mixture classifier, etc. An improvement to QDF, named regularized discriminant analysis (RDA), was shown to be effective to overcome inadequate sample size [3]. The modified quadratic discriminant function (MQDF) proposed by Kimura et al. was shown to improve the accuracy, memory, and computation efficiency of the QDF [4,5]. Non-parametric classifiers include the Parzen window classifier, the nearest neighbor (1-NN) and k-NN rules, the decision-tree, the subspace method, etc. Neural networks for pattern recognition include the multilayer perceptron (MLP) [6], the radial basis function (RBF) network [7], the probabilistic neural network (PNN) [8], the polynomial classifier (PC) [9,10], etc. The LVQ (learning vector quantization) classifier [11,12] can be viewed as a hybrid since it takes the 1-NN rule for classification while the prototypes are designed in discriminative learning as for neural classifiers. The recently emerged classifier, the support vector machine (SVM) [13,14], has many unique properties compared to traditional statistical and neural classifiers.

For character recognition, the pattern classifiers are usually used for classification based on heuristic feature extraction [15] so that a relatively simple classifier can achieve high accuracy. The efficiency of feature extraction and the simplicity of classification algorithms are preferable for real-time recognition on low-cost computers. The features frequently used in character recognition include the chaincode feature (direction code histogram) [4], the K-L expansion (PCA) [16–18], the Gabor transform [19], etc. Some techniques were proposed to extract more discriminative features to achieve high accuracy [20, 21]. Neural networks can also directly work on character bitmaps to perform recognition. This scheme needs a specially designed and rather complicated architecture to achieve high performance, such as the convolutional neural network [22].

For pattern classification, neural classifiers are generally trained in discriminative learning, i.e., the parame-

ters are tuned to separate the examples of different classes as much as possible. Discriminative learning has the potential to yield high classification accuracy, but training is time-consuming and the generalization performance often suffers from over-fitting. In contrast, for statistical classifiers, the training data of each class is used separately to build a density model or discriminant function. Neural networks can also be built in this philosophy, called the relative density approach [23]. This approach is possible (and usually necessary) to fit more parameters without degradation of generalization performance. The instances of this approach are the subspace method [24], the mixture linear model [23], and the auto-associative neural network [25]. We can view the statistical classifiers and the relative density approach as density models or generative models, as opposed to the discriminative models.

In character field recognition, especially integrated segmentation-recognition (ISR) [26–28], we are concerned not only with the classification accuracy of the underlying classifier, but also resistance to outliers. In this paper, we mean by "outliers" the patterns out of the classes that we aim to detect and classify. In ISR, because the characters cannot be segmented reliably prior to classification, the trial segmentation will generate some intermediate non-character patterns. The non-character patterns are outliers and should be assigned low confidence by the underlying classifier so as to be rejected. In this paper, we will evaluate the outlier rejection performance as well as the classification accuracy of some classifiers.

The evaluated classifiers include a statistical classifier (MQDF), three neural classifiers (MLP, RBF classifier, PC), and an LVQ classifier. We selected these classifiers as objects because they are efficient in the sense that high accuracy can be achieved at moderate memory space and computation cost. SVM does not belong to this category because it is very expensive in learning and recognition even though it gives superior accuracy [29]. In the test case of handwritten digit recognition, we will give the results of classification accuracy, ambiguity rejection, and outlier rejection. The performance of outlier rejection is measured in terms of the tradeoff between the false acceptance of outlier patterns and the false rejection of character patterns.

In classification experiments, some additional statistical classifiers are used as benchmarks. The statistical classifiers are "automatic" in the sense that the performance is not influenced by human factors in design [30]. Among the benchmark classifiers, the 1-NN rule is very expensive in recognition. QDF and RDA also have far more parameters than the evaluated classifiers. Whereas LDF has far fewer parameters and the performance is less sensitive to training sample size, its accuracy is insufficient. A single-layer neural network (SLNN) with the same decision rule as LDF but trained by gradient descent to minimize the empirical mean square error (MSE), is tested as well.

To enhance the outlier rejection capability of neural classifiers, some artificial non-character images are generated and used in training together with the character data. The enhanced versions of MLP, RBF classifier, and PC are referred to as EMLP, ERBF, and EPC, respectively. For the LVQ classifier, the deviation of prototypes from the sample distribution is restricted via regularization in training. Experimental results prove that training with outlier data and the regularization of prototype deviation improve the outlier resistance with little loss of classification accuracy.

To our knowledge, this study is the first to systematically evaluate the outlier rejection performance of pattern classifiers. MQDF has produced promising results in handwriting recognition [5,31,32]. In previous works, it was compared with statistical classifiers [33] but was rarely compared with neural classifiers. Besides this, in the implementation of the RBF and LVQ classifiers, we have made efforts to promote their classification accuracy. In training the RBF and ERBF classifiers, the center vectors as well as the connecting weights are updated in discriminative learning [34,35]. The prototypes of the LVQ classifier are learned under the minimum classification error (MCE) criterion [36]. The resulting classifier gives better performance than the traditional LVQ of Kohonen [11].

The rest of this paper is organized as follows. Section 2 reviews related previous works. Section 3 describes the experiment database and the underlying feature extraction method. Section 4 briefly introduces the evaluated classifiers. Section 5 gives the decision rules for classification and rejection. Section 6 presents the experimental results and Section 7 draws concluding remarks.

2 Previous works

Some previous works have contributed to the performance comparison of various classifiers for character recognition and other applications. In the following we will first outline the results of special evaluations in character recognition and then those of other evaluations. Lee and Srihari compared a variety of feature extraction and classification schemes in handwritten digit recognition [37]. Their results showed that the chaincode feature, the gradient feature, and the GSC (gradient, stroke, and concavity) feature [20] yielded high accuracies. As for classification, they showed that the k-NN rule outperformed MLP and a binomial classifier. Kreßel and Schürmann compared some statistical and neural classifiers in digit recognition. Their results favor the performance of PC and MLP [10]. In digit recognition, Jeong et al. compared the performance of some classifiers with variable training sample size [38]. They showed that the 1-NN rule, MLP, and RDA give high accuracy and their performance is less sensitive to sample size.

Blue et al. compared some classifiers in fingerprint classification and digit recognition and reported that the PNN and the k-NN rule performed well for either problem, whereas MLP and the RBF classifier performed well only in fingerprint classification [16]. Holmström et al. compared a large collection of statistical and neural

classifiers in two application problems: handwritten digit recognition and phoneme classification [17]. Their results showed that the Parzen classifier, the 1-NN and k-NN rules, and some learning classifiers yield good performance, while MLP, RDA, and QDF perform well in digit recognition only. Jain et al. provided comparative results of some classifiers and classifier combination schemes on several small datasets including a digit dataset [39]. Their results favor the performance of non-parametric statistical classifiers (1-NN, k-NN, Parzen). Some works have focused on the performance of LVQ. In speech recognition experiments, Kohonen high lighted the performance of LVQ over QDF and the k-NN rule [11,40]. Liu and Nakagawa compared some prototype learning algorithms (variations of LVQ) in handwritten digit recognition and Chinese character recognition. They showed that LVQ classifiers trained by discriminative learning outperform the 1-NN and k-NN rules [12].

The union of the classifiers evaluated in the above works is obviously very large, yet the intersection contains few classifiers. MLP and the 1-NN and k-NN rules were tested in most evaluation studies and they mostly show good performance. Other classifiers of good performance include the Parzen window classifier and RDA. The RBF classifier, PC, and the LVQ classifier were not widely tested. PC has shown superior performance in [10] while the performance of the RBF and LVQ classifiers depends on the implementation of learning algorithm. Regarding the comparison of different classifiers, the order of performance depends on the underlying application. In addition, for the same application, the order of performance depends on the dataset, pattern representation, and the training sample size.

The outlier resistance of pattern classifiers has been addressed in handwriting recognition. The experiments of [20,41,42] showed that neural networks are inefficient regarding rejecting outlier patterns. Gori and Scarselli showed theoretically that MLP is inadequate for rejection and verification tasks [43]. Some works have sought to improve the performance of outlier rejection. Bromley and Denker showed that the outlier rejection capability could be improved if the neural network was trained with outlier data [44]. Gader et al. assigned fuzzy membership values as targets in neural network training and showed that the fuzzy neural network produced higher accuracy in word recognition although the accuracy of isolated character recognition was sacrificed [41,42]. Chiang proposed a hybrid neural network for character confidence assignment, which is effective to resist outliers in word recognition [45]. Tax and Duin exploited the instability of classifier outputs to detect outliers [46], while Suen et al. achieved outlier resistance by combining multiple neural networks [47].

Besides the efforts of giving in-built outlier resistance to classifiers, some measures have been taken to improve the overall performance of handwriting recognition. Martin and Rashid trained a neural network to signify whether an input window is a centered character or not [48]. Gader et al. have also trained inter-character networks to measure the spatial compatibility of adjacent patterns so as to reduce segmentation error [41,42]. Ha designed a character detector by combining the outputs of neural classifiers and the transition features of the input image [49]. Zhu et al. discriminated between connected character images and normal character images using the Fourier transform [50]. LeCun et al. proposed a global training scheme for segmentation-recognition systems, wherein the segmentation errors are under-weighted [22].

3 Database and feature extraction

For evaluation experiments, we extracted some digit data from the CD of NIST Special Database 19 (SD19). NIST SD19 contains the data of SD3 and SD7, which consist of the character images of 2,100 writers and 500 writers, respectively. SD3 was released as the training data of the First Census OCR Systems Conference and SD7 was used as the test data. It was revealed that the character images of SD3 are cleaner than those of SD7, so the classifiers trained with SD3 failed to give high accuracy compared to SD7 [51]. Therefore, some researchers mixed the data of SD3 and SD7 to make new training and test datasets, such as the MNIST database [29]. The MNIST database has been widely used in the benchmarking of machine learning and classification algorithms. Unfortunately, it provides only the normalized and gray-scaled data and we could not find the original images. We hence compiled a new experiment database. Specifically, we use the digit patterns of writers 0–399 (SD3) and writers 2100–2299 (SD7) for training, the digit patterns of writers 400–499 (SD3) and writers 2,300–2,399 (SD7) for cross validation, and the digit patterns of writers 500–699 (SD3) and writers 2,400-2,599 (SD7) for testing. The validation dataset is used in discriminative learning initialized with multiple seeds to select a parameter set.

In total, the training, validation, and test datasets contain the digit patterns of 600 writers, 200 writers, and 400 writers, respectively. The numbers of patterns of each class are listed in Table 1. Some images of test data are shown in Fig. 1.

To test the outlier rejection performance, we generated two types of outlier data. The type 1 outlier patterns were generated via merging and splitting digit images. A pair of digit images generate four outlier patterns: full-full combination, full-half combination, half-full combination, and half-half combination. Two groups of ten digits are combined into 100 pairs. In a pair of digits, the right one is scaled with the aspect ratio preserved such that the two images have the same diameter of the minimum bounding box. The vertical centers of two images are aligned in the merged image, and for merging a half image, the digit image is split at the horizontal center. We generated 16,000 training patterns of type 1 outlier from the training digit data and 10,000 test patterns from the test digit data. Some examples of type 1 outlier data are shown in Fig. 2.

Table 1. Numbers of patterns of each databset and each class

Dataset	Total	0	1	2	3	4	5	6	7	8	9
Training	66,214	6,639	7,440	6,593	6,795	6,395	5,915	6,536	6,901	6,487	6,513
Validation	22,271	2,198	2,509	2,245	2,275	2,158	1,996	2,201	2,353	2,161	2,175
Test	45,398	4,469	5,145	4,524	4,603	4,387	4,166	4,515	4,684	4,479	4,426

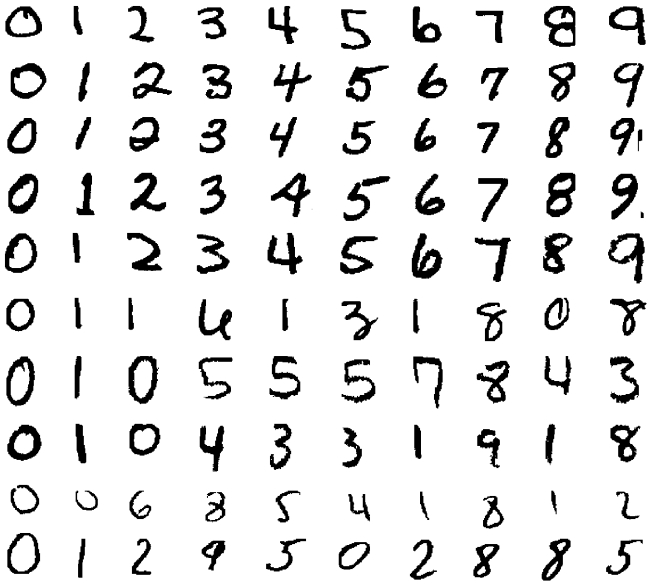

Fig. 1. Examples of test digit data

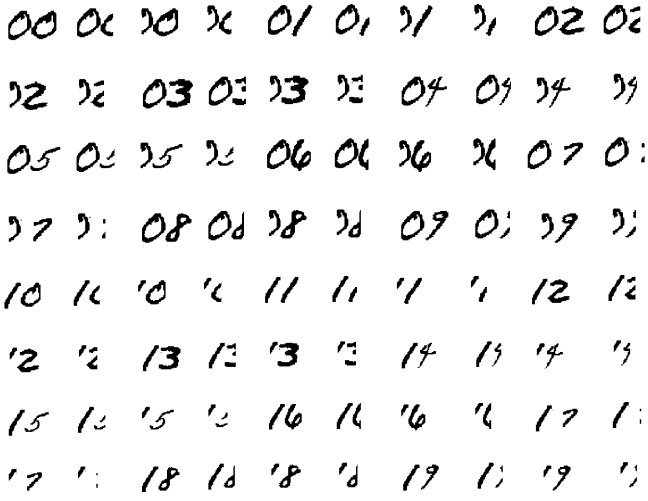

Fig. 2. Examples of type 1 outlier data

To test the resistance to confusing outlier patterns, we collected some handwritten English letter images of NIST SD3 as type 2 outlier data. The type 2 outlier dataset has 8,800 patterns, 200 from each of 44 classes (all letters except "IOZiloqz"). That we use the English letter images as outliers to reject does not imply that we aim to separate letters from digits. Instead, we use the letter images only to test the outlier rejection capability of the pattern classifiers.

Each pattern (either digit or outlier) is represented as a feature vector of 100 direction measurements (chain-code feature). First, the pattern image is scaled into a standard size of a 35×35 grid. To alleviate the distortion caused by size normalization, we render the aspect ratio of the normalized image adaptable to the original aspect ratio [52]. The normalized image is centered in the standard plane if the boundary is not filled. The contour pixels of the normalized image are assigned to four direction planes corresponding to the orientations of chain-codes in a raster scan procedure [21]. On the 35×35 grid of a direction plane, 5×5 blurring masks are uniformly placed to extract 25 measurements. The blurring mask is a Gaussian filter with the variance parameter determined by the sampling theorem. In total, 100 measurements are obtained from four direction planes. We did not try more discriminative features since our primary intention was to evaluate the performance of pattern classifiers.

The feature measurements obtained as such are causal variables, i.e., the value is always positive. It was shown that by power transformation of variables, the density function of causal variables becomes closer to a Gaussian distribution [1]. This is helpful to improve the classification performance of statistical classifiers as well as neural classifiers. Experimental results have demonstrated the effectiveness of power transformation [53]. We transform the measurements by $y = x^u$ with $u = 0.5$. The transformed measurements compose a new feature vector for pattern classification.

4 Evaluated classifiers

4.1 Statistical classifiers

Statistical classifiers are generally based on the Bayesian decision rule, which classifies the input pattern to the class of maximum a posteriori probability. The QDF is obtained under the assumption of equal a priori probabilities and multivariate Gaussian density for each class. The LDF is obtained by further assuming that all the classes share a common covariance matrix.

Derived from negative log-likelihood, the QDF is actually a distance metric, i.e., the class of minimum distance is assigned to the test pattern. On an input pattern $\mathbf{x}$, the QDF of a class has the form:

$$g_0(\mathbf{x}, \omega_i) = (\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i) + \log |\Sigma_i| = \sum_{j=1}^{d} \frac{1}{\lambda_{ij}} [(\mathbf{x} - \mu_i)^T \phi_{ij}]^2 + \sum_{j=1}^{d} \log \lambda_{ij}, \quad (1)$$

where μ_i and Σ_i denote the mean vector and the covariance matrix of class ω_i, respectively. λ_{ij}, $j = 1, 2, \ldots, d$, denote the eigenvalues of class ω_i sorted in decreasing order, and ϕ_{ij}, $j = 1, 2, \ldots, d$, denote the corresponding eigenvectors.

In the QDF, replacing the minor eigenvalues λ_{ij} ($j > k$) with a larger constant δ_i, the modified quadratic discriminant function (MQDF2) [1] is obtained:

$$\begin{aligned} & g_2(\mathbf{x}, \omega_i) \\ &= \sum_{j=1}^{k} \frac{1}{\lambda_{ij}} [(\mathbf{x} - \mu_i)^T \phi_{ij}]^2 + \sum_{j=k+1}^{d} \frac{1}{\delta_i} [(\mathbf{x} - \mu_i)^T \phi_{ij}]^2 \\ &\quad + \sum_{j=1}^{k} \log \lambda_{ij} + (d-k) \log \delta_i \\ &= \sum_{j=1}^{k} \frac{1}{\lambda_{ij}} [(\mathbf{x} - \mu_i)^T \phi_{ij}]^2 + \frac{1}{\delta_i} D_c(\mathbf{x}) \\ &\quad + \sum_{j=1}^{k} \log \lambda_{ij} + (d-k) \log \delta_i, \end{aligned} \tag{2}$$

where k denotes the number of principal components and $D_c(\mathbf{x})$ is the square Euclidean distance in the complement subspace:

$$D_c(\mathbf{x}) = \|\mathbf{x} - \mu_i\|^2 - \sum_{j=1}^{k} [(\mathbf{x} - \mu_i)^T \phi_{ij}]^2$$

Compared to the QDF, the MQDF2 saves much memory space and computation cost because only the principal components are used. The parameter δ_i of MQDF2 can be set class-independent as proposed by Kimura et al. [4], which performs fairly well in practice. It can also be class-dependent as the average variance in the complement subspace spanned by the minor eigenvectors [54, 55].

The RDA improves the performance of the QDF in another way. It smoothes the covariance matrix of each class with the pooled covariance matrix and the identity matrix. We simply combine the sample estimate covariance matrix with the identity matrix:

$$\hat{\Sigma}_i = (1 - \gamma)\Sigma_i + \gamma \sigma_i^2 I, \tag{3}$$

where $\sigma_i^2 = \frac{1}{d}\mathrm{tr}(\Sigma_i)$, and $0 < \gamma < 1$. We also combine the principle of RDA into MQDF. After smoothing the covariance matrix as in (3), the eigenvectors and eigenvalues are computed and the class-dependent constant δ_i is set as the average variance in the complement subspace. To differentiate from the MQDF2 without regularization, we refer to the discriminant function with regularization as MQDF3.

[1] In [4], MQDF1 referred to another modification of the QDF without truncation of eigenvalues.

4.2 Neural classifiers

We use an MLP with one hidden layer to perform the classification task. Each unit in the hidden layer and the output layer has sigmoid nonlinearity. The error back-propagation (BP) algorithm [6] is used to learn the connecting weights by minimizing the mean square error (MSE) over a set of N_x examples:

$$E = \frac{1}{N_x} \Big\{ \sum_{n=1}^{N_x} \sum_{k=1}^{M} \big[y_k(\mathbf{x}^n, \mathbf{w}) - t_k^n \big]^2 + \lambda \sum_{w \in W} w^2 \Big\}, \tag{4}$$

where λ is a coefficient to control the decay of the connecting weights (excluding the biases). $y_k(\mathbf{x}^n, \mathbf{w})$ is the output for class k on an input pattern $\mathbf{x}^n$; t_k^n is the desired value of class k, with value 1 for the genuine class and 0 otherwise. If $\mathbf{x}^n$ is an outlier pattern, however, all the target values are set to 0. The criterion function of (4) is used for the training of MLP, RBF classifier, and PC.

The RBF classifier has one hidden layer with each hidden unit being a Gaussian kernel, and each output is a linear combination of the Gaussian kernels with sigmoid non-linearity to approximate the class membership. For training the RBF classifier, the BP algorithm (stochastic gradient descent to minimize the empirical MSE) can be used to update all the parameters [7]. It was reported in [34, 35] that the updating of the kernel widths does not benefit the performance. Hence we compute the kernel widths from the sample partition by k-means clustering and fix them during BP training. The center vectors are initialized from clustering and are updated by gradient descent along with the connecting weights.

The polynomial classifier (PC) is a single layer network with the polynomials of the input measurements as inputs. We use a PC with binomial inputs on the feature subspace learned by PCA (principal component analysis) on the pooled sample data. Denoting the feature vector in the m-dimensional ($m < d$) subspace as $\mathbf{z}$, the output corresponding to a class is computed by:

$$y_k(\mathbf{x}) = s\Big[\sum_{i=1}^{m} \sum_{j=i}^{m} w_{kij}^{(2)} z_i(\mathbf{x}) z_j(\mathbf{x}) + \sum_{i=1}^{m} w_{ki}^{(1)} z_i(\mathbf{x}) + w_{k0} \Big] \tag{5}$$

where $z_j(\mathbf{x})$ is the projection of $\mathbf{x}$ onto the jth principal axis of the feature subspace.

4.3 LVQ Classifier

For the LVQ classifier, we adopt the prototype learning algorithm with the minimum classification error (MCE) criterion [36]. The prototypes are updated by stochastic gradient descent on a training sample set with aim of minimizing a loss function relevant to the classification error. We give in the following the learning rules, while the details can be found in [12].

On an input pattern $\mathbf{x}$, the closest prototype $\mathbf{m}_{ki}$ in the genuine class k and the closest rival prototype $\mathbf{m}_{rj}$ from class r are searched for. The misclassification loss is computed by

$$l_k(\mathbf{x}^n) = l_k(\mu_k) = \frac{1}{1+e^{-\xi\mu_k}}, \tag{6}$$

with

$$\mu_k(\mathbf{x}) = d(\mathbf{x}, \mathbf{m}_{ki}) - d(\mathbf{x}, \mathbf{m}_{rj}).$$

The distance metric is the square Euclidean distance.

On a training pattern, the prototypes are updated by

$$\begin{cases} \mathbf{m}_{ki} = \mathbf{m}_{ki} + 2\alpha(t)\xi l_k(1-l_k)(\mathbf{x}-\mathbf{m}_{ki}) \\ \mathbf{m}_{rj} = \mathbf{m}_{rj} - 2\alpha(t)\xi l_k(1-l_k)(\mathbf{x}-\mathbf{m}_{rj}) \end{cases}, \tag{7}$$

where $\alpha(t)$ is a learning rate, which is sufficiently small and decreases with time. It was suggested that the parameter ξ increases progressively in learning [12]. Nevertheless, our recent experiments showed that a constant ξ leads to convergence as well.

To restrict the deviation of prototypes from the sample distribution, we incorporate the WTA (winner-take-all) competitive learning rule to give the new learning rule:

$$\begin{cases} \mathbf{m}_{ki} = \mathbf{m}_{ki} + 2\alpha(t)[\xi l_k(1-l_k)(\mathbf{x}-\mathbf{m}_{ki}) + \lambda(\mathbf{x}-\mathbf{m}_{ki})] \\ \mathbf{m}_{rj} = \mathbf{m}_{rj} - 2\alpha(t)\xi l_k(1-l_k)(\mathbf{x}-\mathbf{m}_{rj}) \end{cases}, \tag{8}$$

where λ is the regularization coefficient. The regularization trades off the classification accuracy but is effective to improve the outlier resistance.

5 Decision rules

In classification, the neural classifiers take the class corresponding to the maximum output while MQDF and the LVQ classifier take the class of minimum distance. For the neural classifiers, whether the output is linear or nonlinear does not affect the classification result. For ambiguity rejection and outlier rejection, this also makes little difference. We take linear outputs in subsequent experiments. For the LVQ classifier, the distance of a class is defined as the distance between the input pattern and the closest prototype from this class.

In classification, a pattern is considered ambiguous if it cannot be reliably assigned to a class, whereas a pattern assigned low confidence for all hypothesized classes is considered as an outlier. Chow gave a rejection rule whereby the pattern is rejected if the maximum a posteriori probability is below a threshold [56]. Dubuison and Masson gave a distance reject rule based on the mixture probability density of hypothesized classes [57], which can be used for outlier rejection. Due to the difficulty of probability density and a posteriori probability estimation, pattern recognition engineers are prone to use empirical rules based on the classifier outputs (class scores or distances) [58]. Usually, two thresholds are set for the measure of the top rank class and the difference of scores between two top rank classes, respectively. Even though the two thresholds can be used jointly, we assume they play different roles and will test their effects separately. We call the rejection rule based on the top rank class and the one based on two top rank classes rejection rule 1 (RR1) and rejection rule 2 (RR2), respectively.

For the neural classifiers, we denote the maximum output and the second maximum output as y_i and y_j, corresponding to class i and class j, respectively. By RR1, if $y_i < T_1$, the input pattern is rejected; while by RR2, if $y_i - y_j < T_2$, the input is rejected. RR1 rejects the input pattern because all classes have low confidence so the pattern is considered as an outlier, whereas RR2 rejects because the two top rank classes cannot be discriminated reliably, so the pattern is ambiguous. For distance-based classifiers, including MQDF and the LVQ classifier, we denote the distances of two top rank classes as d_i and d_j, respectively. The relation $d_i \leq d_j$ holds. By RR1, if $d_i > D_1$, the input pattern is rejected; while by RR2, if $d_j - d_i < D_2$, the input pattern is rejected.

Even though in the experiments we make hard decisions regarding classification, ambiguity rejection, and outlier rejection with variable thresholds, they are not necessarily made for intermediate patterns in practical handwriting recognition integrating segmentation and classification. Instead, the classifier gives membership scores to the hypothesized classes for each pattern, and the class identities of patterns are determined in the global scoring of the character field. However, the performances of classification accuracy, ambiguity discrimination, and outlier resistance are important for the overall recognition system all the time. Our experimental results with hard decisions hence provide an indication of the performance of classifiers.

6 Experimental results

6.1 Accuracy on variable parameter complexity

We first trained the five evaluated classifiers with a variable number of parameters so as to choose an appropriate parameter complexity for each classifier. The parameter δ_i of the MQDFs was set in three ways. The class-independent δ_i (this classifier is referred to as *Const*) was set to be proportional to the average variance of ten classes. The class-dependent δ_i was set to the average of minor eigenvalues for both the MQDF2 (this classifier is referred to as *Aver*) and the MQDF3. The regularization coefficient of the MQDF3 was set to $\gamma = 0.2$. The number of principal eigenvectors was set to $k = 10i$, $i = 1, 2, \ldots, 8$.

The number of hidden units of MLP and RBF classifier was set to $N_h = 50i$, $i = 1, 2, \ldots, 6$. For the LVQ classifier, each class has the same number of prototypes $n_p = 5i$, $i = 1, 2, \ldots, 6$. For the PC, the dimension

of feature subspace was set so that the total number of parameters (including the eigenvectors and connecting weights) approximately equals the number of parameters of MLP. Consequently, the dimensions of the subspace corresponding to the hidden unit numbers are $m = 24, 37, 47, 56, 64, 71$. The weight decay coefficient was set to 0.05 for MLP, 0.02 for the RBF classifiers, and 0.1 for the PC. The regularization coefficient of the LVQ classifier was set to $\lambda = 0.05/var$, where var is the average within-cluster variance after k-means clustering of the sample data. Each configuration of neural classifiers and LVQ was trained with three sets of initialized parameters from different random seeds. After training, the parameter set that gives the highest accuracy in the validation was retained.

The accuracies of the MQDFs on the test data are plotted in Fig. 3. We can see that the accuracy of the MQDF3 increases with the number of eigenvectors but saturates at a certain point, while the accuracy of the MQDF2 (Aver) and the MQDF2 (Const) decreases with the number of eigenvectors from a certain point. The performance of the MQDF2 (Const) and the MQDF3 are superior to that of the MQDF2 (Aver), while the MQDF3 performs better than the MQDF2 (Const) on a large number of eigenvectors. We choose the number $k = 40$ for the MQDF3 for subsequent tests.

The accuracies of the neural and LVQ classifiers (which are typical of discriminative models) on the test data are plotted in Fig. 4 (the hidden unit number of the LVQ classifier is the total number of prototypes and the PC has a corresponding dimension of subspace). We can see that the discriminative models yield fairly high accuracy even when the parameter complexity is low, and the accuracy increases slowly with parameter complexity until it saturates at a certain point. The four classifiers achieve the highest accuracy at 250 hidden units or 300 hidden units while these two points make little difference. To make the four classifiers have approximately the same complexity, we selected the classifier configurations corresponding to 250 hidden units. In comparison of the four classifiers, PC gives the highest accuracy, globally. It is also evident that the RBF classifier outperforms MLP, and the LVQ classifier is inferior to MLP. An additional advantage of PC is that the performance is not influenced by the random initialization of connecting weights, so it is not necessary to train with multiple trials.

Comparing the results of Fig. 3 and Fig. 4, we can see that the accuracy of the discriminative models is prominently higher than that of the MQDFs. It will be shown later that the discriminative models also have fewer parameters than the MQDF.

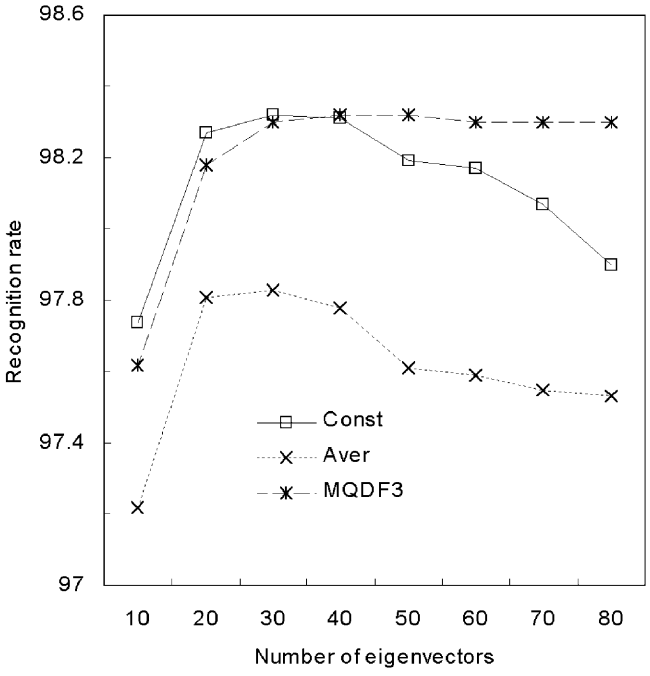

Fig. 3. Recognition accuracies of the MQDFs

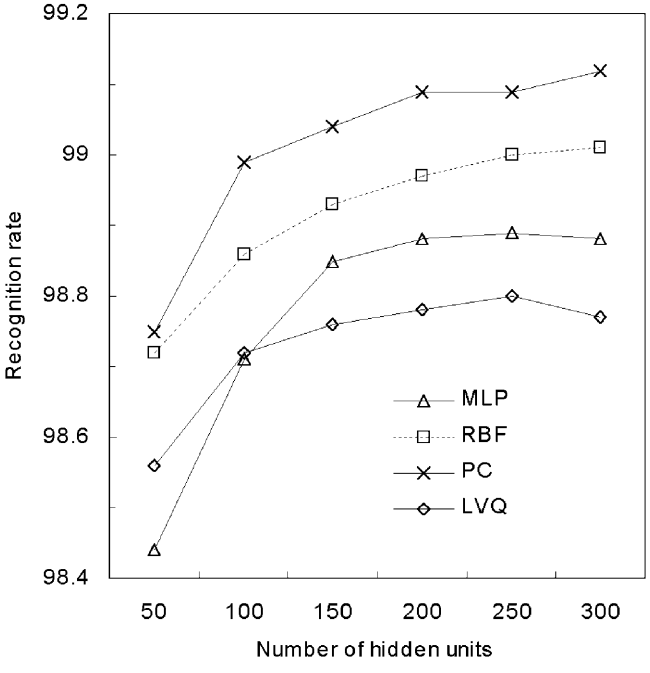

Fig. 4. Recognition accuracies of the discriminative models

6.2 Accuracy on variable sample size

The evaluated classifiers as well as some benchmark classifiers were trained with a variable number of patterns. In addition to the whole set of 66,214 training patterns (s66k), the classifiers were trained with variable sizes of subsets of the whole training data. To generate training subsets, a fraction of patterns are uniformly extracted from the samples of each class. At the fractions 1/2, 1/4, 1/8, 1/16, and 1/32, we obtained subsets of 33,103 patterns (s33), 16,549 patterns (s16k), 8,273 patterns (s8k), 4,134 patterns (s4k), and 2,064 patterns (s2k). The parameter complexity of MLP, the RBF classifier, and the LVQ classifier is variable on the training sample size, while the parameter complexity of other classifiers is fixed. On the sample sizes s33, s16k, s8k, s4k, and s2k, the number of hidden units is 200, 160, 120, 80, and 40,

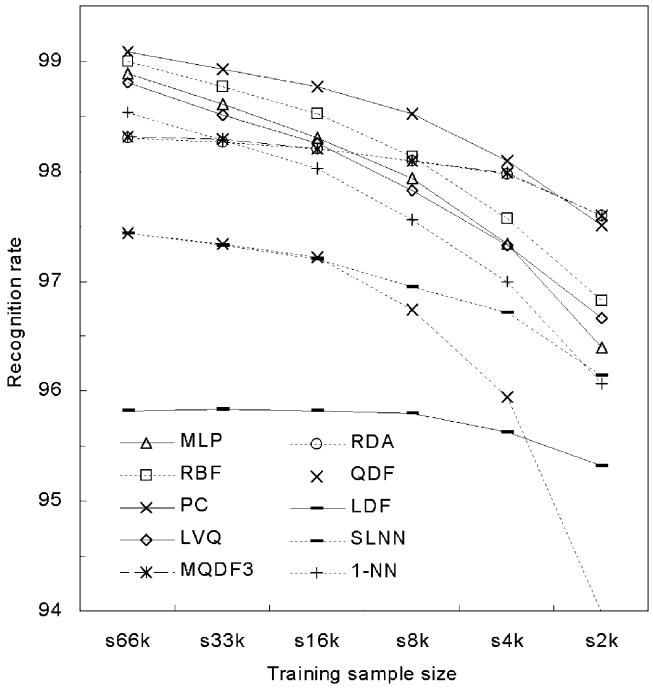

Fig. 5. Recognition accuracy versus training sample size

respectively, for the MLP, RBF, and LVQ classifiers. It was observed that their performance saturates at fewer hidden units on a smaller sample size. However, even on a small sample size, the PC yields high accuracy on high dimensions of feature subspace. Hence, we fixed the dimension of subspace as 56 (corresponding to 200 hidden units for MLP).

Trained with variable sample sizes, the accuracies on 45,398 test patterns are plotted in Fig. 5. We can see that the sensitivity of classification accuracy to the training sample size differs drastically from classifier to classifier. The accuracy of the MQDF3 is almost identical to that of RDA, and their performance is highly stable against the training sample size. The discriminative models give higher accuracies than the 1-NN rule in all sample sizes, and the accuracy of PC is the highest except on the smallest sample size. It is evident that the accuracies of the discriminative models are rather sensitive to the training sample size. As for the benchmark classifiers, the accuracy of the 1-NN rule is fairly high, but is as sensitive to sample size as the discriminative models. QDF is very sensitive to the training sample size. While the performance of LDF is insensitive to sample size, its accuracy is insufficient. The SLNN trained by gradient descent gives much higher accuracy than LDF. Comparing the MQDF3 to the discriminative models, we can see that the discriminative models yield higher accuracy on large sample sizes while the MQDF3 yields higher accuracy on small sample sizes.

The parameter and computation complexities of the classifiers are listed in Table 2. The number of parameters and the number of multiplications in classifying an input pattern are given, and the ratio of parameters/computations to LDF/SLNN is given to indicate the relative complexity. The number d (= 100) denotes the dimensionality of the input pattern. MLP and the RBF classifier have 250 hidden units each, the subspace dimension of the PC is $m = 64$, and the LVQ classifier has 25 prototypes for each class. The given computation number of the LVQ classifier and the 1-NN rule is the upper bound because the search of the nearest neighbor can be accelerated using partial distances. The computation of some classifiers (such as the RBF classifier and the MQDF) is a little more complicated than the given index because they involve division, exponential or logarithm calculation in addition to the multiplications. We can see that for each classifier, the computation complexity is approximately proportional to the parameter complexity. The four discriminative models have approximately the same complexity. The MQDF has more parameters and costs more in computation than the discriminative models, but when compared to the QDF and the RDA, it saves more than half of the memory space and computation.

6.3 Ambiguity rejection

We tested the performance of eight classifiers, namely, MLP, RBF, PC, LVQ, MQDF3, EMLP, ERBF, and EPC, in terms of ambiguity rejection (reject-error tradeoff) and outlier rejection. The enhanced versions of the neural classifiers have the same parameter complexity as their original counterparts. They were trained with 66,214 digit patterns and 16,000 outlier patterns of type 1. The cumulative accuracies of the classifiers on 45,398 test patterns are shown in Table 3. It is shown that on training with outlier data, the enhanced neural classifiers do not lose accuracy compared to their original versions.

Two rejection rules (RR1 and RR2) were used to test the reject-error tradeoff of eight classifiers. Comparing the results of two rules as in Fig. 6, we can see that the rejection rule RR2 is more appropriate for ambiguity rejection than RR1. The reject-error tradeoff of RR2 is better than that of RR1 for all the classifiers. The reject-error plots of the discriminative models are evidently better than that of the MQDF3, and among them, the plot of PC is the best. The reject-error tradeoff of the enhanced neural classifiers is close to that of their original versions. From the results, we can say that the contrast of reject-error tradeoff is approximately consistent with the contrast of classification accuracy. This implies that improving the classification accuracy generally benefits the rejection of ambiguous patterns.

6.4 Outlier rejection

Two rejection rules RR1 and RR2 were used to test the rejection of 10,000 type 1 outlier patterns with variable thresholds. The thresholds of RR1 and RR2 for outlier rejection were also used to test the digit patterns (45,398

Table 2. Parameter and computation complexities of classifiers

Classifier	Parameters	Para. ratio	Multiplications	Comp. ratio
MLP/RBF	$10(250+1)+250(d+1)$	27.49	$250 \cdot d + 10 \cdot 250$	27.5
PC	$m \cdot d + 10[\binom{m}{2} + 2m + 1]$	27.57	$m \cdot d + \binom{m}{2} + 10[\binom{m}{2} + 2m]$	29.86
LVQ	$10 \cdot 25 \cdot d$	24.75	$\leq 10 \cdot 25 \cdot d$	25
MQDF	$10[(40+1)d+41]$	41	$10[40(d+1)+40+1]$	41.41
QDF/RDA	$10(d+d^2)$	100	$10 \cdot d(d+2)$	102
LDF/SLNN	$10(d+1)$	1	$10 \cdot d$	1
1-NN	$66,214 \cdot d$	6,555.8	$\leq 66,214 \cdot d$	6,621.4

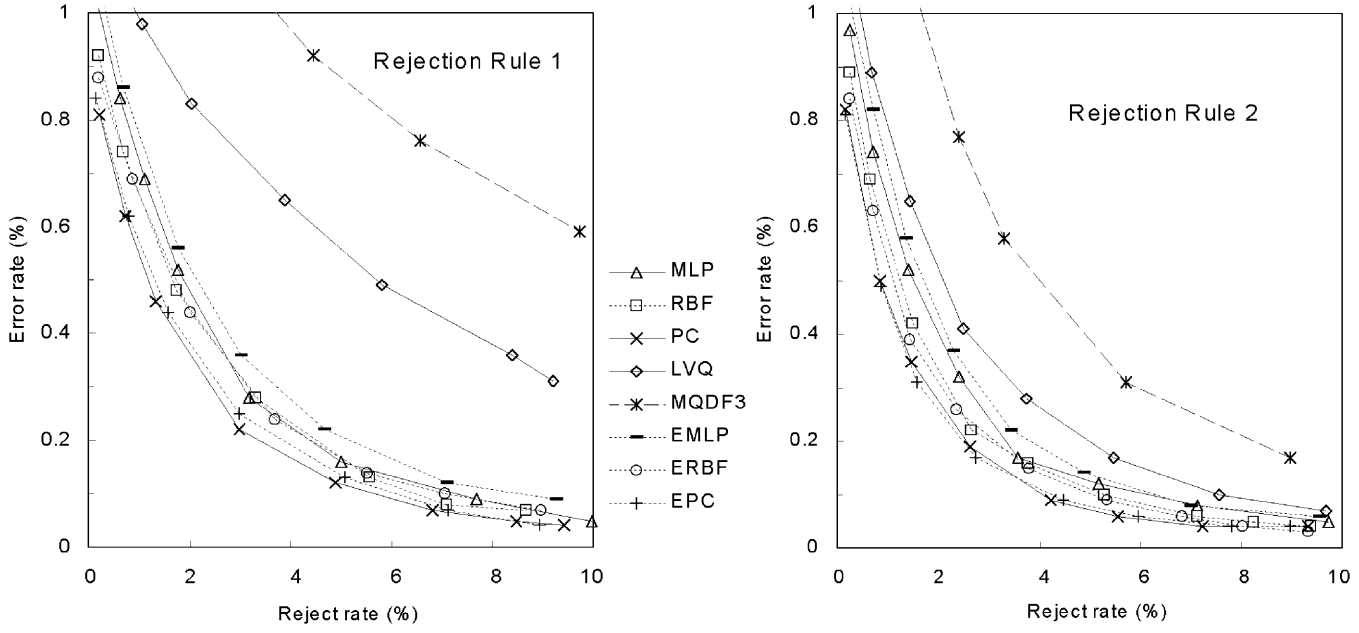

Fig. 6. Reject-error tradeoff of character classification

images) so as to give the tradeoff between the false rejection of character patterns and the false acceptance (false alarm) of outlier patterns. The plots of the false reject-alarm tradeoff are shown in Fig. 7. We can see that for the majority of classifiers, the outlier rejection performance of RR1 is better than that of RR2. As exceptions, for the neural classifiers trained without outlier data, the performance of RR2 is better than that of RR1. While for the enhanced neural classifiers, RR1 performs much better than RR2. For the MQDFs and the LVQ classifier, the performance of RR2 is very poor. Based on these results, we conclude that the rejection rule RR1 is more appropriate for outlier rejection.

Comparing the false reject-alarm plots of eight classifiers on type 1 outlier data, we can see that the outlier rejection performance of the MQDF3 is by far the best. The performance of the enhanced neural classifiers has been largely improved compared to their original versions and the reject-alarm tradeoff approaches that of the MQDF3. By restricting the prototype deviation from the sample distribution, the LVQ classifier also performs fairly well in outlier rejection. The performance of the MQDF3 is very promising in that its parameters were trained without outlier data.

On the type 2 outlier data, the rejection rule RR1 performs better than RR2 for all the classifiers. The reject-alarm plots on type 2 outlier data are shown in Fig. 8. Since the type 2 outlier data was not used in training and the pattern shapes of English letters inherently resemble the digit patterns, the false acceptance rate is relatively high. Additionally, in this case, the difference of performance between different classifiers is not so remarkable as for type 1 outlier data. However, as shown in Fig. 8, some classifiers can still reject more than 50% of English letter images at a low rejection rate of digit patterns. We can see that the MQDF3 and the ERBF classifier are the top performers, whereas the EMLP and the EPC are inferior despite performing very well on type 1 outlier data. The LVQ classifier also performs fairly well on type 2 outlier data, even though it was not trained with outlier data as for the MQDF3.

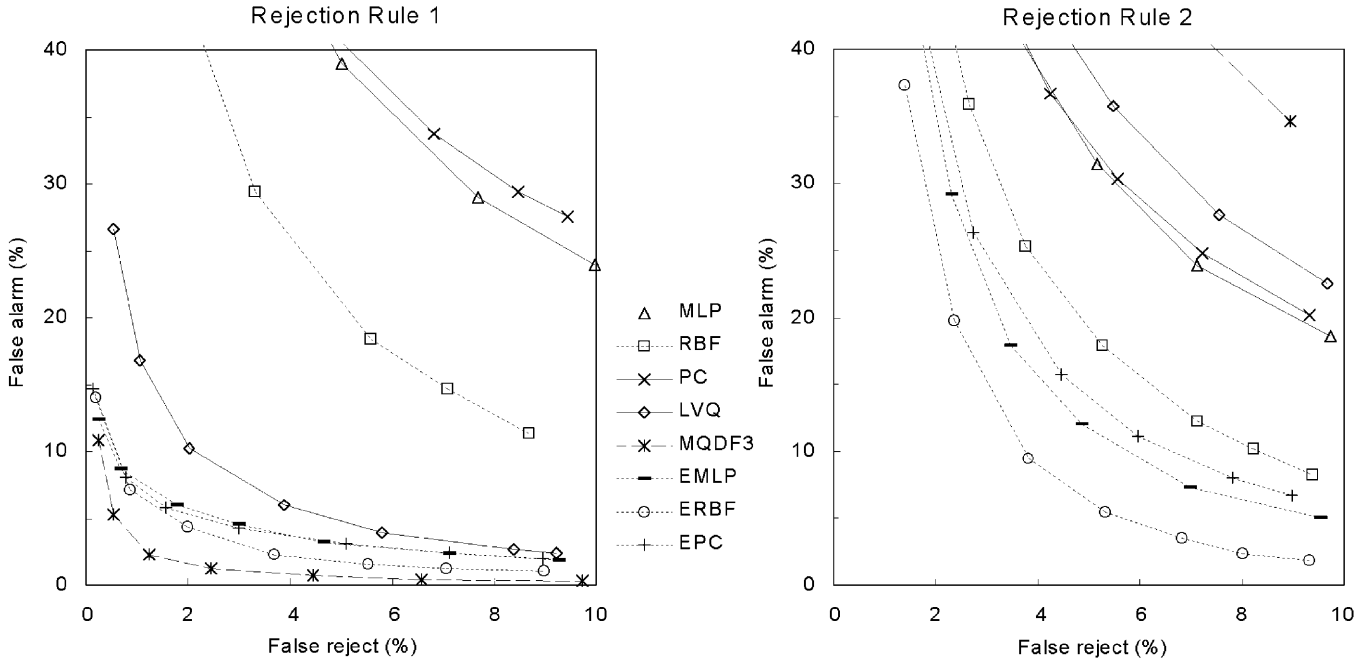

Fig. 7. False reject-alarm tradeoff on type 1 outliers

Table 3. Cumulative accuracies on the test data (%)

Classifier	Top rank	2 ranks	3 ranks
MQDF3	98.32	99.51	99.80
LVQ	98.80	99.71	99.91
MLP	98.89	99.72	99.90
EMLP	98.85	99.68	99.87
RBF	99.00	99.78	99.94
ERBF	99.04	99.77	99.92
PC	99.09	99.81	99.94
EPC	99.10	99.82	99.92

6.5 Rejection analysis

In this section we will show some examples of false acceptance of outliers and false rejection of characters by RR1. We will not proceed with the ambiguity rejection because it has been addressed in many previous works.

We analyse the outlier rejection performance of five classifiers, namely, MQDF3, LVQ, EMLP, ERBF, and EPC. By tuning the rejection threshold, we made the rejection rate of character patterns to be around 2%. The rejection rates and the false acceptance rates are listed in Table 4. We can see that at a similar rejection rate, the MQDF3 yields low acceptance rates to both type 1 and type 2 outliers. On the type 2 outliers, the LVQ classifier and the ERBF classifier also show good outlier resistance.

Some examples of falsely rejected digit patterns by the five classifiers are shown in Fig. 9, where each pattern is rejected by at least one classifier. The recognition result of each classifier is given in Fig. 9, and "-1" denotes rejection. We can see that the digit patterns rejected by RR1 are mostly contaminated by noise or unduly deformed.

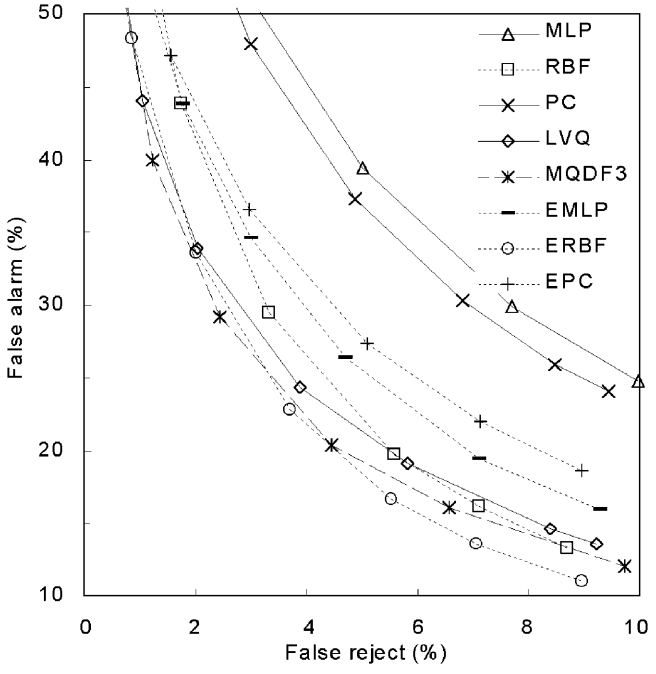

Fig. 8. False reject-alarm tradeoff on type 2 outliers

They are prone to be assigned low confidence to all hypothesized classes or misclassified to another class. Actually, some of the rejected dig it patterns can be viewed as mis-segmentation or mis-labelled patterns.

Some examples of falsely accepted outliers of type 1 and type 2 are shown in Figs. 10 and 11, respectively. They are accepted by at least one classifier. The recognition result of each classifier is given, and the blank cell denotes correct rejection. We can see that the pattern

Table 4. False rejection and false acceptance rates (%)

Classifier	False rej.	False acc. 1	False acc. 2
MQDF3	1.95	1.46	32.5
LVQ	2.04	10.06	33.9
EMLP	2.03	6.82	49.1
ERBF	2.01	4.35	33.5
EPC	2.02	5.15	42.6

"False acc. 1" and "False acc. 2" denote the false acceptance rates of type 1 outliers and type 2 outliers, respectively

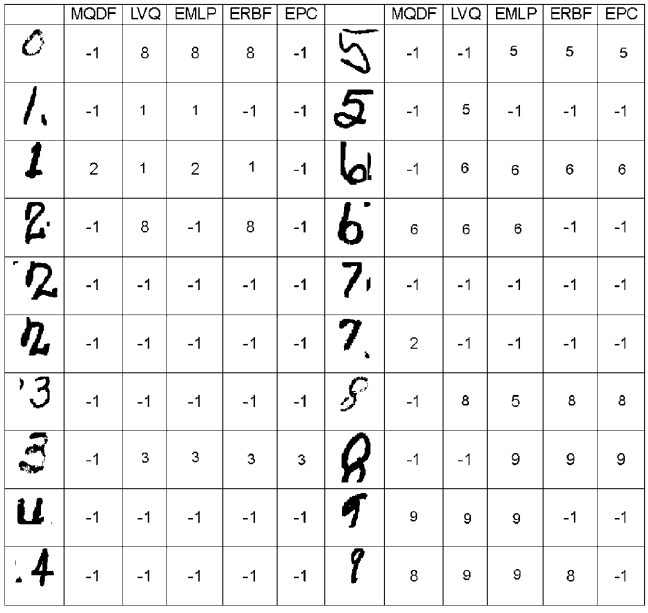

Fig. 9. Examples of false rejection of digit patterns

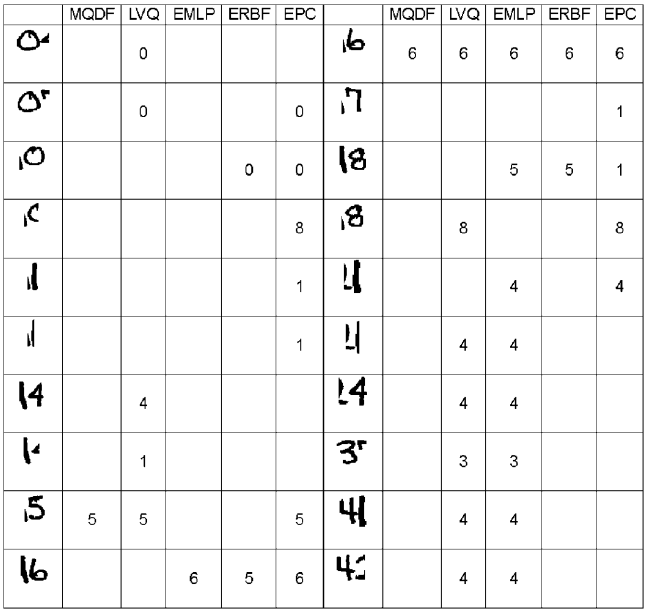

Fig. 10. Examples of false acceptance of type 1 outliers

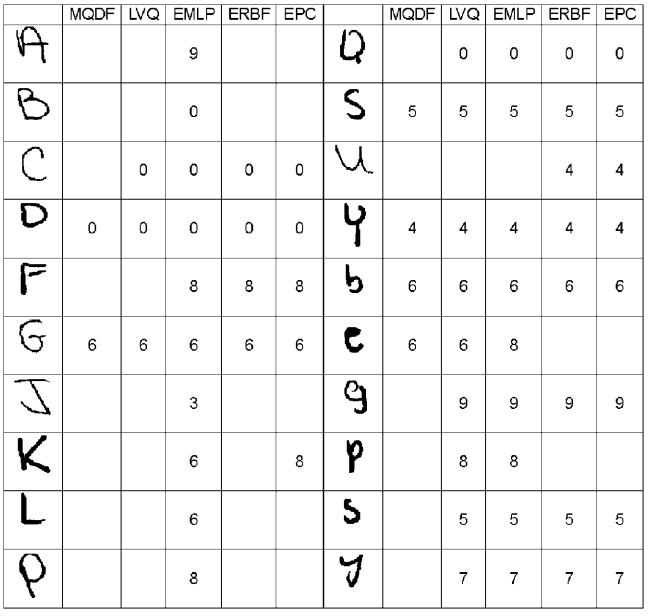

Fig. 11. Examples of false acceptance of type 2 outliers

shapes of the accepted outliers resemble digit patterns to a large extent. Some of them are assigned different classes by the classifiers. Thus, we assume that combining multiple classifiers by majority voting can also reduce the false acceptance of outliers.

Now we qualitatively explain the mechanism of outlier rejection. The MQDF is a density model in that its parameters are estimated from character data in a similar way to ML (maximum likelihood). LVQ can be viewed as a hybrid of a density model and a discriminative model because the restriction of prototype deviation from the sample data is connected to the principle of ML. With a density model, the patterns fitting this model are expected to have a high score while the outlier patterns are expected to have a low score. This is why the MQDF and the LVQ classifier perform well in outlier rejection. The promising outlier resistance of the ERBF classifier is also due to the hybrid nature of density model (the Gaussian kernels model the sample distribution) and discriminative learning. The EMLP and the EPC have the potential to give better outlier rejection performance if they are trained with a large set of outlier data of various shapes or an outlier-oriented learning algorithm is adopted.

7 Concluding remarks

We selected five classifiers (MQDF, MLP, the RBF classifier, PC, and the LVQ classifier) to evaluate their performance in handwritten digit recognition. These classifiers are efficient in the sense that they yield fairly high accuracies with moderate memory space and computation cost. The results show that their accuracies are comparable to or higher than the 1-NN rule while their complexity is much lower. The MQDF is a statistical classifier

and can be viewed as a density model, whereas the other four classifiers are discriminative models. It was shown in the experiments that the discriminative models give higher classification accuracies than the MQDF but are susceptible to small sample sizes. As a density model, the MQDF exhibits superior performance in outlier rejection even though it was not trained with outlier data. The outlier resistance of the neural classifiers was promoted by training with outlier data, yet the outlier rejection performance is still inferior to that of the MQDF.

Based on the nature of the classifiers and the experimental results, we can suggest some guidelines regarding choice of classifiers. This amounts to the choice between density models and discriminative models. The density model, such as the MQDF, is preferable for small sample sizes. It is also well scalable to large category problems such as Chinese character recognition. The training process is computationally cheap because the discriminant function of each class is learned independently. This also facilitates the increment/decrement of categories without re-training all categories. On the other hand, the discriminative models are appropriate when high accuracy is desired and a large sample size is available. This is generally true for small category problems.

The experimental results also reveal the insufficiencies of the classifiers and hence suggest some research directions. The density model gives low classification accuracy while the discriminative models are weak in outlier resistance. A desirable classifier should give both high accuracy and strong outlier resistance. This can be realized by hybridizing density models with discriminative models internally or combining them externally. Internal integration points to the design of new architectures and learning algorithms, whereas external combination points to the mixture of multiple experts.

Acknowledgements. The authors are grateful to Prof. George Nagy of Rensselaer Polytechnic Institute for reading through the manuscript and giving invaluable comments. The comments of the anonymous reviewers also led to significant improvements in the quality of this paper.

References

1. K. Fukunaga: Introduction to statistical pattern recognition. 2nd edn, Academic, New York (1990)
2. R.O. Duda, P.E. Hart, D.G. Stork: Pattern classification. 2nd edn, Wiley Interscience, New York (2000)
3. H. Friedman: Regularized discriminant analysis. J. Am. Stat. Assoc. 84(405):166–175 (1989)
4. F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake: Modified quadratic discriminant functions and the application to Chinese character recognition. IEEE Trans. Pattern Anal. Mach. Intell. 9(1):149–153 (1987)
5. F. Kimura, M. Shridhar: Handwritten numeral recognition based on multiple algorithms. Pattern Recognition 24(10):969–981 (1991)
6. D.E. Rumelhart, G.E. Hinton, R.J. Williams: Learning representations by back-propagation errors. Nature 323(9):533–536 (1986)
7. C.M. Bishop: Neural networks for pattern recognition. Clarendon, Oxford (1995)
8. D.F. Specht: Probabilistic neural networks. Neural Networks 3:109–118 (1990)
9. J. Schürmann: Pattern classification: a unified view of statistical and neural approaches. Wiley Interscience, New York (1996)
10. U. Kreßel, J. Schürmann: Pattern classification techniques based on function approximation. In: H. Bunke, P.S.P. Wang (eds.), Handbook of Character Recognition and Document Image Analysis, World Scientific, Singapore, 1997, pp. 49–78
11. T. Kohonen: Improved versions of learning vector quantization. Proc. 1990 Int. Joint Conf. Neural Networks, Vol.I, pp. 545–550
12. C.-L. Liu, M. Nakagawa: Evaluation of prototype learning algorithms for nearest neighbor classifier in application to handwritten character recognition. Pattern Recognition 34(3):601–615 (2001)
13. V. Vapnik: The nature of statistical learning theory. Springer, Berlin Heidelberg New York (1995)
14. C.J.C. Burges: A tutorial on support vector machines for pattern recognition. Knowl. Discovery Data Mining 2(2):1–43 (1998)
15. O.D. Trier, A.K. Jain, T.Taxt: Feature extraction methods for character recognition – a survey. Pattern Recognition 29(4):641–662 (1996)
16. J. Blue, et al: Evaluation of pattern classifiers for fingerprint and OCR applications. Pattern Recognition 27(4):485–501 (1994)
17. L. Holmström, et al: Neural and statistical classifiers – taxonomy and two case studies. IEEE Trans. Neural Networks 8(1): 5–17 (1997)
18. M.D. Garris, C.L. Wilson, J.L. Blue: Neural network-based systems for handprint OCR applications. IEEE Trans. Image Process. 7(8):1097–1112 (1998)
19. A. Shustorovich: A subspace projection approach to feature extraction: the two-dimensional Gabor transform for character recognition. Neural Networks 7(8):1295–1301 (1994)
20. J.T. Favata, G. Srikantan, S.N. Srihari: Handprinted character/digit recognition using a multiple feature/resolution philosophy. Proc. 4th Int. Workshop on Frontiers of Handwriting Recognition 1994, pp. 57–66
21. C.-L. Liu, Y-J. Liu, R-W. Dai: Preprocessing and statistical/structural feature extraction for handwritten numeral recognition. In: A.C. Downton, S. Impedovo (eds.) Progress of Handwriting Recognition, World Scientific, Singapore, 1997, pp. 161–168
22. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner: Gradient-based learning applied to document recognition. Proc. IEEE 86(11):2278–2324 (1998)
23. G.E. Hinton, P. Dayan, M. Revow: Modeling the manifolds of images of handwritten digits. IEEE Trans. Neural Networks 8(1):65–74 (1997)
24. E. Oja: Subspace methods of pattern recognition. Research Studies, Letchworth, UK (1983)
25. F. Kimura, et al.: Handwritten numeral recognition using autoassociative neural networks. Proc. 14th Int. Conf. Pattern Recognition, Brisbane, 1998, Vol.I, pp. 166–171
26. R.G. Casey, E. Lecolinet: A survey of methods and strategies in character segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 18(7):690–706 (1996)

27. H. Fujisawa, Y. Nakano, K. Kurino: Segmentation methods for character recognition: from segmentation to document structure analysis. Proc. IEEE 80(7):1079–1092 (1992)
28. Y. Saifullah, M.T. Manry: Classification-based segmentation of ZIP codes. IEEE Trans. System Man Cybernet. 23(5):1437–1443 (1993)
29. Y. LeCun, et al: Comparison of learning algorithms for handwritten digit recognition. Proc. Int. Conf. Artificial Neural Networks. F. Fogelman-Soulié, P. Gallinari (eds.), Nanterre, France, 1995, pp. 53–60
30. R.P.W. Duin: A note on comparing classifiers. Pattern Recognition Lett. 17:529–536 (1996)
31. F. Kimura, M. Shridhar: Segmentation-recognition algorithm for zip code field recognition. Mach. Vision Appl. 5:199–210 (1992)
32. F. Kimura, Y. Miyake, M. Sridhar: Handwritten ZIP code recognition using lexicon free word recognition algorithm. Proc. 3rd Int. Conf. Document Analysis and Recognition, 1995, pp. 906–910
33. F. Kimura, et al.: Evaluation and synthesis of feature vectors for handwritten numeral recognition. IEICE Trans. Inf. Syst. E79-D(5):436–442 (1996)
34. D. Wettschereck, T. Dietterich: Improving the performance of radial basis function networks by learning center locations. In: J.E. Moody, S.J. Hanson, R.P. Lippmann (eds.), Advances in Neural Information Processing Systems 4. Morgan-Kauffmann, San Francisco, 1992, pp. 1133–1140
35. L. Tarassenko, S. Roberts: Supervised and unsupervised learning in radial basis function classifiers. IEEE Proc. Vis. Image Signal Process. 141(4):210–216 (1994)
36. B.-H. Juang, S. Katagiri: Discriminative learning for minimization error classification. IEEE Trans. Signal Process. 40(12):3043–3054 (1992)
37. D.-S. Lee, S.N. Srihari: Handprinted digit recognition: a comparison of algorithms. Proc. 3rd Int. Workshop on Frontiers of Handwriting Recognition, 1993, pp. 153–164
38. S.W. Jeong, S.H. Kim, W.H. Cho: Performance comparison of statistical and neural network classifiers in handwritten digits recognition. In: S.-W. Lee (ed.), Advances in Handwriting Recognition, World Scientific, Singapore, 1999, pp. 406–415
39. A.K. Jain, R.P.W. Duin, J. Mao: Statistical pattern recognition: a review. IEEE Trans. Pattern Anal. Mach. Intell. 22(1):4–37 (2000)
40. T. Kohonen: Statistical pattern recognition with neural networks: benchmarking studies. Proc. 1988 Int. Joint Conf. Neural Networks, Vol. I, pp. 61–68
41. P.D. Gader, M. Mohamed, J.-H. Chiang: Handwritten word recognition with character and inter-character neural networks. IEEE Trans. Syst. Man Cyber. Part B: Cybern. 27(1):158–164 (1997)
42. P.D. Gader, et al.: Neural and fuzzy methods in handwriting recognition. IEEE Comp. 1997, pp. 79–86
43. M. Gori, F. Scarselli: Are multilayer perceptrons adequate for pattern recognition and verification? IEEE Trans. Pattern Anal. Mach. Intell. 20(11):1121–1132 (1998)
44. J. Bromley, J.S. Denker: Improving rejection performance on handwritten digits by training with rubbish. Neural Comput. 5:367–370 (1993)
45. J.-H. Chiang: A hybrid neural network model in handwritten word recognition. Neural Networks 11(2):337–346 (1998)
46. D.M.J. Tax, R.P.W. Duin: Outlier detection using classifier instability. In: A. Amin, et al. (eds.), Advances in Pattern Recognition: SSPR'98 & SPR'98, Springer, Berlin Heidelberg New York, 1998, pp. 593–601
47. C.Y. Suen, K. Liu, N.W. Strathy: Sorting and recognizing cheques and financial documents. In: S.-W. Lee, Y. Nakano (eds.), Document Analysis Systems: Theory and Practice, Springer, Berlin Heidelberg New York, 1999, pp. 173–187
48. G.L. Martin, M. Rashid: Recognizing overlapping handprinted characters by centered-object integrated segmentation and recognition. In: J.E. Moody, S.J. Hanson, R.P. Lippmann (eds.), Advances in Neural Information Processing Systems 4, Morgan Kaufmann, San Francisco, 1992, pp. 504–511
49. T.M. Ha, J. Zimmermann, H. Bunke: Off-line handwritten numeral string recognition by combining segmentation-based and segmentation-free methods. Pattern Recognition 31(3):257–272 (1998)
50. X. Zhu, Y. Shi, S. Wang: A new distinguishing algorithm of connected character images based on Fourier transform. Proc. 4th Int. Conf. Document Analysis and Recognition, 1999, pp. 788–791
51. T. Ha, H. Bunke: Off-line handwritten numeral recognition by perturbation method, IEEE Trans. Pattern Anal. Mach. Intell. 19(5):535–539 (1997)
52. C.-L. Liu, M. Koga, H. Sako, H. Fujisawa: Aspect ratio adaptive normalization for handwritten character recognition. In: T. Tan, Y. Shi, W. Gao (eds.), Advances in Multimodal Interfaces – ICMI 2000, Lecture Notes in Computer Science, vol. 1948. Springer, Berlin Heidelberg New York, 2000. pp. 418–425
53. T. Wakabayashi, S. Tsuruoka, F. Kimura, Y. Miyake: On the size and variable transformation of feature vector for handwritten character recognition. Trans. IEICE Japan J76-D-II(12):2495–2503 (1993)
54. F. Sun, S. Omachi, H. Aso: Precise selection of candidates for handwritten character recognition using feature regions. IEICE Trans. Inf. Syst. E79-D(5):510–515 (1996)
55. B. Moghaddam, A. Pentland: Probabilistic visual learning for object representation. IEEE Trans. Pattern Anal. Mach. Intell. 19(7):696–710 (1997)
56. C.K. Chow: On optimum error and reject tradeoff. IEEE Trans. Inform. Theory IT-16:41–46 (1970)
57. B. Dubuisson, M. Masson: A statistical decision rule with incomplete knowledge about classes. Pattern Recognition 26(1):155–165 (1993)
58. L.P. Cordella, C.D. Stefano, F. Tortorella, M. Vento: A method for improving classification reliability of multilayer perceptrons. IEEE Trans. Neural Networks 6(5):1140–1147 (1995)

Cheng-Lin Liu was born in Hunan Province, China, in 1967. He received his B.S. degree in electronics from Wuhan University, his M.E. degree in electronic engineering from Beijing Polytechnic University, and his Ph.D. degree in pattern recognition from the Institute of Automation, Chinese Academy of Sciences, in 1989, 1992, and 1995, respectively. He was a postdoctor fellow in Korea Advanced Institute of Science and Technology (KAIST) and later in Tokyo University of Agriculture and Technology from March 1996 to March 1999. Since

then he has been a researcher at the Central Research Laboratory, Hitachi, Tokyo, Japan. He received a technical award from Hitachi in 2001 for his contribution to the development of an address reading algorithm for mail sorting machines. His research interests include pattern recognition, artificial intelligence, image processing, neural networks, machine learning, and, especially, the application of these principles to handwriting recognition.

Hiroshi Sako received his B.E. and M.E. degrees in mechanical engineering from Waseda University, Tokyo, Japan, in 1975 and 1977, respectively. In 1992, he received his D.Eng. degree in computer science from the University of Tokyo. From 1977 to 1991, he worked in the field of industrial machine vision at the Central Research Laboratory of Hitachi, Tokyo, Japan. From 1992 to 1995, he was a senior research scientist at Hitachi Dublin Laboratory, Ireland, where he did research in the facial and hand-gesture recognition. Since 1996, he has been with the Central Research Laboratory of Hitachi, where he directs a research group of character recognition and image recognition. Dr. Sako was a recipient of the 1988 Best Paper Award from the Institute of Electronics, Information, and Communication Engineers (IEICE) of Japan for his paper on a real-time visual inspection algorithm of semiconductor wafer patterns, and one of the recipients of the Industrial Paper Awards from the 12th IAPR International Conference on Pattern Recognition, Jerusalem, Israel, 1994 for his paper on a real-time facial-feature tracking techniques. He has authored technical papers and patents in the area of pattern recognition, image processing, and neural networks. He is a member of IEEE, IEICE, JSAI, and IPSJ.

Hiromichi Fujisawa received his B.E., M.E., and D.Eng. degrees in Electrical Engineering from Waseda University, Japan, in 1969, 1971, and 1975, respectively. He joined the Central Research Laboratory, Hitachi in 1974. He has engaged in research and development works on OCR, document understanding, knowledge-based document retrieval, full-text search of Japanese documents, etc. Currently, he is a Senior Chief Researcher at the Central Research Laboratory, supervising researches on document understanding, including mailpiece address recognition and form recognition, and also information systems such as eGovernment. From 1980 through 1981, he was a visiting scientist at the Computer Science Department of Carnegie Mellon University, USA. Besides working at Hitachi, he was a guest lecturer at Waseda University from 1985 to 1997. Currently, he is a guest lecturer at Kogakuin University, Tokyo. He is a Fellow of the IAPR, a senior member of the IEEE, and a member of ACM, AAAI, the Information Processing Society of Japan (IPSJ), and the Institute of Electronics, Information and Communication Engineers (IEICE), Japan.