A NEW APPROACH TO MULTIPLE CLASS PATTERN CLASSIFICATION WITH RANDOM MATRICES

DONG Q. WANG AND MENGJIE ZHANG

Received 8 April 2003 and in revised form 2 December 2003

We describe a new approach to multiple class pattern classification problems with noise and high dimensional feature space. The approach uses a random matrix X which has a specified distribution with mean M and covariance matrix $r_{ij}(\Sigma_s + \Sigma_\epsilon)$ between any two columns of X. When Σ_ϵ is known, the maximum likelihood estimators of the expectation M, correlation Γ, and covariance Σ_s can be obtained. The patterns with high dimensional features and noise are then classified by a modified discriminant function according to the maximum likelihood estimation results. This new method is compared with a multilayer feed forward neural network approach on nine digit recognition tasks of increasing difficulty. Both methods achieved good results for those classification tasks, but the new approach was more effective and more efficient than the neural network method for difficult problems.

1. Introduction

This paper presents a new approach to the multiple class pattern classification process. Consider the situation where the multivariate observation $\vec{x}_i$ with p dimensions on an object consists of two independent components $\vec{y}_i$ and $\vec{\varepsilon}_i$ $(i = 1, 2, \ldots, n)$, where n is total sample size. Let $\vec{y}_i$ have a p-dimensional multivariate normal distribution with mean vector $\vec{\mu}$ and covariance matrix Σ_s, while $\vec{\varepsilon}_i$ has a multivariate normal distribution with mean vector $\vec{0}$ and covariance Σ_ε.

If we let $\vec{x}_i = \vec{y}_i + \vec{\varepsilon}_i$ and the vector $\vec{x}_i$ has a multivariate normal distribution with mean vector $\vec{\mu}$, then the covariance structure between $\vec{x}_i$ and $\vec{x}_j$ is constructed by

$$\operatorname{cov}(\vec{x}_i, \vec{x}_j) = \gamma_{ij}(\Sigma_s + \Sigma_\varepsilon), \tag{1.1}$$

where $\gamma_{ij} = \gamma_{ji}$ is the correlation between $\vec{x}_i$ and $\vec{x}_j$ and $\gamma_{ii} = 1$ if $i = j$. Thus the probability density function of the observation matrix $X_{p\times n} = (\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n)$ is given by Wang and

Journal of Applied Mathematics and Decision Sciences 2005:3 (2005) 165–175
DOI: 10.1155/JAMDS.2005.165

Lawoko [1]:

$$f(X \mid M,\Sigma_s,\Sigma_\varepsilon,\Gamma) = (2\pi)^{-np/2}\,|\Sigma_s+\Sigma_\varepsilon|^{-n/2}|\Gamma|^{-p/2}\exp\left\{-\frac{1}{2}\operatorname{tr}\left[(\Sigma_s+\Sigma_\varepsilon)^{-1}(X-M)\Gamma^{-1}(X-M)'\right]\right\}, \tag{1.2}$$

where $M=\mu\mathbf{1}'$, $\Gamma=\{\gamma_{rs}\}$ is the correlation matrix, which is estimated by sample correlation matrix for X. The term $\operatorname{tr}(\cdot)$ denotes the trace of a matrix, and $\mathbf{1}$ is a column vector of unit elements. It is assumed that Σ_s, Σ_ε and Γ are positive definite matrices. Note that in pattern recognition $\vec{y_i}$ and $\vec{\varepsilon_i}$ are referred to as "signal" and "noise," respectively. With the model, the observation vector X can be decomposed into independent signal and noise components, and its covariance matrix can be written as $\Gamma\otimes(\Sigma_s+\Sigma_\varepsilon)$, where $\otimes$ denotes the Kronecker product.

2. Estimation of parameters

2.1. Estimation of covariance matrix Σ_s. With the assumptions that Γ and Σ_ε are known and $n\gg p$ in (1.2), Wang and Lawoko [1] proved the following results for normally distributed random matrices.

(a) The maximum likelihood estimators of Σ_ϵ and M are obtained as

$$\begin{aligned}\hat{\Sigma}_s &= n^{-1}(X-\hat{M})\Gamma^{-1}(X-\hat{M})'-\Sigma_\varepsilon\\ \hat{M} &= \hat{\mu}\mathbf{1}' = (\mathbf{1}'\Gamma^{-1}\mathbf{1})^{-1}X\Gamma^{-1}\mathbf{1}\mathbf{1}'.\end{aligned} \tag{2.1}$$

(b) If we let $\hat{\Sigma}_s^* = n(n-1)^{-1}\hat{\Sigma}_s+(n-1)^{-1}\Sigma_\varepsilon$, then Σ_s^* is an unbiased estimator of Σ_s and the covariance of Σ_s^* is given by

$$\operatorname{cov}(\hat{\Sigma}_s^*) = (n-1)^{-1}(\Sigma_s+\Sigma_\varepsilon)^2+(n-1)^{-1}\operatorname{tr}\left[(\Sigma_s+\Sigma_\varepsilon)\right](\Sigma_s+\Sigma_\varepsilon), \tag{2.2}$$

while

$$\operatorname{cov}(\hat{M}) = n(\mathbf{1}'\Gamma^{-1}\mathbf{1})^{-1}\Sigma_s+n(\mathbf{1}'\Gamma^{-1}\mathbf{1})^{-1}\Sigma_\varepsilon. \tag{2.3}$$

(c) The covariance matrix between the unbiased estimator $\hat{\Sigma}_s^*$ of Σ_s and $\hat{M}$ is calculated by

$$\operatorname{cov}\{\hat{M},\hat{\Sigma}_s^*\} = (\mathbf{1}'\Gamma^{-1}\mathbf{1})^{-1}\Gamma\mathbf{1}\mathbf{1}'\Gamma^{-1}(\Sigma_s+\Sigma_\varepsilon)CM'+M[2M'+(\mathbf{1}'\Gamma^{-1}\mathbf{1})^{-1}I]CM', \tag{2.4}$$

where

$$C = (n-1)^{-1}\Gamma^{-1}\left[I-(\mathbf{1}'\Gamma^{-1}\mathbf{1})^{-1}\mathbf{1}\mathbf{1}'\Gamma^{-1}\right]. \tag{2.5}$$

2.2. Estimation of correlation matrix Γ. Suppose that Γ is unknown in (1.2), the problem of Σ_s (and Σ_ε), which is only possible after the separation of "signal" from "noise," is considered for a specific model [2]. We now need to estimate the correlation matrix in

(1.2). The log-likelihood function from (1.2) is written as

$$\begin{aligned} L &= \operatorname{loglik}(M, \Sigma_s, \Gamma \mid X, \Sigma_\varepsilon) \\ &= \text{constant} + \left(\frac{n}{2}\right)\log|\Delta^{-1}| + \left(\frac{p}{2}\right)\log|\Gamma^{-1}| \\ &\quad - \left(\frac{1}{2}\right)\operatorname{tr}\left[H\Delta^{-1}H'(X-M)\Gamma^{-1}(X-M)'\right], \end{aligned} \tag{2.6}$$

where $\Delta = H'(\Sigma_s + \Sigma_\varepsilon)H$ and H is an orthogonal matrix. Differentiation of L with respect to Γ and Δ yields

$$\begin{aligned} dL = &\left\{\frac{n}{2}\operatorname{tr}(\Delta) - \frac{1}{2}\operatorname{tr}\left[H(X-M)\Gamma^{-1}(X-M)'H'\right]\right\}d\Delta^{-1} \\ &+ \left\{\frac{n}{2}\operatorname{tr}(\Gamma) - \frac{1}{2}\operatorname{tr}\left[(X-M)'H\Delta^{-1}H'(X-M)\right]\right\}d\Gamma^{-1}. \end{aligned} \tag{2.7}$$

Equating dL to zero, we obtain the equations

$$\begin{aligned} (X-M)\Gamma^{-1}(X-M) &= n\Sigma \\ (X-M)'\Sigma^{-1}(X-M) &= p\Gamma, \end{aligned} \tag{2.8}$$

where $\Sigma = \Sigma_s + \Sigma_\varepsilon$. Noting that

$$M = \left(\mathbf{1}'\Gamma^{-1}\mathbf{1}\right)^{-1}X\Gamma^{-1}\mathbf{1}\mathbf{1}', \tag{2.9}$$

we have the following matrix equation in Γ^{-1}

$$\begin{aligned} &X'X - \left(\mathbf{1}'\Gamma^{-1}\mathbf{1}\right)^{-1}X'X\Gamma^{-1}\mathbf{1}\mathbf{1}' - \left(\mathbf{1}'\Gamma^{-1}\mathbf{1}\right)^{-1}\mathbf{1}\mathbf{1}'\Gamma^{-1}X'X + \left(\mathbf{1}'\Gamma^{-1}\mathbf{1}\right)^{-2}\mathbf{1}\mathbf{1}'\Gamma^{-1}X'X\Gamma^{-1}\mathbf{1}\mathbf{1}' \\ &\quad = \mathbf{0}_{n\times n}, \end{aligned} \tag{2.10}$$

where $\mathbf{0}_{n\times n}$ denotes a zero matrix of dimension $n \times n$, and $\mathbf{1}$ is a column vector of unit elements.

From this we obtain the equation

$$\left[X'X - \left(\mathbf{1}'\Gamma^{-1}\mathbf{1}\right)^{-1}\mathbf{1}\mathbf{1}'\Gamma^{-1}X'X\right]\left[I_n - \left(\mathbf{1}'\Gamma^{-1}\mathbf{1}\right)^{-1}\mathbf{1}\mathbf{1}'\Gamma^{-1}\right]' = \mathbf{0}_{n\times n}, \tag{2.11}$$

so that

$$\left[I_n - \left(\mathbf{1}'\Gamma^{-1}\mathbf{1}\right)^{-1}\mathbf{1}\mathbf{1}'\Gamma^{-1}\right]X'X\left[I_n - \left(\mathbf{1}\Gamma^{-1}\mathbf{1}\right)^{-1}\mathbf{1}\mathbf{1}'\Gamma^{-1}\right]' = \mathbf{0}_{n\times n}. \tag{2.12}$$

If we let $Y = I_n - (\mathbf{1}'\Gamma^{-1}\mathbf{1})^{-1}\mathbf{1}\mathbf{1}'\Gamma^{-1}$ and $A = X'X$, the expression is a quadratic form, that is,

$$YAY' = \mathbf{0}_{n\times n}. \tag{2.13}$$

It is known that a solution $Y = Y^*$ can be obtained through numerical methods [3]. Once Y^* is known, we can solve for Γ^{-1} from

$$Y^* = I_n - \left(\mathbf{1}'\Gamma^{-1}\mathbf{1}\right)^{-1}\mathbf{1}\mathbf{1}'\Gamma^{-1}. \tag{2.14}$$

That is, the equation

$$\mathbf{1}\mathbf{1}'\Gamma^{-1} - \left(\mathbf{1}'\Gamma^{-1}\mathbf{1}\right)(I - Y^*) = \mathbf{0}_{n\times n} \tag{2.15}$$

is linear in Γ^{-1} and has a unique solution [4], provided that the coefficient matrix is nonsingular. Finally, the elements of Γ are obtained from Γ^{-1}. Note that the solution $\hat{\Gamma} = \{\hat{\gamma}_{ij}\}$ obtained from the above equations will not necessarily satisfy the usual properties of a true correlation matrix, which would be that

(1) Γ is a positive definite symmetric matrix.

(2) $|\gamma_{ij}| \le 1$, with $\gamma_{ii} = 1$ for all i.

In order to find a true correlation matrix $\tilde{\Gamma} = \{\tilde{\gamma}_{ij}\}$ which, on the basis of some measure, is as close as possible to $\hat{\Gamma}$, several methods can be used. Some of these techniques, like the "shrinking" and "eigenvalue" methods, are summarized in Rousseeuw and Molenberghs [5].

2.3. Generating a modified discriminant function (MDF). When Σ_ϵ is known and Σ_s and Γ are estimated as in Sections 2.1 and 2.2, the covariance matrix $\hat{\Gamma} \otimes (\hat{\Sigma}_s + \Sigma_\epsilon)$ of X can be obtained by maximum likelihood method. Then the modified discriminant function MDF for each class j is given by

$$\mathrm{MDF}_j(\vec{x}) = -\ln\left|\hat{\Gamma} \otimes (\hat{\Sigma}_s + \Sigma_\epsilon)\right| - (\vec{x} - \vec{\bar{x}}_j)^T\left(\hat{\Gamma} \otimes (\hat{\Sigma}_s + \Sigma_\epsilon)\right)^{-1}(\vec{x} - \vec{\bar{x}}_j), \tag{2.16}$$

where $j = 1, 2, \ldots, c$; $\vec{\bar{x}}_j$ is an estimator of the sample mean vector of the jth cluster of random matrix X, and c is the number of classes. Note that we use the covariances $\hat{\Gamma} \otimes (\hat{\Sigma}_s + \Sigma_\epsilon)$ in MDF.

Equation (2.16) can be further simplified to give a linear modified discriminant function (LMDF) as follows:

$$\begin{aligned}\mathrm{LMDF}_j(\vec{x}) = {} & -\ln\left|\hat{\Gamma} \otimes (\hat{\Sigma}_s + \Sigma_\epsilon)\right| - \vec{\bar{x}}_j^T\left(\hat{\Gamma} \otimes (\hat{\Sigma}_s + \Sigma_\epsilon)\right)^{-1}\vec{\bar{x}}_j \\ & + \vec{\bar{x}}_j^T\left(\hat{\Gamma} \otimes (\hat{\Sigma}_s + \Sigma_\epsilon)\right)^{-1}\vec{x}.\end{aligned} \tag{2.17}$$

The classification rule is given as follows: assign any given observation, $\vec{x}^*$, if $\mathrm{LMDF}_i(\vec{x}^*) \ge \mathrm{LMDF}_j(\vec{x})$, for all $j \ne i$, then the item $\vec{x}^*$ is assigned class i. Classification functions of linear modified discriminant analysis assume equal variance-covariance matrices for all the groups and a multivariate normal distribution. Note that we use the central limit theorem, where the total noise can be approximated as Gaussian or normal distribution. The Polar method can be used for generating noise, and relies on having good uniform random number generator. We describe the method to generate noise in Section 3, where we use it to adjust the mean and variance of signals for making digit recognition with classification problems of increasing difficulty.

Figure 3.1. Digit pattern examples with different rates of noise.

Also note that the MDF approach differs from the method of Support Vector Machines presented in Duda et al. [6]. Their method is a special case in (2.16) and (2.17) where the random samples are correlated with a covariance $\Gamma \otimes (\Sigma_s + \Sigma_\epsilon)$, and classification functions can be also used to develop discriminant regions.

3. Numerical example

3.1. Simulation data sets. To investigate the power of our new approach, we used nine multiple dimension digit recognition tasks in the experiments. Each task involves a file (a collection) of binary digit images. Each file contains 100 examples for each of the 10 digits $(0,1,\ldots,9)$, making a total number of 1000 digit examples. Each digit example is an image of 7×7 bitmap. These tasks were chosen to provide classification problems of increasing difficulty, as shown in Table 3.1. In all of these recognition problems, the goal is to automatically recognize which of the 10 classes (digits $0,1,2,\ldots,9$) each pattern (digit example) belongs to. Except for the first file which contains clean patterns, all data patterns in the other eight files have been corrupted by noise. The amount of noise in different files was randomly generated based on the percentage of flipped pixels and was given by the two numbers nn in the file name. For example, the first row of this table shows that, recognition Task 1 is to classify those clean digit patterns into the ten different classes. In this task, there are 1000 patterns in total, 500 are used for training and 500 for testing. In Task 3, 10% of pixels, chosen at random, have been flipped. Before starting the training/learning process, all the training examples are randomly ordered.

Examples of the 9 tasks are shown in Figure 3.1. The 9 lines of digit examples correspond to the 9 recognition tasks in Table 3.1. The first 3 tasks, one with clean data and two with only 5% and 10% of flipped rate, are relatively straightforward for human eyes, though there is still some difficulty in distinguishing between "3" and "9." With the increase of the flipped rate in these patterns such as Task 4 and Task 5, it becomes more

Table 3.1. Nine digit recognition tasks.

Task	File name	Noise amount	Total patterns	Training set	Test set
1	digit00	0%	1000	500	500
2	digit05	5%	1000	500	500
3	digit10	10%	1000	500	500
4	digit15	15%	1000	500	500
5	digit20	20%	1000	500	500
6	digit30	30%	1000	500	500
7	digit40	40%	1000	500	500
8	digit50	50%	1000	500	500
9	digit60	60%	1000	500	500

Table 3.2. Results for optimal error rate for Task 6.

Digit	0	1	2	3	4	5	6	7	8	9
OER	0.00	0.00	0.00	0.26	0.00	0.08	0.16	0.00	0.20	0.26
Prior Probability	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10

difficult to classify these digit patterns, even if humans can still recognize the majority. From Task 6 to Task 9, however, it is very difficult, even impossible, for human eyes to make good discrimination. We hypothesized that our new method will do a good job for the first three tasks, but cannot be excellent for Tasks 6 to 9. We also want to investigate whether our new method can achieve an acceptable performance for these difficult tasks and whether the new method outperforms a neural network approach (Section 4) on these tasks.

3.2. An MDF classification example. This subsection uses an example to briefly describe how to obtain the classification error for each task by applying the new method (2.16). After applying (2.17) to each task, the conferences of discriminant function can be obtained. In Task 6 in Table 3.1, for example, there is 30% of noise flipped in the 1000 digits. The discriminant function for digit "2" is:

$$\begin{aligned} \mathrm{LMDF}_2(\vec{x}) = {} & -9.24\times 10^8 + 4.55x_1 + 4.41x_2 + 4.48x_3 + 7.83x_4 + \cdots \\ & + 6.57x_{47} + 6.580x_{48} + 1.85\times 10^9 x_{49}, \end{aligned} \tag{3.1}$$

where vector $\vec{x} = (x_1, x_2, x_3, \ldots, x_{48}, x_{49})^T$.

Task 6 classification results for test data are summarized in Table 3.2. The total optimal error rate (OER) for this task is 0.106, or the classification accuracy is 89.40% on average of 10 runs.

Table 4.1. Example patterns used in neural networks for Task 1.

Class	Input pattern	Output pattern
0	00111000100010100000110000011000001010001000111100	1000000000
1	00010000011000010100000010000001000000100000111110	0100000000
2	01111101000001000000101111101000000100000001111111	0010000000
3	01111101000001000000101111100000001100000010111110	0001000000
4	10000001000010100001010000101111111000001000000010	0000100000
5	11111111000000100000011111100000001100000010111110	0000010000
6	01111101000001100000011111101000001100000010111110	0000001000
7	11111111000010000010000010000010000001000000010000	0000000100
8	01111101000001100000101111101000001100000010111110	0000000010
9	01111101000001100000101111110000001100000010111110	0000000001

4. The neural networks approach

This section briefly describes the neural network approach to this problem. This approach involves the following steps: determination of the neural network architecture, network training and network testing for classification.

4.1. Network architecture. Multilayer feed forward neural networks have been proved to be suitable for classification and prediction problem [7, 8, 9, 10, 11]. In this approach, we use a three layer network (with a single hidden layer) to perform the digit recognition problems. The task then becomes determining the number of input nodes, the number of output nodes and the number of hidden nodes.

To avoid feature selection and hand-crafting of feature extraction programs, we directly used the raw pixels as inputs to neural networks. Since each digit example in our recognition tasks is a 7×7 bitmap, we used 49 input nodes in the network architecture. The ten classes of digits, from 0 to 9, form the output nodes in the network. Example patterns containing input patterns and corresponding output patterns for the ten classes of digits for the first digit recognition task are shown in Table 4.1.

The number of hidden nodes was determined by an empirical search method of "trial and error" during network training. We have found that 10–20 hidden nodes were suitable for these classification problems and that the process was relatively robust using these number of hidden nodes. An example neural network architecture with a non-flipped bitmap pattern from class "0" in Task 1 is shown in Figure 4.1.

4.2. Network training and testing. We used the back error propagation algorithm [12] with the following two variations to train the network.

(i) *Online learning.* Rather than updating the weights after presenting all the examples in a full epoch, we update the weights after presenting each bit map pattern.

(ii) *Fan-in.* Weight initialization and weight changes are modified by the *fan-in* factor. The weights are divided by the number of inputs of a node (referred to as the *fan-in* factor of the node) before network training and the size of the weight change of a node is updated accordingly during network training.

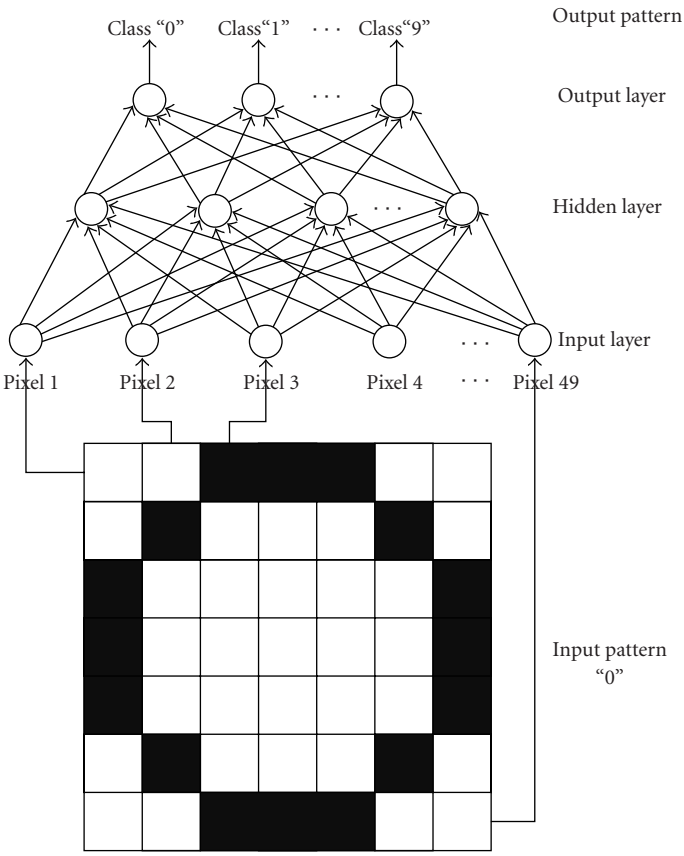

Figure 4.1. An example neural network associated with a bit map pattern "0."

During network training and testing, the network classification is considered correct if the largest activation value produced by the neural network is for the output node which corresponds to the target class. Otherwise, the classification is incorrect. For example, if the actual activation values of all the output nodes for a given digit pattern is $(0.32, 0.12, 0.45, 0.85, 0.23, 0.21, 0.13, 0.15, 0.33, 0.45)$ and the target output pattern is "0001000000," then this digit was correctly classified as digit "3" by the network; if the target output pattern is "0000000001," then this digit ("9") was incorrectly classified as digit "3" by the network.

5. Results and discussion

This section describes a comparison of the experimental results of the modified discriminant analysis method and the neural network method. For the neural network approach, we used a network architecture of 49-15-10. The network was trained with a learning rate of 0.5, without momentum. For the MDF approach, we used a priori probability of 0.1

Table 5.1. Results for different tasks and methods.

Recognition task	Recognition Accuracy $\mu \pm \sigma(\%)$	
	Neural Networks	MDF
(1) digits00	100 ± 0	100 ± 0
(2) digits05	99.00 ± 0.155	99.00 ± 0.012
(3) digits10	96.06 ± 0.156	96.40 ± 0.011
(4) digits15	94.26 ± 0.220	94.60 ± 0.017
(5) digits20	91.44 ± 0.310	91.60 ± 0.008
(6) digits30	89.10 ± 0.361	89.40 ± 0.015
(7) digits40	80.18 ± 0.384	81.40 ± 0.005
(8) digits50	75.86 ± 0.451	76.80 ± 0.029
(9) digits60	58.70 ± 0.677	63.60 ± 0.017

for all the ten classes. The results of the two methods on the unseen data in the test set for all the nine tasks are shown in Table 5.1. For both approaches, the training and testing are repeated 10 times and the average results (mean μ and standard deviation σ) on the test set are presented and compared.

As can be seen from Table 5.1, both methods achieved quite promising results on all of the nine tasks. On the clean data (Task 1), both methods achieved 100% accuracy. On the data sets with different noisy (flipped) rates ranging from 5% to 60% (Tasks 2 to 9), the new MDF method always achieved a higher *mean* recognition accuracy and a lower *standard deviation* than the neural network method, which suggests that the new MDF approach is more robust and more stable for these tasks. As expected, the performances from both approaches deteriorated for the recognition tasks of increasing difficulty.

5.1. Analysis. After further checking the results, we found that misclassification mainly came from the digit patterns of "3," "6," "8," and "9." As can be seen from Table 3.2, the optimal error rates were quite big (0.26, 0.16, 0.20 and 0.26) for these digits but quite small for other digits. After we checked these digit patterns from Figure 3.1, this was not surprising. On the relatively clean data patterns (Tasks 1 or 2), these digits are very similar (the gap is only two or three pixels). On the flipped digit patterns, some of them are more similar and even human eyes cannot distinguish between them. It is quite promising that both approaches achieved such good results.

5.2. Training time and preparation time. In terms of training efficiency, the MDF approach was also better than neural networks for these tasks. While training time for the MDF method was only about 2 seconds for each task for a single run, it took about 20–30 seconds to train the network on average. Furthermore, there is a major disadvantage for the use of neural networks. It is often very time consuming to determine an appropriate number of hidden nodes in the network architecture and a set of good learning parameters, which usually involves an empirical search in the experiments. The MDF approach, however, only needs to set the prior probabilities for different classes, which is relatively

easy since they can be usually set equally. Thus the MDF method generally takes a shorter preparation time and a shorter training time for the same problem.

6. Conclusions and further work

The goal of this paper was to develop an effective and efficient approach for high dimension, multiple class, noisy pattern classification problems. This goal was achieved by developing a noisy factor and a modified discriminant function in our discriminant approach. The second goal was to investigate whether this approach could do a good enough job for those problems on both clean data and noisy data. Nine digit recognition tasks of increasing difficulty were used as examples in the experiments. A neural network classifier was also developed for the purpose of comparison.

The results suggest that both the MDF approach and the neural network approach did a very good job for the noisy data. On all the 8 noisy tasks presented in this paper, the new MDF approach always achieved better classification performances than the neural network method. Furthermore, it was also more stable, took a shorter preparation time and a much shorter training time than the neural network method. As expected, the performance from both approaches deteriorated as the degree of difficulty of the recognition problems was increased.

Also, both the MDF approach and the neural network approach performed quite well on the very noisy tasks. This is inconsistent with our hypothesis, which did not expect them to achieve good results. This suggests that our new method and the neural network classifier are better than human eyes on these multivariant types of tasks.

To further investigate the power of the MDF method, we will apply it to other classification problems with multiple dimension, noisy data in the future.

Acknowledgments

We would like to thank Megan Clark for helpful comments. Thanks also go to Peter Andreae for a number of useful discussions.

References

[1] D. Q. Wang and C. R. O. Lawoko, *Estimation of parameters for normally distributed random matrices*, Linear Algebra Appl. **210** (1994), no. 1, 199–208.

[2] P. Switzer, *Min/max autocorrelation factors for multivariate spatial imagery*, Computer Science and Statistics: The Interface (L. Billard, Ed.), chapter 13–16, North Holland, Amsterdam, 1985.

[3] J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall Series in Computational Mathematics, Prentice-Hall, New Jersey, 1983.

[4] F. A. Graybill, *Introduction to Matrices with Applications in Statistics*, Wadsworth, California, 1969.

[5] P. J. Rousseeuw and G. Molenberghs, *Transformation of non-positive semidefinite correlation matrix*, Comm. Statist. Theory Methods **22** (1993), no. 4, 965–984.

[6] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed., John Wiley & Sons, New York, 2000.

[7] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*, Springer Series in Statistics, Springer, New York, 2001.

[8] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, *Handwritten zip code recognition with a back-propagation network*, Advances in Neural Information Processing Systems (D. S. Touretzky, Ed.), vol. 2, Morgan Kaufmann, California, 1990, pp. 323–331.

[9] M. W. Roth, *Survey of neural network technology for automatic target recognition*, IEEE Trans. Neural Networks **1** (1990), no. 1, 28–43.

[10] P. Winter, S. Sokhansanj, H. C. Wood, and W. Crerar, *Quality assessment and grading of lentils using machine vision*, Agricultural Institute of Canada Annual Conference, Canadian Society of Agricultural Engineering, Saskatoon, SK S7N 5A9, July 1996, CASE paper No. 96-310.

[11] W. Yonggwan, P. D. Gader, and P. C. Coffield, *Morphological shared-weight networks with applications to automatic target recognition*, IEEE Trans. Neural Networks **8** (1997), no. 5, 1195–1203.

[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning internal representations by error propagation*, Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Volume 1: Foundations (D. E. Rumelhart, J. L. McClelland, and the PDP research group, Eds.), chapter 8, MIT Press, Massachusetts, 1986, pp. 318–362.

Dong Q. Wang: School of Mathematics, Statistics, and Computer Science, Victoria University of Wellington, Wellington 6001, New Zealand

E-mail address: dong.wang@mcs.vuw.ac.nz

Mengjie Zhang: School of Mathematics, Statistics, and Computer Science, Victoria University of Wellington, Wellington 6001, New Zealand

E-mail address: mengjie.zhang@mcs.vuw.ac.nz