PERGAMON

Pattern Recognition 36 (2003) 2429–2439

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition

Xuechuan Wang*,[1], Kuldip K. Paliwal

School of Microelectronical Engineering, Nathan Campus, Griffith University, Brisbane, Qld 4111, Australia

Received 18 March 2002; accepted 16 December 2002

Abstract

Feature extraction is an important component of a pattern recognition system. It performs two tasks: transforming input parameter vector into a feature vector and/or reducing its dimensionality. A well-defined feature extraction algorithm makes the classification process more effective and efficient. Two popular methods for feature extraction are linear discriminant analysis (LDA) and principal component analysis (PCA). In this paper, the minimum classification error (MCE) training algorithm (which was originally proposed for optimizing classifiers) is investigated for feature extraction. A generalized MCE (GMCE) training algorithm is proposed to mend the shortcomings of the MCE training algorithm. LDA, PCA, and MCE and GMCE algorithms extract features through linear transformation. Support vector machine (SVM) is a recently developed pattern classification algorithm, which uses non-linear kernel functions to achieve non-linear decision boundaries in the parametric space. In this paper, SVM is also investigated and compared to linear feature extraction algorithms.

Keywords: Feature extraction; Dimensionality reduction; MCE; SVM

1. Introduction

The aim of pattern recognition systems is to classify input data as one of K classes. Conventional pattern recognition systems have two components: feature analysis and pattern classification, as shown in Fig. 1. Feature analysis is achieved in two steps: parameter extraction step and feature extraction step. In the parameter extraction step, information relevant for pattern classification is extracted from the input data in the form of a p-dimensional parameter vector x. In the feature extraction step, the parameter vector x is transformed to a feature vector y, which has a dimensionality m $(m \leqslant p)$. If the parameter extractor is properly designed so that the parameter vector x is matched to the pattern classifier and its dimensionality is low, then there is no necessity for the feature extraction step. However in practice, some parameter vectors are not suitable for pattern classifiers. For example, parameter vectors have to be decorrelated before applying them to a classifier based on Gaussian mixture models (with diagonal variance matrices). Furthermore, the dimensionality of parameter vectors is normally very high and needs to be reduced for the sake of less computational cost and system complexity. Due to these reasons, feature extraction has been an important problem in pattern recognition tasks.

Feature extraction can be conducted independently or jointly with either parameter extraction or classification. LDA and PCA are the two popular independent feature extraction methods. Both of them extract features by projecting the original parameter vectors into a new feature space through a linear transformation matrix. But they optimize the transformation matrix with different intentions. PCA optimizes the transformation matrix by finding the largest

* Corresponding author. Tel.: (514)8751266ext.3045.
E-mail addresses: wang@me.gu.edu.au, wwang@inrs-telecom.uquebec.ca (X. Wang), k.paliwal@me.gu.edu.au (K.K. Paliwal).

[1] Current address: INRS Telecommunication, Place Bonaventure, 800 de la Gauchetiere West, Level C, Suite 6900, Montreal, Quebec, Canada, H5A 1K6.

0031-3203/03/$30.00 © 2003 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/S0031-3203(03)00044-X

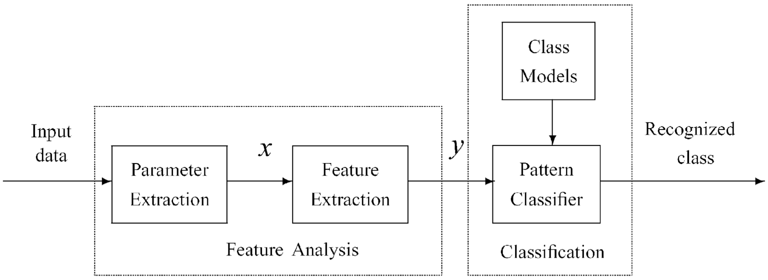


Fig. 1. Conventional pattern recognition system.

variations in the original feature space [1–3]. LDA pursues the largest ratio of *between*-class variation and *within*-class variation when projecting the original feature to a subspace [4–6]. The drawback of independent feature extraction algorithms is that their optimization criteria are different from the classifier's minimum classification error criterion, which may cause inconsistency between feature extraction and the classification stages of a pattern recognizer and consequently, degrade the performance of classifiers [7]. A direct way to overcome this problem is to conduct feature extraction and classification jointly with a consistent criterion. MCE training algorithm [7–9] provides such an integrated framework. It is a type of discriminant analysis but achieves minimum classification error directly when extracting features. This direct relationship has made MCE training algorithm widely popular in a number of pattern recognition applications, such as dynamic time-wrapping based speech recognition [10,11] and hidden Markov model (HMM) based speech and speaker recognition [12–14].

LDA, PCA and MCE training algorithm are linear feature extraction algorithms. The advantage of linear algorithms is their ability to reduce feature dimensionalities. However, they have the limitation that the decision boundaries generated are linear and have little computational flexibility. SVM is a recently developed pattern classification algorithm with non-linear formulation. It is based on the idea that the classification that affords dot-products can be computed efficiently in higher dimensional feature spaces [15–17]. The classes which are not linearly separable in the original parametric space can be linearly separated in the higher dimensional feature space. Because of this, SVM has the advantage that it can handle the classes with complex non-linear decision boundaries. Different from conventional systems as shown in Fig. 1, SVM is a highly integrated pattern recognition system as shown in Fig. 2. SVM has now evolved into an active area of research [18–21].

This paper investigates LDA, PCA and MCE algorithms for feature extraction. A generalized MCE (GMCE) training algorithm is proposed to mend the shortcomings of the MCE training algorithms. The performances of MCE and GMCE training algorithms are compared to those of LDA and PCA on both Deterding and TIMIT databases. SVM is investigated in this paper as an integrated pattern classification system. Its performance is compared to those of LDA, PCA, MCE algorithms.

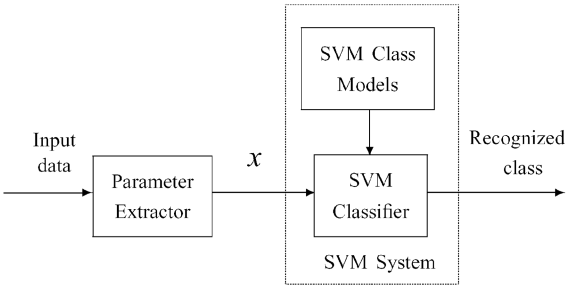


Fig. 2. SVM recognition system.

The rest of this paper is organized as follows: Section 2 gives a brief introduction to LDA and PCA. Section 3 introduces the framework of MCE training algorithm. An alternative MCE training algorithm, which uses a ratio model of misclassification measure, is proposed. MCE training algorithm is applied to dimensionality reduction tasks and a generalized MCE (GMCE) training algorithm is proposed. Section 4 introduces the formulation of SVM . In Section 5 we compare LDA, PCA, MCE, GMCE training algorithms and SVM in a vowel recognition task on TIMIT database. The results are analyzed and compared.

2. Standard feature extraction methods

2.1. Linear discriminant analysis

The goal of linear discriminant analysis is to separate the classes by projecting classes' samples from p-dimensional space onto a finely orientated line. For a K-class problem, $m=\min(K-1,p)$ different lines will be involved. Thus the projection is from a p-dimensional space to a c-dimensional space [22].

Suppose we have K classes, $\mathscr{X}_1,\mathscr{X}_2,\dots,\mathscr{X}_K$. Let the ith observation vector from the $\mathscr{X}_j$ be x_{ji}, where $j=1,\dots,J$ and $i=1,\dots,N_j$. J is the number of classes and N_j is the number of observations from class j. The *within*-class covariance matrix $S_{\rm W}$ and *between*-class covariance matrix $S_{\rm B}$ are defined as

$$S_{\rm W} = \sum_{j=1}^{K} S_j = \sum_{j=1}^{K} \frac{1}{N_j} \sum_{i=1}^{N_j} (x_{ji} - \mu_j)(x_{ji} - \mu_j)^{\rm T},$$

$$S_{\rm B} = \sum_{j=1}^{K} N_j(\mu_j - \mu)(\mu_j - \mu)^{\rm T}, \tag{1}$$

where $\mu_j = 1/N_j \sum_{i=1}^{N_j} x_{ji}$ is the mean of class j and $\mu = 1/N \sum_{i=1}^{N} x_i$ is the global mean.

The projection from observation space to feature space is accomplished by a linear transformation matrix T:

$$y = T^{\rm T} x. \tag{2}$$

The corresponding *within*-class and *between*-class covariance matrices in the feature space are:

$$\tilde{S}_{\rm W} = \sum_{j=1}^{K} \sum_{i=1}^{N_j} (y_{ji} - \tilde{\mu}_j)(y_{ji} - \tilde{\mu}_j)^{\rm T},$$

$$\tilde{S}_{\rm B} = \sum_{j=1}^{K} N_j(\tilde{\mu}_j - \tilde{\mu})(\tilde{\mu}_j - \tilde{\mu})^{\rm T}, \tag{3}$$

where $\tilde{\mu}_j = 1/N_j \sum_{i=1}^{N_j} y_{ji}$ and $\tilde{\mu} = 1/N \sum_{i=1}^{N} \tilde{y}_i$. It is straightforward to show that:

$$\tilde{S}_{\rm W} = T^{\rm T} S_{\rm W} T,$$

$$\tilde{S}_{\rm B} = T^{\rm T} S_{\rm B} T. \tag{4}$$

A *linear discriminant* is then defined as the linear functions for which the objective function

$$J(T) = \frac{|\tilde{S}_{\rm B}|}{|\tilde{S}_{\rm W}|} = \frac{|T^{\rm T} S_{\rm B} T|}{|T^{\rm T} S_{\rm W} T|} \tag{5}$$

is maximum. It can be shown that the solution of Eq. (5) is that the ith column of an optimal T is the generalized eigenvector corresponding to the ith largest eigenvalue of matrix $S_{\rm W}^{-1} S_{\rm B}$ [6].

2.2. Principal component analysis

PCA is a well-established technique for feature extraction and dimensionality reduction [23,2]. It is based on the assumption that most information about classes is contained in the directions along which the variations are the largest. The most common derivation of PCA is in terms of a standardised linear projection which maximises the variance in the projected space [1]. For a given p-dimensional data set $\mathscr{X}$, the m principal axes $T_1, T_2, \ldots, T_m$, where $1 \leqslant m \leqslant p$, are orthonormal axes onto which the retained variance is maximum in the projected space. Generally, $T_1, T_2, \ldots, T_m$ can be given by the m leading eigenvectors of the sample covariance matrix $S = 1/N \sum_{i=1}^{N} (x_i - \mu)^{\rm T}(x_i - \mu)$, where $x_i \in \mathscr{X}$, μ is the sample mean and N is the number of samples, so that:

$$S T_i = \lambda_i T_i, \quad i \in 1, \ldots, m, \tag{6}$$

where λ_i is the ith largest eigenvalue of S. The m principal components of a given observation vector $x \in \mathscr{X}$ are given by

$$y = [y_1, \ldots, y_m] = [T_1^{\rm T} x, \ldots, T_m^{\rm T} x] = T^{\rm T} x. \tag{7}$$

The m principal components of x are decorrelated in the projected space [2]. In multi-class problems, the variations of data are determined on a global basis, that is, the principal axes are derived from a global covariance matrix:

$$\hat{S} = \frac{1}{N} \sum_{j=1}^{K} \sum_{i=1}^{N_j} (x_{ji} - \hat{\mu})(x_{ji} - \hat{\mu})^{\rm T}, \tag{8}$$

where $\hat{\mu}$ is the global mean of all the samples, K is the number of classes, N_j is the number of samples in class j, $N = \sum_{j=1}^{K} N_j$ and x_{ji} represents the ith observation from class j. The principal axes $T_1, T_2, \ldots, T_m$ are therefore the m leading eigenvectors of $\hat{S}$:

$$\hat{S} T_i = \hat{\lambda}_i T_i, \quad i \in 1, \ldots, m, \tag{9}$$

where $\hat{\lambda}_i$ is the ith largest eigenvalue of $\hat{S}$. An assumption made for feature extraction and dimensionality reduction by PCA is that most information of the observation vectors is contained in the subspace spanned by the first m principal axes, where $m < p$. Therefore, each original data vector can be represented by its principal component vector with dimensionality m.

3. Minimum classification error training algorithm

3.1. Derivation of MCE criterion

Consider an input vector x, the classifier makes its decision by the following decision rule:

$$x \in \text{Class } k \quad \text{if } g_k(x, \Lambda) = \max_{\text{for all } i \in K} g_i(x, \Lambda), \tag{10}$$

where $g_i(x, \Lambda)$ is discriminant function of x to class i, Λ is the parameter set and K is the number of classes. The negative of $g_k(x, \Lambda) - \max_{\text{for all } i \neq k} g_i(x, \Lambda)$ can be used as a measure of misclassification of x. This form, however, is not differentiable and needs further modification. In [7], a modified version is introduced as a misclassification measure. For the kth class, it is given by

$$d_k(x, \Lambda) = -g_k(x, \Lambda) + \left[\frac{1}{N-1} \sum_{\text{for all } i \neq k} (g_i(x, \Lambda))^{\eta} \right]^{1/\eta}, \tag{11}$$

where η is a positive number and $g_k(x, \Lambda)$ is the discriminant of observation x to its known class k. When η approaches ∞, it reduces to

$$d_k(x, \Lambda) = -g_k(x, \Lambda) + g_j(x, \Lambda), \tag{12}$$

where class j has the largest discriminant value among all the classes other than class k. Obviously, $d_k(x, \Lambda) > 0$

implies misclassification, $d_k(x,\Lambda) < 0$ means correct classification and $d_k(x,\Lambda)=0$ suggests that x sits on the boundary. The loss function is then defined as a monotonic function of misclassification measure. The sigmoid function is often chosen since it is a smoothed zero-one function suitable for gradient descent algorithm. The loss function is thus given as

$$l_k(x,\Lambda) = f(d_k(x,\Lambda)) = \frac{1}{1+\mathrm{e}^{-\xi d_k(x,\Lambda)}}, \tag{13}$$

where $\xi > 0$. For a training set $\mathscr{X}$, the empirical loss is defined as

$$L(\Lambda) = E\{l_k(x,\Lambda)\} = \sum_{k=1}^{K}\sum_{i=1}^{N_k} l_k(x^{(i)},\Lambda), \tag{14}$$

where N_k is the number of samples in class k. Clearly, minimizing the above empirical loss function will lead to the minimization of the classification error. As a result, Eq. (14) is called the MCE criterion [7,8]. The class parameter set Λ is therefore obtained by minimizing the loss function through the steepest gradient descent algorithm. This is an iterative algorithm and the iteration rules are:

$$\Lambda_{t+1} = \Lambda_t - \varepsilon \nabla L(\Lambda)|_{\Lambda=\Lambda_t},$$

$$\nabla L(\Lambda) = \begin{bmatrix} \partial L/\partial\lambda_1 \\ \vdots \\ \partial L/\partial\lambda_d \end{bmatrix}, \tag{15}$$

where t denotes tth iteration, $\lambda_1,\ldots,\lambda_d \in \Lambda$ are class parameters, $\varepsilon > 0$ is the adaption constant. For $s=1,2,\ldots,d$, the gradient $\nabla L(\Lambda)$ can be computed as follows:

$$\frac{\partial L}{\partial\lambda_s} = \xi\sum_{i=1}^{N_k} L^{(i)}(1-L^{(i)})\frac{\partial g_k(x^{(i)},\Lambda)}{\partial\lambda_s} \quad \text{if } \lambda_s \in \text{ class } k,$$

$$\frac{\partial L}{\partial\lambda_s} = -\xi\sum_{i=1}^{N_j} L^{(i)}(1-L^{(i)})\frac{\partial g_j(x^{(i)},\Lambda)}{\partial\lambda_s} \text{if } \lambda_s \in \text{ class } j. \tag{16}$$

In the case of Mahalanobis distance measure-based discriminant functions, $\Lambda = \{\mu,\Sigma\}$, where μ is class mean and Σ is covariance matrix. The differentiation of discriminant functions with respect to Λ is

$$\frac{\partial g_m(x^{(i)},\Lambda)}{\partial\mu} = -(x-\mu)^{\mathrm{T}}\Sigma^{-1} - \Sigma^{-1}(x-\mu),$$

$$m = 1,\ldots,K,$$

$$\frac{\partial g_m(x^{(i)},\Lambda)}{\partial\Sigma} = -(x-\mu)^{\mathrm{T}}(\Sigma^{-1})^2(x-\mu),$$

$$m = 1,\ldots,K. \tag{17}$$

An alternative definition of misclassification measure can be used to enhance the control of the joint behaviour of discriminant functions $g_k(x,\Lambda)$ and $g_j(x,\Lambda)$. The alternative misclassification is defined as follows:

$$d_k(x,\Lambda) = \frac{[1/(N-1)\sum_{\text{for all } i\neq k} g_i(x,\Lambda)^{\eta}]^{1/\eta}}{g_k(x,\Lambda)}. \tag{18}$$

To the extreme case, i.e. $\eta \to \infty$, Eq. (18) becomes

$$d_k(x,\Lambda) = \frac{g_j(x,\Lambda)}{g_k(x,\Lambda)}. \tag{19}$$

The class parameters and transformation matrix are optimized using the same adaption rules as shown in Eq. (15). The gradients with respect to Λ are computed as

$$\frac{\partial L}{\partial\lambda_s} = -\xi\sum_{i=1}^{N_j} L^{(i)}(1-L^{(i)})\frac{g_j(x^{(i)},\Lambda)}{[g_k(x^{(i)},\Lambda)]^2}\frac{\partial g_k(x^{(i)},\Lambda)}{\partial\lambda_s}$$

$$\text{if } \lambda_s \in \text{ class } k,$$

$$\frac{\partial L}{\partial\lambda_s} = \xi\sum_{i=1}^{N_k} L^{(i)}(1-L^{(i)})\frac{1}{g_k(x^{(i)},\Lambda)}\frac{\partial g_j(x^{(i)},\Lambda)}{\partial\lambda_s}$$

$$\text{if } \lambda_i \in \text{ class } j, \tag{20}$$

where $\lambda_s \in \Lambda, s=1,\ldots,d$. The differentiation of discriminant functions can be computed by Eq. (17).

3.2. Using MCE training algorithms for dimensionality reduction

As with other feature extraction methods, MCE reduces feature dimensionality by projecting the input vector into a lower dimensional feature space through a linear transformation $T_{m\times p}$, where $m < p$. Let the class parameter set in the feature space be $\tilde{\Lambda}$. Accordingly, the loss function becomes

$$l(x,\tilde{\Lambda},T) = l(d_k(Tx,\tilde{\Lambda})) = \frac{1}{1+\mathrm{e}^{-\alpha d(Tx,\tilde{\Lambda})}}. \tag{21}$$

The empirical loss over the whole data set is given by

$$L(\tilde{\Lambda},T) = E\{l(d_k(Tx,\tilde{\Lambda}))\} = \sum_{k=1}^{K}\sum_{i=1}^{N_k} l_k(x^{(i)},\tilde{\Lambda},T). \tag{22}$$

Since Eq. (22) is a function of T, the elements in T can be optimized together with the parameter set $\tilde{\Lambda}$ in the same gradient descent procedure. The adaption rule for T is

$$T_{sq}(t+1) = T_{sq}(t) - \varepsilon\left.\frac{\partial L}{\partial T_{sq}}\right|_{T_{sq}=T_{sq}(t)} \tag{23}$$

where t denotes tth iteration, ε is the adaption constant or learning rate and s and q are the row and column indicators of transformation matrix T. The gradient with respect to T can be computed by

Conventional MCE:

$$\frac{\partial L}{\partial T_{sq}} = \xi\sum_{k=1}^{K}\sum_{i=1}^{N_k} L^{(i)}(1-L^{(i)})$$

$$\times\left(\frac{\partial g_k(Tx^{(i)},\tilde{\Lambda})}{\partial T_{sq}} - \frac{\partial g_j(Tx^{(i)},\tilde{\Lambda})}{\partial T_{sq}}\right).$$

Alternative MCE:

$$\frac{\partial L}{\partial T_{sq}} = \xi \sum_{k=1}^{K} \sum_{i=1}^{N_k} L^{(i)}(1 - L^{(i)}) \times \frac{(\partial g_j(Tx^{(i)}, \tilde{\Lambda})/\partial T_{sq}) g_k(Tx^{(i)}, \tilde{\Lambda}) - (\partial g_k(Tx^{(i)}, \tilde{\Lambda})/\partial T_{sq}) g_j(Tx^{(i)}, \tilde{\Lambda})}{[g_k(Tx^{(i)}, \tilde{\Lambda})]^2}, \tag{24}$$

where in Mahalanobis distance based discriminant functions:

$$\frac{\partial g_m(Tx^{(i)}, \tilde{\Lambda})}{\partial T} = (Tx - \tilde{\mu})^{\mathrm{T}} \tilde{\Sigma}^{-1} x + x^{\mathrm{T}} \tilde{\Sigma}^{-1} (Tx - \tilde{\mu}), \quad m = 1, \ldots, K. \tag{25}$$

3.3. Generalized MCE training algorithm

One of the major concerns about MCE training for dimensionality reduction is the initialization of the parameters. This is because the gradient descent method used in MCE training algorithm does not guarantee the global minimum value. The optimality of MCE training process is largely dependent on the initialization of T and class parameter sets Λ.

Among these parameters, transformation matrix T is crucial to the success of MCE training since it filters the class information to be brought into the decision space. Paliwal et al. [9] give an initialization of the MCE training algorithm, in which T is taken to be a unity matrix. However, in many cases, this is a convenient way of initialization rather than an effective way because the classification criterion has not been considered in the initialization. In order to increase the generalization of MCE training algorithm, it is necessary to embed the classification criteria into the initialization process. From the searching point of view, we can regard MCE training as two sequential search procedures: one is general but rough search for the initialization of parameters and the other, local but thorough search for the optimization of parameters. The former search procedure will provide a global optimized initialization of class parameters and the latter will make a thorough search to find the relevant local minimum. Fig. 3 compares the normal MCE training process to the generalized MCE training process. So far, no criterion on general searching process has been proposed. However we can employ current feature extraction methods to this process. In our practice, we employ LDA and PCA for the general searching process for the initialization of class parameters.

4. Support vector machine

4.1. Constructing SVM

Considering a two-class case, suppose the two classes are ω_1 and ω_2 and we have a set of training data $\mathscr{X} = \{x_1, \ldots, x_N\} \subset \mathscr{R}^p$. The training data are labeled by the following rule:

$$y_i = \begin{cases} +1, & x_i \in \omega_1, \\ -1, & x_i \in \omega_2. \end{cases} \tag{26}$$

The basic idea of SVM estimation is to project the input observation vectors non-linearly into a high dimensional feature space $\mathscr{F}$ and then compute a linear function in $\mathscr{F}$. The functions take the form

$$f(x) = (w \cdot \Phi(x)) + b \tag{27}$$

with $\Phi : \mathscr{R}^p \to \mathscr{F}$ *and* $w \in \mathscr{F}$, where $(\cdot)$ denotes the dot product. Ideally, all the data in these two classes satisfy the following constraint:

$$y_i(w \cdot \Phi(x_i)) + b - 1 \geqslant 0 \quad \forall i. \tag{28}$$

Considering the points $\Phi(x_i)$ in $\mathscr{F}$ for which the equality in Eq. (28) holds, these points lie on two hyper-planes $H_1 : (w \cdot \Phi(x_i)) + b = +1$ and $H_2 : (w \cdot \Phi(x_i)) + b = -1$. These two hyper-planes are parallel and no training points fall between them. The margin between them is $2/\|w\|$. Therefore we can find a pair of hyper-planes with maximum margin by minimizing $\|w\|^2$ subject to Eq. (28) [24]. This problem can be written as a convex optimization problem:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|w\|^2 \\ \text{subject to} & y_i(w \cdot \Phi(x_i)) + b - 1 \geqslant 0 \quad \forall i, \end{array} \tag{29}$$

where the first function is *primal* objective function and the second function is the corresponding constraints. Eq. (29) can be solved by constructing a Lagrange function from both the primal function and the corresponding constraints. Hence we introduce positive Lagrange multipliers α_i, $i = 1, \ldots, N$, one for each constraint in Eq. (29). The Lagrange function is given by

$$L_P = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i y_i (w \cdot \Phi(x_i) + b) + \sum_{i=1}^{N} \alpha_i, \tag{30}$$

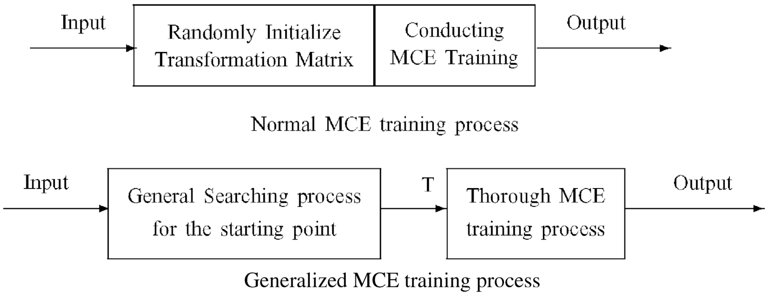
L_P must be minimized with respect to w and b, which requires the gradient of L_P to vanish with respect to w and b. The gradients are given by

$$\frac{\partial L_P}{\partial w_s} = w_s - \sum_{i=1}^{N} \alpha_i y_i \Phi(x_{is}) = 0, \qquad s = 1, \ldots, p,$$

$$\frac{\partial L_P}{\partial b} = -\sum_{i=1}^{N} \alpha_i y_i = 0, \tag{31}$$

where p is the dimension of space $\mathscr{F}$. Combine these conditions and other constraints on primal functions and Lagrange multipliers, we obtain the *Karush–Kuhn–Tucker* (KKT) conditions:

$$\frac{\partial L_P}{\partial w_s} = w_s - \sum_{i=1}^{N} \alpha_i y_i \Phi(x_{is}) = 0, \qquad s = 1, \ldots, p,$$


Input
Randomly Initialize Transformation Matrix
Conducting MCE Training
Output
Input
General Searching process for the starting point
T
Thorough MCE training process
Output

Normal MCE training process

Generalized MCE training process

Fig. 3. A comparison between the normal MCE training process and the generalized MCE training process.

$$\frac{\partial L_P}{\partial b} = -\sum_{i=1}^{N} \alpha_i y_i = 0,$$

$$y_i(w \cdot \Phi(x_i)) + b - 1 \geqslant 0 \quad \forall i,$$

$$\alpha_i \geqslant 0 \quad \forall i,$$

$$\alpha_i(y_i(w \cdot \Phi(x_i)) + b - 1) = 0 \quad \forall i, \tag{32}$$

where w, b and α are the variables to be solved. From KKT condition Eq. (31) we obtain

$$w = \sum_{i=1}^{N} \alpha_i y_i \Phi(x_i),$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0. \tag{33}$$

Therefore,

$$\begin{aligned} f(x) &= \sum_{i=1}^{N} \alpha_i y_i (\Phi(x_i) \cdot \Phi(x)) + b \\ &= \sum_{i=1}^{N} \alpha_i y_i k(x_i, x) + b, \end{aligned} \tag{34}$$

where $k(x_i, x) = (\Phi(x_i) \cdot \Phi(x_j))$ is a kernel function that uses dot product in feature space. Substitute Eq. (33) into Eq. (30). This leads to maximization of the dual function L_D:

$$L_D = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k_{ij} + \sum_{i=1}^{N} \alpha_i. \tag{35}$$

Writing the dual function incorporating the constraints, we obtain the dual optimization problem:

$$\begin{aligned} \text{maximize} \quad & -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k_{ij} + \sum_{i=1}^{N} \alpha_i \\ \text{subject to} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0, \\ & \alpha_i \geqslant 0 \quad \forall i. \end{aligned} \tag{36}$$

Both the primal problem L_P (Eq. (30)) and the dual problem L_D (Eq. (35)) are constructed from the same objective function but with different constraints. Optimization of this primal–dual problem is a type of convex optimization problem and can be solved by the interior point algorithm [21]. However, the discussion on interior point algorithm is beyond the scope of this paper. A detailed discussion on this algorithm is given by Vanderbei in [25].

4.2. Multi-class SVM classifiers

SVM is a two-class based pattern classification algorithm. Therefore a multi-class based SVM classifier has to be constructed. So far the best method of constructing multi-class SVM classifier is not clear [26]. Scholkopf et al. [27] proposed a "one vs. all" type classifier. Clarkson and Moreno [26] proposed a "one vs. one" type classifier. Their structures are shown in Fig. 4.

Both types of classifiers are in fact combinations of two-class based SVM sub-classifiers. When an input data vector x enters the classifier, a K-dimensional value vector $f^{(i)}(x)$, $i = 1, \ldots, K$ (one dimension for each class) is generated. The classifier then classifies x by the following classification criteria:

$$x \in \text{ Class } i \quad \text{if } f^{(i)}(x) = \max_{\text{for all } j \in K} f^{(j)}(x). \tag{37}$$

5. Classification experiments

Our experiments focus on vowel recognition tasks. Two databases are used. We start with Deterding vowels database [28]. The advantage of starting with it is that the computational burden is small. Deterding database is used to evaluate different types of GMCE training algorithms and SVM classifiers. Then, feature extraction and classification algorithms are tested with TIMIT database [29]. The feature extraction and classification algorithms involved in the experiments are listed in Table 1.

In order to evaluate the performance of the linear feature extraction algorithms (PCA, LDA, MCE and GMCE), we have used a minimum distance classifier. Here, a feature vector y is classified to jth class if the distance $d_j(y)$ is

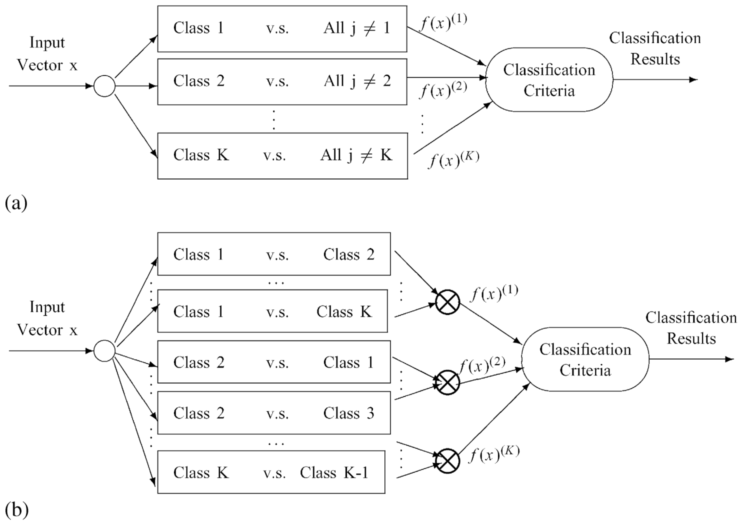


Fig. 4. Two types of multi-class SVM classifier: (a) Structure of "one vs. all" multi-class SVM Classifier; (b) Structure of "one vs. one" multi-class SVM classifier.

Table 1
Feature extraction and classification algorithms used in the experiments

Parameter used	Dimension	Feature extractor	Classifier
LAR(Deterding)	10	PCA	Minimum distance (Mahalanobis)
″	″	LDA	″
″	″	MCE	″
″	″	GMCE	″
MFCC(TIMIT)	21	PCA	″
″	″	LDA	″
″	″	MCE	″
″	″	GMCE	″
LAR(Deterding)	10	NONE	SVM one vs. one
LAR(Deterding)	10	NONE	SVM one vs. all
MFCC(TIMIT)	21	NONE	SVM one vs. one

less than the other distances $d_i(y)$, $i = 1, \ldots, K$. We use Mahalanobis distance measure to compute the distance of a feature vector from a given class. Thus, the distance $d_i(y)$ is computed as follows:

$$d_i(y) = (y - \mu_i)^{\mathrm{T}} \Sigma_i^{-1} (y - \mu_i), \tag{38}$$

where μ_i is the mean vector of class i and Σ_i is the covariance matrix. In our experiments, we use full covariance matrix.

Three types of SVM kernel function are evaluated on Deterding database. The formulation of kernel functions is as follows:

Linear kernel:

$$k(x, y) = x \cdot y,$$

Polynomial kernel:

$$k(x, y) = (x \cdot y + 1)^p,$$

RBF kernel:

$$k(x, y) = \mathrm{e}^{|x-y|^2/2\delta^2}. \tag{39}$$

5.1. Deterding database experiments

Deterding vowels database has 11 vowel classes as shown in Table 2. This database has been used in the past by a number of researchers for pattern recognition applications [26,28,30,31]. Each of these 11 vowels is uttered 6 times by 15 different speakers. This gives a total of 990 vowel tokens. A central frame of speech signal is excised from each of these vowel tokens. A 10th order linear prediction analysis is performed on each frame and the resulting Linear Prediction Coefficients (LPCs) are converted to 10 log-area (LAR) parameters. 528 frames from eight speakers are used to train the models and 462 frames from the rest seven speakers are used to test the models.

Table 3 compares the results LDA, PCA, the conventional form and the alternative form of MCE training algorithm. The results show that the alternative MCE training algorithm has the best performance. Thus we use the alternative MCE in the following experiments.

Two types of GMCE training algorithm are investigated in Deterding database experiments. One uses LDA for general search and the other uses PCA. Figs. 5 and 6 show the experiment results. Since the alternative MCE training algorithm is chosen for MCE training, we denote these two types of GMCE training algorithms as GMCE+LDA and GMCE+PCA, respectively. The normal alternative MCE

Table 2
Vowels and words used in Deterding database

Vowel	Word	Vowel	Word	Vowel	Word	Vowel	Word
i	heed	O	hod	I	hid	C:	hoard
E	head	U	hood	A	had	u:	who’d
a:	hard	3:	heard	Y	hud		

Table 3
Comparison of various feature extractors

DATABASE	Conventional MCE	Alternative MCE	LDA	PCA
VOWELS(Train)	85.6%	99.1%	97.7%	97.7%
VOWELS(Test)	53.7%	55.8%	51.3%	49.1%

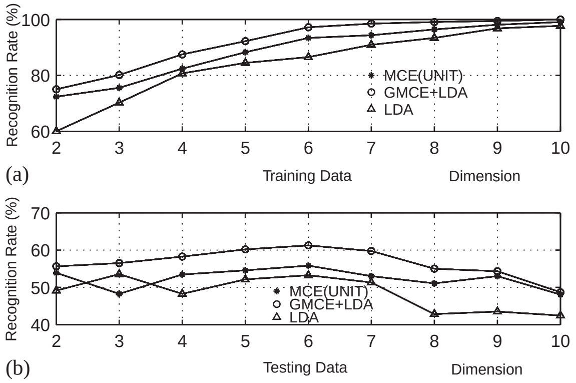


Fig. 5. Results of MCE(UNIT), GMCE+LDA, LDA on Deterding database.

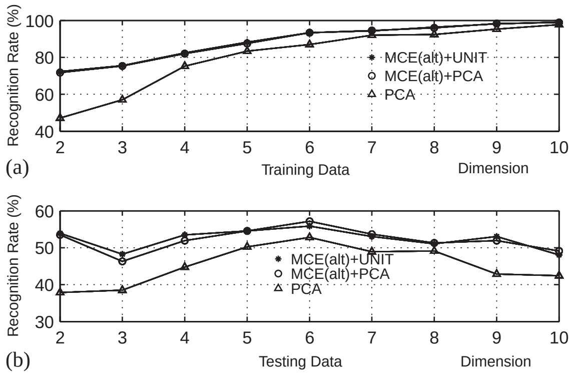


Fig. 6. Results of MCE(UNIT), GMCE+PCA, PCA on Deterding database.

Table 4
Deterding vowel data-set classification results

Kernel	Classifier	Training (%)	Testing (%)
Linear	One vs. all	49.43	40.91
Linear	One vs. one	79.73	53.03
Polynomial	One vs. all	59.85	42.42
Polynomial	One vs. one	90.53	55.63
RBF	One vs. all	78.98	51.95
RBF	One vs. one	90.34	58.01

training algorithm is denoted as MCE(UNIT). Observations from these results can be summarized as follows:

- GMCE training algorithm has an improved performance when LDA is used for the general search for the initial transformation matrix and GMCE+LDA demonstrates the best performance among MCE(UNIT), GMCE+PCA, LDA and PCA.
- The performance of GMCE training algorithm is not improved when PCA is employed for the general searching process.
- Performances of GMCE+LDA and GMCE+PCA on testing data show that the best classification results are usually obtained when the dimensionality is reduced to 50–70%.

Table 4 shows the classification results of different SVM classifiers. The order of polynomial kernel function is 3. The classification result show that the performance of RBF kernel function is the best among the three types of kernels. The overall performance of “one vs. one” multi-class classifier is much better than “one vs. all” multi-class classifier. Among all the six types of SVM classifiers, the “one vs. one” multi-class classifier with RBF kernel function has

Table 5
Number of selected phonemes in training data-set

Phonemes	aa	ae	ah	ao	aw	ax	ay	eh	oy	uh
Training	541	665	313	445	126	207	395	591	118	57
Testing	176	214	136	168	40	89	131	225	49	21
Phonemes	el	en	er	ey	ih	ix	iy	ow	uw	Total
Training	145	97	384	346	697	583	1089	336	106	7241
Testing	42	34	135	116	239	201	381	116	37	2550

the best overall performance and is thus selected for further experiments.

5.2. TIMIT database experiments

In order to provide results on bigger database, we use the TIMIT database for vowel recognition. This database contains a total of 6300 sentences, 10 sentences spoken by each of the 630 speakers. Training part of this database is used for training the vowel recognizer's and the test part for testing. The vowels used in the classification tasks are selected from the vowels, semi-vowels and nasals given in TIMIT database. Altogether 17 vowels, 1 semi-vowel and 1 nasal are selected for the vowel classification experiments. The TIMIT database comes with phonemic transcription and associated acoustic phonemic boundaries. The center 20 msec segments of selected vowels are excised from each sentence. Spectral analysis is performed on these segments and each segment is represented by a 21-dimension Mel-Frequency Cepstral Coefficients (MFCCs) feature vectors. Each vector contains 1 energy coefficient and 20 MFCCs. Mahalanobis distance based minimum distance classifier is used as pattern classifier. Table 5 shows the number of segments of each vowel used in the experiment.

5.2.1. Comparison of separate and integrated pattern recognition systems

Fig. 7 shows the results of separate pattern recognition systems, PCA and LDA plus classifier and integrated systems, MCE and SVM in feature extraction and classification tasks. The dimensionalities used in the experiments are from 3 to 21—full dimension. The horizontal axis of the figure is the dimension axis. The vertical axis represents the recognition rates. Since SVM is not suitable for dimensionality reduction, it is applied to classification tasks only and the results of SVM appears in the figure as single points. Observations from Fig. 7 can be summarized as follows:

- LDA has a fairly flat performance curve. It performs the best in the low-dimensional feature spaces (Dimension 3–12) among LDA, PCA and MCE training algorithm on training data. On testing data, LDA performs better than PCA and MCE in low-dimensional spaces (dimension 3–15) too.
- MCE training algorithm performs better than LDA and PCA in the high-dimensional feature spaces (Dimension 13–21) on training data. On the testing data, MCE training algorithm performs better than PCA and MCE in the high-dimensional spaces from dimension 16–21.
- The performances of SVM on training data are not as good as those of LDA, PCA and MCE training algorithm. However, SVM performs much better than LDA, PCA and MCE training algorithm on testing data.

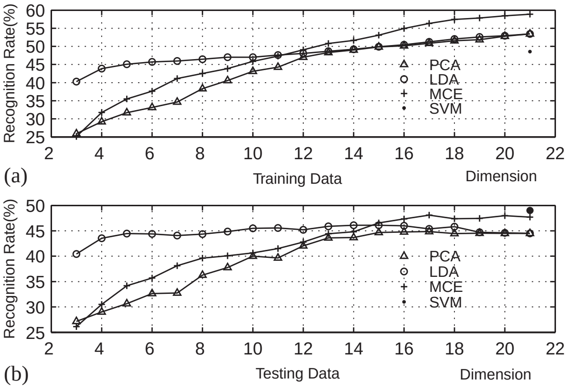


Fig. 7. Results of LDA, PCA, MCE and SVM on TIMIT database.

5.2.2. Analysis of GMCE training algorithm

In this section, we investigate the performance of GMCE. Two types of GMCE are used. One employs LDA for general search, which we denote as GMCE+LDA. The other employs PCA for general search and we denote as GMCE+PCA. Experiments results are shown in Figs. 8 and 9. Observations from the two figures can be summarized as follows:

- When GMCE uses LDA as the general search tool, the performances of GMCE are better than both LDA and MCE in all dimensions. When GMCE uses PCA in the general search process, the general performances of GMCE are not significantly improved.
- In high-dimensional feature spaces (Dimension 15–21), the performances of GMCE+LDA are close to those of MCE training algorithm, which are better than LDA.
- In medium-dimensional (Dimension 7–15) and low-dimensional (Dimension 3–7) feature spaces, GMCE

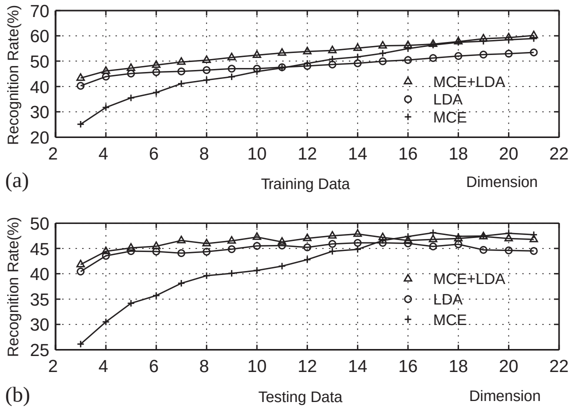


Fig. 8. Results of GMCE+LDA, MCE and LDA on TIMIT database.

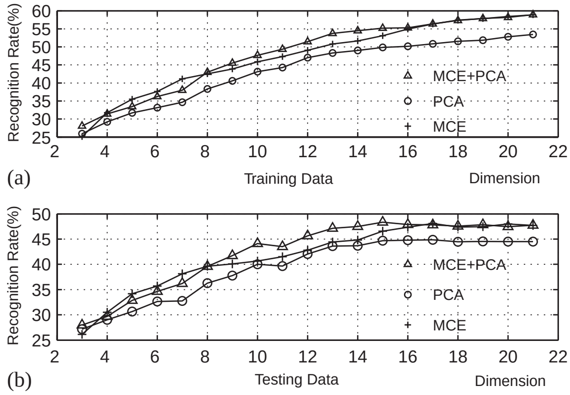


Fig. 9. Results of GMCE+PCA, MCE and PCA on TIMIT database.

+LDA has significantly better performances than both LDA and MCE.

6. Conclusions

In this paper, we investigate major feature extraction and classification algorithms. Five algorithms are involved in the investigation. They are LDA, PCA, MCE training algorithm, proposed GMCE training algorithm and SVM. From the observation of the experimental results, the following conclusions can be drawn:

- Pattern classification systems that integrate feature extraction and classification, such as MCE and GMCE training algorithms, show better performance than systems with independent feature extraction algorithm, such as LDA, PCA.
- GMCE training algorithm has an improved performance over MCE training algorithm. The significant improvements are in low-dimensional feature spaces.
- The performance of SVM is interesting. It is poorer on training data than that of LDA, PCA and MCE and GMCE training algorithms, while on testing data, it is better than the other four linear feature extraction algorithms. This implies that linearly extracted feature models have better fitness to training data, while SVM has better generalization properties.

7. Summary

Conventional pattern recognition systems have two components: feature analysis and pattern classification. Feature analysis is achieved in two steps: parameter extraction step and feature extraction step. LDA and PCA are the two popular independent feature extraction methods. The drawback of independent feature extraction algorithms is that their optimization criteria are different from the classifier's minimum classification error criterion, which may cause inconsistency between feature extraction and the classification stages of a pattern recognizer. A direct way to overcome this problem is to conduct feature extraction and classification jointly with a consistent criterion. MCE training algorithm provides such an integrated framework. LDA, PCA and MCE training algorithm are linear feature extraction algorithms. The advantage of linear algorithms is their ability to reduce feature dimensionalities. However, they have the limitation that the decision boundaries generated are linear. SVM is a recently developed pattern classification algorithm with non-linear formulation. It is based on the idea that the classification that affords dot-products can be computed efficiently in higher dimensional feature spaces. SVM has the advantage that it can handle the classes with complex non-linear decision boundaries. This paper investigates LDA, PCA and MCE algorithms for feature extraction. A generalized MCE (GMCE) training algorithm is proposed to mend the shortcomings of the MCE training algorithms. SVM is investigated in this paper as an integrated pattern classification system.

References

[1] M. Hotelling, Analysis of a complex of statistical variables into principal components, J. Educational Psychol. 24 (1933) 498–520.

[2] I.T. Jolliffe, Principal Component Analysis, Springer-Verlag, New York, 1986.

[3] C.R. Rao, The use and interpretation of principal component analysis in applied research, Sankhya A 26 (1964) 329–358.

[4] E.L. Bocchieri, J.G. Wilpon, Discriminative analysis for feature reduction in automatic speech recognition, Proceedings of IEEE International Conference of Acoustics, Speech and Signal Processing, Vol. 1, 1992, pp. 501–504.

[5] W.L. Poston, D.J. Marchette, Recursive dimensionality reduction using Fisher's linear discriminant, Pattern Recognition 31 (7) (1998) 881–888.

[6] D.X. Sun, Feature dimension reduction using reduced-rank maximum likelihood estimation for hidden Markov model,

Proceedings of International Conference on Spoken Language Processing, Philadelphia, USA, 1996, pp. 244–247.

[7] B.H. Juang, S. Katagiri, Discriminative learning for minimum error classification, IEEE Trans. Signal Process. 40 (12) (1992) 3043–3054.

[8] S. Katagiri, C.H. Lee, B.H. Juang, A generalized probabilistic descent method, Proceedings of the Acoustic Society of Japan, Fall Meeting, 1990, pp. 141–142.

[9] K.K. Paliwal, M. Bacchiani, Y. Sagisaka, Simultaneous design of feature extractor and pattern classifier using the minimum classification error training algorithm, Proceedings of IEEE Workshop on Neural Networks for Signal Processing, Boston, USA, September, 1995, pp. 67–76.

[10] P.C. Chang, S.H. Chen, B.H. Juang, Discriminative analysis of distortion sequences in speech recognition, Proceedings of IEEE International Conference of Acoustics, Speech and Signal Processing, Vol. 1, 1991, pp. 549–552.

[11] I. Komori, S. Katagiri, GPD training of dynamic programming-based speech recognizer, J. Acoust. Soc. Japan E, 13 (6) (1992) 341–349.

[12] W. Chou, B.H. Juang, C.H. Lee, Segmental GPD training of HMM based speech recognizer, Proceedings of IEEE International Conference of Acoustics, Speech and Signal Processing, Vol. 1, 1992, pp. 473–476.

[13] C.S. Liu, C.H. Lee, W. Chou, B.H. Juang, A study on minimum error discriminative training for speaker recognition, J. Acoust. Soc. Amer. 97 (1) (1995) 637–648.

[14] D. Rainton, S. Sagayama, Minimum error classification training of HMMs-implementation details and experimental results, J. Acoust. Soc. Japan E, 13 (6) (1992) 379–387.

[15] B.E. Boser, I.M. Guyon, V. Vapnik, A training algorithm for optimal margin classifiers, in: D. Haussler (Ed.), 5th Annual ACM Workshop on COLT, Pittsburgh, PA, 1992, pp. 144–152.

[16] V. Roth, V. Steinhage, Nonlinear discriminant analysis using kernel functions, Technical Report, Nr IAI-TR-99-7, ISSN 0944-8535, University Bonn, 1999.

[17] V. Vapnik, The Nature of Statistical Learning Theory, Springer, NY, 1995.

[18] T. Joachims, Making large-scale SVM learning practical, in: B. Scholkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge, USA, 1998, pp. 169–184.

[19] B. Scholkopf, C. Gurges, V. Vapnik, Incorporating invariances in support vector learning machines, International Conference on Artificial Neural Networks—ICANN'96, Berlin, 1996, pp. 47–52.

[20] B. Scholkopf, P. Bartlett, A. Smola, R. Williamson, Support vector regression with automatic accuracy control, Proceedings of 8th International Conference on Artificial Neural Networks, Perspectives in Neural Computing, Berlin, 1998, pp. 111–116.

[21] A.J. Smola, B. Scholkopf, A tutorial on support vector regression, NeuroCOLT2 Technical Report Series NC2-TR-1998-030, ESPRIT working group on Neural and Computational Learning Theory "NeuroCOLT 2", 1998.

[22] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, John Wiley, New York, 1973.

[23] G.W. Cottrell, Principal components analysis of images via back propagation, SPIE Proceedings in Visual Communication and Image Processing, Vol. 1001, 1988, pp. 1070–1077.

[24] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery. 2 (2) (1998) 955–974.

[25] R.J. Vanderbei, LOQO: An interior point code for quadratic programming, Optimization Methods and Software. 11 (1999) 451–484.

[26] P. Clarkson, P.J. Moreno, On the use of support vector machines for phonetic classification, in: Proceedings of International Conference on Acoustics, Speech, Signal Processing'99, 1999, pp. 585–588.

[27] B. Scholkopf, C. Gurges, V. Vapnik, Extracting support data for a given task, Proceedings of First International Conference on Knowledge Discovery and Data Mining, Menlo Park, 1995, pp. 252–257.

[28] T. Robinson, Dynamic error propagation networks, Ph.D. Thesis, Cambridge University Engineering Department, February 1989.

[29] W. Fisher, G. Doddington, K. Goudie-Mashall, The DARPA speech recognition research database: specifications and status, Proceedings DARPA Speech Recognition Workshop, 1986, pp. 93–99.

[30] A. Sankar, R. Mammone, Neural tree networks, in: R. Mammone, Y. Zeevi (Eds.), Neural Networks: Theory and Applications, Academic Press, New York, 1991, pp. 281–302.

[31] A.C. Tsoi, T. Pearson, Comparison of the three classification techniques, CART, C4.5 and multi-layer perceptrons, in: R.L. Lippman, J.E. Moody, D.S. Touretzky (Eds.), Advances in Neural Information Processing Systems, Vol. 3, Morgan Kaufmann, San Mateo, CA, 1990, pp. 963–969.

About the Author—WANG, XUECHUAN received his B.E. degree in Mechanical Engineering from Hunan University, Changsha, China in 1992 and M.E. degree in Signal Processing from Huazhong University of Science and Technology, Wuhan, China in 1995. He has been a Ph.D. candidate since 1998 in Signal Processing Lab at Griffith University, Brisbane, Australia. His current research interests include discriminant learning and speech recognition.

About the Author—KULDIP K. PALIWAL received the B.S. degree from Agra University, India in 1969, the M.S. degree from Aligarh University, India, in 1971 and the Ph.D. degree from Bombay University, India, in 1978. Since 1993, he has been a Professor at the Griffith University, Brisbane, Australia. He has worked at a number of organizations including Tata Institute of Fundamental Research, Bombay, India, Norwegian Institute of Technology, Trondheim, Norway, University of Keele, U.K., AT& T Bell Laboratories, Murray Hill, New Jersey, U.S.A. and Advanced Telecommunication Research Laboratories, Kyoto, Japan. He has co-edited two books: *Speech Coding and Synthesis* (Elsevier, 1995) and *Speech and Speaker Recognition*: *Advanced Topics* (Kluwer, 1996). He has published more than 100 papers in international journals. He is a recipient of the 1995 IEEE Signal Processing Society Senior Award. He has been an Associate Editor of the IEEE Transactions on Speech and Audio Processing, and IEEE Signal Processing Letters. His current research interests include speech processing, image coding and neural networks.