American Journal of Applied Sciences 2 (10): 1445-1455, 2005
ISSN 1546-9239

Pattern Classification: An Improvement Using Combination of VQ and PCA Based Techniques

[1]Alok Sharma, [1]Kuldip K. Paliwal and [2]Godfrey C. Onwubolu
[1]Signal Processing Lab., Griffith University, Brisbane, Australia
[2]Department of Engineering, University of the South Pacific, Suva, Fiji

Abstract: This study firstly presents a survey on basic classifiers namely Minimum Distance Classifier (MDC), Vector Quantization (VQ), Principal Component Analysis (PCA), Nearest Neighbor (NN) and K-Nearest Neighbor (KNN). Then Vector Quantized Principal Component Analysis (VQPCA) which is generally used for representation purposes is considered for performing classification tasks. Some classifiers achieve high classification accuracy but their data storage requirement and processing time are severely expensive. On the other hand some methods for which storage and processing time are economical do not provide sufficient levels of classification accuracy. In both the cases the performance is poor. By considering the limitations involved in the classifiers we have developed Linear Combined Distance (LCD) classifier which is the combination of VQ and VQPCA techniques. The proposed technique is effective and outperforms all the other techniques in terms of getting high classification accuracy at very low data storage requirement and processing time. This would allow an object to be accurately classified as quickly as possible using very low data storage capacity.

Key words: VQPCA, Classification accuracy, LCD, total parameter requirement, processing time

INTRODUCTION

Pattern classification/recognition is an area where we learn how to better familiarize the objects to the machine and get actions or decisions based on the observed categories of the pattern. A pattern could be human face, sampled speech, handwritten or printed digits, any letter, gesture, spoken word, financial data, biometric data or any statistical data. Humans naturally classify/recognize patterns from the environment in everyday life. A five year old kid can adapt to different type of objects or patterns and react accordingly. This adaptation is taken for granted until we come to teach a machine to classify/recognize and provide actions or decisions on the same patterns.

The more the patterns available, the better the decision would be. This gives hope to design a classifier system. For the last five decades research is going on in this field to provide an optimum classifier/recognizer. But the classifier performance is still far behind the perception of a human brain. However, pattern classification/recognition plays a crucial role in the areas like banking, multimedia communication, data synthesis, speech or image processing, forensic sciences, computer vision and remote sensing, data mining, robotics and artificial intelligence. It emerged as an essential and integral part of daily life. The evolving computational demand in pattern classification makes this field very challenging and thus open for research. For example in image recognition, several thousands of multidimensional patterns are required for processing which makes the implementation of the classifier system quite impossible.

There are two main categories of pattern classification (i) supervised classification: where the state of nature for each pattern is known and (ii) unsupervised classification: where the state of nature is unknown and learning is based on the similarity of patterns[1]. In this study only supervised pattern classification procedures have been considered. A supervised classification could be subdivided into two main phases namely training phase and testing phase. In the training phase the classifier is learned by known categories (classes) of patterns and in the classification or the testing phase unknown patterns which were out of the training datasets assign class labels of train patterns for which the distance from the test pattern to the prototype(s) is minimized.

The performance of a classifier depends upon several factors. Some of the main factors are (i) number of training samples available to the classifier. (ii) Generalization ability i.e. its performance in classifying test patterns which were not used during the training stage. (iii) Classification error-some measured value based on the incorrect decision of the class labelling of any given pattern. (iv) Complexity - in some cases (due to classifier design) the number of features or attributes (dimensions) are relatively larger than the number of training samples usually referred as the curse of dimensionality, (v) speed-processing speed of training and/or testing phase(s) and (vi) storage-amount of parameters required to store after the training phase, for classification (testing) purposes[1].

For a given classifier model and a fixed number of training samples, the performance may depend on the

Corresponding Author: Alok Sharma, Signal Processing Lab., Griffith University, Brisbane, Australia

generalization capability (accuracy), speed and implementation cost (due to storage of information). The number of parameters required to perform classification task (testing) after the training procedure, is referred as 'total parameters'. For a given classifier we can associate the total parameters to the implementation cost of the classification system and the generalization capability may depend upon the type of parameters (distribution, values etc.) Used. The higher the total parameters required for classification task the costlier the system would be. Another important factor in classifier design is the speed or the processing time required to do the task. It is possible in a class that in two different instances the total parameter requirement is same but the processing time differs. We therefore want to reduce the total parameters and processing time but at the same time last sacrifice the classification accuracy. In other words, we search for the optimal classification accuracy or least classification error, involving as minimum total parameters and processing time as possible. This would allow the system to classify/recognize an object as quickly as possible at minimum cost.

Nearest neighbor (NN) classifier [2] is the most simple classifier found up till now. In NN classifier no special procedure is required to do the training. All the available data (as maximum as possible) are stored to perform classification, where each test pattern is compared for similarity with all the available training data (pattern). The test pattern is assigned the class label of that training pattern, which is the closest to the test pattern. A major drawback of NN approach is its large total parameter requirement to perform the classification task. For example, a dataset with 10 classes, having 5000 vectors or patterns in each class with 64 attributes or dimensions would require total parameters as follows:

$$\text{total parameters} = \text{class} \times \text{NoOfVec} \times \text{dimension}$$
$$= 10 \times 5000 \times 64 = 3.2 \times 10^6$$

If the dimension is very high (e.g., in image), then the total parameter requirement for the NN approach will be even more severe which would restrict the practical application of such approach. It can also be seen that an increase in the total parameter does not always lead to better performance. When train patterns and test patterns are closely matched then accuracy obtained by NN approach is good. But when the test patterns do not match with train patterns, NN approach provides poor performance (in terms of accuracy). In the unmatched pattern case the performance of the classifier system does not improve by increasing the total parameters.

The classification accuracy of NN approach can be improved by making the decision of a test pattern for class labelling based on *k* nearest patterns. This method is known as *k*-Nearest Neighbor (KNN)[2] technique. The total parameter requirement for kNN approach is same as that of NN approach except for the computational demand, which is severe in the former approach.

The implementation cost of the classification system could be reduced by estimating each class by a single prototype, usually a centroid. This would help in decreasing the total parameter requirement for the classification task but could be at the price of classification accuracy. This type of classifier is known as Minimum Distance Classifier (MDC). The goal of MDC is to correctly label as many patterns as possible. It provides the minimal total parameter requirement and computational demand. The MDC method finds centroid of classes and measures distances between these centroids and the test pattern. In this method, the test pattern belongs to that class whose centroid is the closest distance to the test pattern. Taking the same above example of 10 classes, the total parameter requirement for the MDC would be just 640, which is about 1/5000 as compared to NN approach. Usually classification accuracy is sacrificed to get this advantage of extremely low processing time and total parameter requirement. MDC is used in many pattern classification applications[3-7] including disease diagnostics[8], classification of digital mammography images[9] and optical media inspection[10].

The natural extension of single prototype is multi prototype, where each class is estimated by several prototypes like in Vector Quantization (VQ)[11,12]. VQ based classifiers are also referred as local classifiers since their partition each class into several disjoint regions or local regions and estimate each region by a prototype (centroid) usually referred as a codeword. The set of codewords is known as codebook of the system. The aim of VQ technique is to find the codebook that minimizes the expected distortion between pattern x and the centroid of j^{th} disjoint region (μ_j) i.e. $D = E[\min_j(\| x - \mu_j \|)]$ where $E[\cdot]$. denotes expectation with respect to x. So the training procedure is to find the codebook and store it for classification tasks. Increasing the number of codewords per class would increase the performance up to some extent but it would also augment the total parameter requirement and processing time. VQ technique is applied in several areas of pattern compression and classification[13], which include image classification[14] speech coding or speech compression[15], speaker recognition[16], high range resolution signature identification[17] and image coding[18].

Another way of performing classification is by utilizing linear subspace classifiers[19,20]. Here each class is represented by its Karhunen-Loéve transform (KLT)[2] or Principal Component Analysis (PCA). The objective of PCA is to find a global linear transform of giving patterns in the feature space and produce class-independent or class-dependent basis vectors. The first basis vector is in the direction of maximum variance of

the given data. The remaining basis vectors are mutually orthogonal and in order, maximize the remaining variances subject to the orthogonality condition. The principal axes are those orthonormal axes onto which the remaining variables under projection are maximized. These orthonormal axes are given by the dominant eigenvectors (i.e. those with the largest associated eigenvalues) of the covariance matrix.

Class-independent PCA finds those h orthonormal axes (subspace dimension) from *d*-dimensional datasets ($h < d$), where h dominant eigenvectors are from the KLT of the data correlation matrix $\Sigma = E[xx^t]$ which is in fact a covariance matrix with zero mean[21]. Class-independent PCA cannot be used for classification purposes since all the classes are scattered over the feature space with different centroid values or mean and variances for each class making impossible to preserve the individual class information by a single KLT for the entire train samples. Therefore dominant eigenvectors are taken for each class separately (class-dependent). For a *c*-class problem, covariance matrix will be given by:

$$\Sigma_j = E[(x-\mu_j)(x-\mu_j)^t] \quad \text{for } j = 1,2,...c$$

where only those *x*, that belong to the j^{th} class have been taken in the expectation function at a time. It has been seen that the subspace classification is further improved by its local linear extension[22]. Here the performance depends upon the subspace dimension and the number of local regions. Kambhatla and Leen[22] and Kambhatla[23] have shown local linear PCA or VQPCA for representation purposes. The goal of VQPCA is to minimize the mean squared reconstruction error $E[\| x - \hat{x} \|^2]$ where $\hat{x}$ is the reconstructed pattern of *x*. Kambhatla[23] showed VQPCA using Euclidean distance (VQPCA-Euc) and VQPCA using reconstruction distance (VQPCA-rec). VQPCA-rec is a better technique than VQPCA-Euc for representation purposes in terms of achieving lesser reconstruction error, but this achievement comes with the expense of higher total parameter requirement and computational demand. For example, taking the same 10 class problem, where each class is subdivided into 4 disjoint regions (local regions), this would require storage of *dxd* (64x64) eigenvector set for each disjoint region together with other parameters (centroid of disjoint region) i.e.:

$$\text{total parameters} = \text{parameters due to eigenvectors} + \text{parameters due to centroid}$$

$$\text{total parameters (VQPCA-rec)} = (d \times d) * \text{class} * \text{level} + (d \times 1) * \text{class} * \text{level}$$

$$\text{total parameters (VQPCA-Euc)} = (d \times h) * \text{class} * \text{level} + (d \times 1) * \text{class} * \text{level}$$

where the term *level* is the number of disjoint regions or local regions per class and h<d. This yield total parameter requirement for VQPCA-rec 1.66×10^5 (for d=64), whereas 7680 (for $h = 2$) for VQPCA-Euc which is $1/(\frac{d+1}{h+1})$ compared to VQPCA-rec. Although the VQPCA-rec model exhibits slight improvement over VQPCA-Euc model, it severely increases the total parameter requirement and computational demand. This would increase the implementation cost and processing time of the classification system. Considering the implementation cost and computational demand we opted for an economical model (VQPCA-Euc) to train the system. Hereafter VQPCA-Euc model will be referred as VQPCA model. Some modification is required in VQPCA model prior to use as a classifier. The current VQPCA model first partitions the data space into disjoint regions and then performs local PCA about each cluster (referred as a disjoint region of a class) center. This is ideal for representation purposes but for the classification task a minor change in distance measurement is required which should reflect the distance of a test pattern from the centroid and dominant eigenvectors of each disjoint region concurrently. The VQPCA model as a classifier does not exhibit very encouraging results but still can be used to perform the classification task. Nonetheless it can be shown that VQPCA model as a classifier behaves satisfactorily in terms of obtaining reasonably well percentage accuracy at low total parameter requirements and processing time.

The performance of VQPCA as a classifier could be significantly improved by combining the linear distances of VQ and VQPCA. The normalized reconstruction distance measure $\| x - \hat{x} \|$ and the normalized distance between the test pattern and the center of disjoint region $\| x - \mu_j \|$, are combined linearly to form a new distance measure for the classification. This distance measure would minimize the combination of the mean squared reconstruction error (MSE) $E[\| x - \hat{x} \|^2]$ and the expected distortion $E[\| x - \mu_j \|]$. Each distance added together may have its own local regions in the feature space where it performs the best. We have introduced this linear combination of distance (LCD) technique and shown in this study that it is a better classifier with no extra total parameter requirement than VQPCA. Classification results obtained by LCD exhibit significant improvement over MCD, VQ, VQPCA, NN and kNN classifiers in terms of achieving higher percentage accuracy or lower classification error and at the same time maintaining the total parameters requirement and processing time as minimum as possible. Consequently, this would allow classification or recognition of the objects as quickly as possible at minimum cost.

Conventional classifiers: The style of notations is adopted from Duda and Hart[24]. In all the discussions

ω_i denotes the state of nature or class label of i^{th} class in a *c*-class problem, $\mathcal{X}$ denotes the set of *n* train samples, $\Omega = \{\omega_i : i = 1, 2, \ldots, c\}$ be the finite set of *c* states of nature and let θ' be the class label of train pattern or prototype such that $\theta' \in \Omega$ .The set $\mathcal{X}$ can be separated by class into *c* subsets $\mathcal{X}_1$, $\mathcal{X}_2$,…, $\mathcal{X}_c$, with the samples in $\mathcal{X}_i$ belonging to ω_i:

$\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ where $x_j \in R^d$ (*d*-dimensional hyperplane):

$$\mathcal{X}_i \subset \mathcal{X} \text{ and } \mathcal{X}_1 \cup \mathcal{X}_2 \cup \ldots \cup \mathcal{X}_c = \mathcal{X}$$

Let n_i denote the number of samples in the subset $\mathcal{X}_i$, therefore $\sum_{i=1}^{c} n_i = n$.

Figure 1 illustrates the class labelling of a test pattern and the relationship between the label of the prototype (θ') and the label of the class (ω). The prototype could be a train pattern, a centroid, a KLT or a group of centroid and KLT depending upon the type of classifier is used. In Fig. 1 two-class problem is considered where each class consists of 3 prototypes. Each of the class is assigned a unique label namely ω_p and ω_q such that $(\omega_p, \omega_q) \in \Omega$. The class labels of the prototypes are $\theta'_i, \theta'_j, \theta'_k$ and $\theta'_l, \theta'_m, \theta'_n$ such that:

$$\omega_p = \theta'_i = \theta'_j = \theta'_k \text{ and } \omega_q = \theta'_l = \theta'_m = \theta'_n$$

The class label of prototype is assigned to a test pattern *x* which is the closest to the prototype based on some distance measurements or conditional probabilities. Therefore if *L(x)* denotes the class label of a test pattern *x* then from the figure $L(x) = \theta'_l = \omega_q$.

NN classifier: The procedure for NN classifier can be subdivided into two main phases namely, training phase and testing or classification phase. In the training phase all the available patterns $\mathcal{X}$ with their corresponding class label information are stored for classification purpose. The total parameter requirement for the NN approach is given by:

$$\text{total parameters} = d \times \sum_{i=1}^{c} n_i = d \times n \quad (1)$$

It can be seen from equation 1 that total parameters depend upon the attribute or dimension *d*, number of class and number of training patterns. In many practical applications the values of *d* and *n* are very large which severely affects the storage requirements and processing time, increasing the cost and reducing the speed of the classifier system.

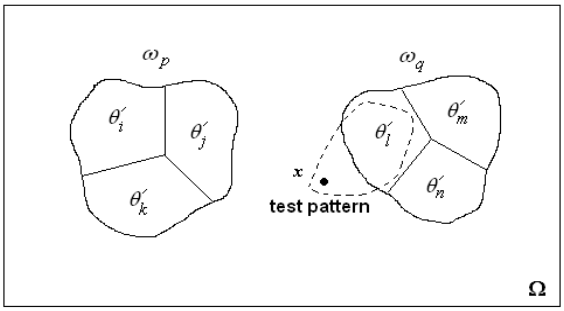

Fig. 1: Class labelling of a test pattern in a two-class problem

kNN classifier: kNN classifier is a generalized form of NN classifier. In this approach *k* nearest train patterns to a test pattern *x* is collected. The test pattern is assigned the class label which has the majority of *k* collected patterns. The training phase of the kNN classifier is similar to NN classifier where all the training patterns together with their class label information are stored for the later use. The total parameter requirement is also same as NN approach. The processing speed of KNN classifier is slower than NN classifier due to the searching of *k* nearest patterns for each of the test patterns. The classification accuracy may improve with the increase in the value *k*. This improvement is usually observed when the test patterns and the train patterns are closely matched. However, in some cases when the test patterns and the train patterns do not match the classification accuracy is poor. In this case increasing the value *k* may not improve the classification accuracy of the system.

MDC classifier: In MDC classifier each class $\mathcal{X}_i$ is represented by single prototype, which is usually the centroid of the class in the feature space. It requires a minimal total parameter requirement and least computational demand. The total parameter requirement for MDC is:

$$\text{total parameters} = d \times c$$

Which is $c / \sum_{i=1}^{c} n_i$ as compared to NN or kNN classifier. This advantage of the lower total parameter requirement and fast computation may achieve by sacrificing some classification accuracy.

VQ classifier: VQ classifier is the further extension of MDC classifier. Here each class is represented by multiple prototypes. VQ partitions a class into several disjoint regions in the feature space usually known as Voronoi regions[12]. The center of Voronoi regions (prototype) is referred as codeword of the classifier and a set of codewords is known as codebook of the classifier system. The aim of VQ is to produce a

codebook that minimizes the expected distortion $E[\| x - \mu_j \|]$. See Linde *et al.*[11] for details. The total parameter requirement is $d \times (Q \times c)$ where *Q* is the level of classifier i.e. Number of disjoint regions or code words for each of the classes.

PCA classifier: Class dependent PCA is considered for classification where each class is represented by its KLT. In a *d*-dimensional feature space left Σ_j and μ_j denote covariance matrix and centroid, f class χ_j in a *c*-class problem respectively, $\hat{x}$ be the reconstructed pattern of *x*, then the goal of the training phase of PCA classifier is to find eigenvectors w_i such that the following criteria is satisfied:

$$\Sigma_j w_i = \lambda_i w_i \quad (2)$$

where, λ_i denotes eigenvalues corresponding to w_i which is obtained by minimizing MSE $E[\| x - \hat{x} \|^2]$. The total parameter requirement for PCA classifier is:

$$\text{total paramaters} = \text{centroid\_paramters} + \text{eigenvector\_parameters}$$
$$\text{total paramaters} = c \times d + c \times (d \times h) = cd(h+1)$$

where, h < d is the number of eigenvectors used.

VQPCA as a classifier: In this approach, firstly, the set of train patterns are partitioned into disjoint regions by applying the VQ technique for each class separately and then KLT is performed on each of the disjoint regions or local region center[22]. The aim of VQPCA is to minimize MSE $E[\| x - \hat{x} \|^2]$ in the local regions. To illustrate training and classification procedures let *Q* be the number of disjoint regions or levels per class. (Details of the training procedure are given in Kambhatla[23]. VQPCA can also be trained using splitting technique[25]).

Training

Step 1: Take train patterns $\chi_i \subset \chi$ of class label ω_i at a time for consideration, where $i = 1, 2, ..., c$.

Step 2: Apply VQ technique and partition χ_i into *Q* disjoint regions; for all $i = 1, 2, ..., c$.

Step 3: For each disjoint region compute centroid μ_j and covariance matrix Σ_j where $j = 1, 2, ..., (c \times Q)$.

Step 4: Evaluate $d \times h$ rectangular matrix of eigenvectors $W_j = \{w_l : l = 1, 2, ..., h\}$ for each disjoint region where $h<d$ and w_i is from equation 2; arrange the obtained eigenvectors such that its corresponding eigenvalues are in descending order. Let the class label of eigenvector set W_i be $\theta'_j \in \Omega$.

Step 5: Store W_j and μ_j with their corresponding class information for classification.

The total parameter requirement for VQPCA can be given by:

$$\text{total paramters} = \text{parameters\_centroids} + \text{paramters\_eigenvectors}$$
$$\text{total paramters} = Q \times d \times c + Q \times (d \times h) \times c = Qdc(h+1)$$

Which is *Q* times the total parameter requirement of PCA classifier.

If VQPCA is used for representation purposes then in the decoding step (here classification) firstly the closest disjoint region to a test pattern *x* is computed. Once the closest region is obtained, the next step is to use its corresponding eigenvector and centroid information to compute reconstructed pattern $\hat{x}$. For classification VQPCA procedure would provide no better performance than VQ technique since the decision would lie only on the closest disjoint region to the test pattern *x* and the computation of KLT for disjoint regions may become redundant. Therefore a procedure for decision making of a test pattern should be adopted that uses both the centroid and direction (eigenvector) information in parallel.

Classification:

Step 1: Compute reconstruction distance δ_j between a test pattern *x* and its reconstructed pattern $\hat{x}$:

$$\delta_j = \| x - \hat{x} \|$$
$$= \| (I - W_j W_j^t)(x - \mu_j) \| \quad \text{for } j = 1, 2, ..., (Q \times c)$$

Step 2: Find the argument for which the reconstruction distance is minimized:

$$k = \arg \min_{j=1}^{Q \times c} \delta_j$$

Step 3: Assign class label $\omega_r = \theta'_k$ to the test pattern *x*, where $\theta'_k \in \Omega$.

Thus, it can be seen that step 1 computes the error of reconstruction distance by using direction and centroid information in one single step for the classification.

LCD classifier: The LCD is a combination of VQ and VQPCA techniques. Empirical results show significant improvement of LCD classifier over previously discussed classifiers in terms of getting higher percentage accuracy with the total parameter requirement no more than VQPCA approach. In our approach the training phase of the classifier is identical to VQPCA classifier thus the total parameter requirement for LCD approach is same as VQPCA approach. However the classification procedure differs. In the classification phase the distance used in VQ classification and the distance used in VQPCA classification is added together with some weighting to form a new distance measure. This combination or addition may reduce expected distortion $E[\| x - \mu_j \|]$

and MSE or root-MSE $E[\| x - \hat{x} \|]$, overall producing improved results for the combination. The improved results achieved could be due to each of the constituent distance performing the best in their local regions in the feature space.

The generalization capability or classification accuracy of a classifier depends on the type of distribution or values used for training and/or testing the classifier. For e.g. If training patterns of each class are spherically distributed, dense, well separated with each other and test pattern are closely matched with their train patterns then techniques such as MDC, VQ, NN and kin may perform better; if outliers are present in the training patterns then techniques such as PCA or VQPCA may give poor performance. However for Gaussian data with matching train and test conditions PCA may provide reasonably high classification accuracy[1] and VQPCA and LCD may provide even better performance than PCA. In the presence of outliers and complex distributions (unmatched train and test conditions) LCD may provide better performance than other techniques.

The concept of combination of multiple classifiers has been previously applied by Xu *et al.*[26] for handwriting recognition. They have illustrated the combination using some basic classifiers such as Bayesian and kNN and shown three categories of combination which depend upon the levels of information available from the classifiers. Jacobs *et al.*[27] suggested supervised learning procedure for systems composed of many separate expert networks. Ho *et al.*[28] used multiple classifier system to recognize degraded machine-printed characters and words from large lexicons. Tresp and Taniguchi[29] presented modular ways for combining estimators. Woods *et al.*[30] and Woods [31] presented a method for combining classifiers that use estimates of each individual classifier's local accuracy in small regions of feature space surrounding a test pattern. Zhou and Imai [32] showed a combination of VQ and multilayer perceptron (MLP) for Chinese syllable recognition. Alimoglu and Alpaydin[33] used the combination of two MLP neural networks for handwritten digit recognition. Kittler *et al.*[34,35] developed a common theoretical framework for combining classifiers which use distinct pattern representations. Breukelen van and Duin[36] showed the use of combined classifiers for the initialization of neural network. Alexandre *et al.*[37] combined classifiers using weighted average after Turner and Gosh[38]. Ueda[39] presented linearly combining multiple neural network classifiers based on statistical pattern recognition theory. Senior[40] used combination of classifiers for fingerprint recognition. Lei *et al.*[41] demonstrated a combination of multiple classifiers for handwritten Chinese character recognition and Yao *et al.*[42] used a combination based on fuzzy integral and Bayes method. Similarly several other research work on combinational classifiers have been reported in the literature.

In our approach the training phase parameters μ_j (centroid) and W_j (eigenvector set) are stored with the class label $\theta'_j \in \Omega$ information for the use in the classification phase which is same as the training phase of VQPCA approach. Let in a *c*-class problem each Class is separately partitioned into *Q* disjoint regions then the classification phase of the LCD approach can be illustrated as follows:

Classification

Step 1: Compute the distance δ_j^1 between a test pattern *x* and the centroid μ_j of the disjoint region:

$$\delta_j^1 = \| x - \mu_j \| \text{ for } j = 1, 2, ..., (Q \times c)$$

Step 2: Compute the reconstruction distance δ_j^2 between a test pattern *x* and its reconstructed pattern $\hat{x}$:

$$\delta_j^2 = \| x - \hat{x} \| = \| (I - W_j W_j^t)(x - \mu_j) \| \quad \text{for } j = 1, 2, ..., (Q \times c)$$

Step 3: Normalize distance δ_j^1 and δ_j^2 to eliminate the difference in their amplitudes that would allow them to contribute equally in decision making.

$$\hat{\delta}_j^1 = \delta_j^1 / \max_{j=1}^{Q \times c}(\delta_j^1) \text{ and } \hat{\delta}_j^2 = \delta_j^2 / \max_{j=1}^{Q \times c}(\delta_j^2)$$

Step 4: Add distance $\hat{\delta}_j^1$ and $\hat{\delta}_j^2$:

$\hat{\delta}_j = \alpha \hat{\delta}_j^1 + (1 - \alpha)\hat{\delta}_j^2$ for $j = 1, 2, ..., (Q \times c)$, where α is a weighting constant in the range $[0,1]$.

Step 5: Find the argument for which the combined distance is minimized:

$$k = \arg \min_{j=1}^{Q \times c} \hat{\delta}_j$$

Step 6: Assign class label $\omega_r = \theta'_k$ to the test pattern *x*, where $\theta'_k \in \Omega$.

The classification phase of LCD technique is simple, computationally inexpensive and attains high classification accuracy or low classification error. The distance $\hat{\delta}_j$ in the classification phase depends on the weighting constant α and the two normalized distance $\hat{\delta}_j^1$ and $\hat{\delta}_j^2$. The weighting constant α (in step 4) is a positive constant in the range $[0,1]$. The appropriate value for α should be taken since bad selection may lead to poor classification accuracy. The two normalized distances $\hat{\delta}_j^1$ and $\hat{\delta}_j^2$ are classification distance of VQ and VQPCA techniques respectively.

Choice of α: The optimum or close to optimum performance by LCD classifier can be obtained by

selecting the appropriate value of α empirically. We have used speech data[43] and image data[44,45] to select the value of α. In this study we have taken α as a numerical constant, however, one can also take α as a probabilistic model which would depend on a test pattern and the distribution of train patterns. This may increase the computation and storage requirements. The discussion about α as a probabilistic model is beyond the scope of this study. In Fig. 2 and 3 classification accuracy for LCD technique is computed for dimension *h* and level *Q*, where $h = 1, 2..., 4$ and $Q = 1, 2, 4, 8, 16$. The values of α are $0.1, 0.2, ..., 0.9$, where choosing α values close to 0.1 and 0.9 will give performance similar to VQPCA approach and VQ approach respectively.

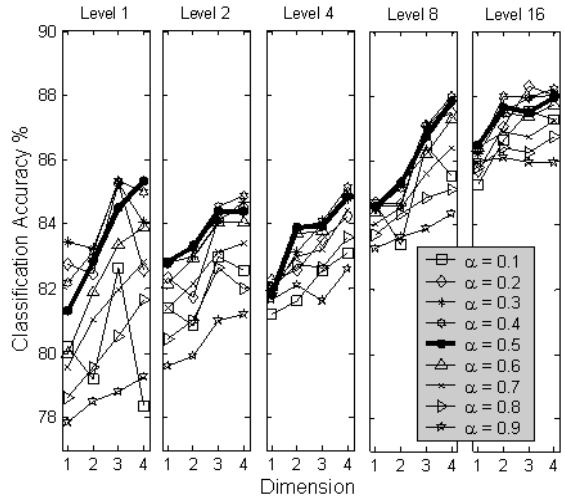

Fig. 2: Classification accuracy for different values of α on image data

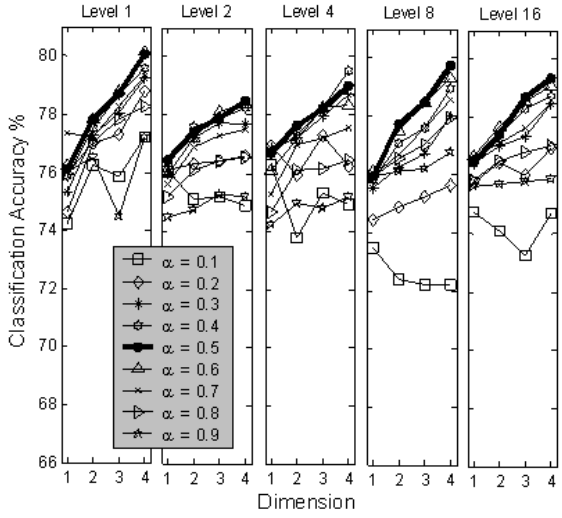

Fig. 3: Classification accuracy for different values of α on speech data

Diverting either upwards ($\alpha = 0.6, ..., 0.9$) or downwards ($\alpha = 0.4, ..., 0.1$) from the center value of α (0.5) will make the distance $\hat{\delta}_j$ biased for $\hat{\delta}_j^1$ or $\hat{\delta}_j^2$ respectively. It can be observed from Fig. 2 and 3 that at $\alpha = 0.5$ classificatiothe LCD techniquetained by LCD technique (in Fig. 2 and 3 denoted by bold lines) is close to optimum. This implies that when the distance $\hat{\delta}_j^1$ and $\hat{\delta}_j^2$ contribute equally in the decision making for a test pattern in the feature space then the classification accuracy is close to optimum. Thus we have taken $\alpha = 0.5$.

Experimentation: For all the experiments two sets of machine learning corpuses have been utilized namely TIMIT database[43] for speech classification and Sat-Image dataset[44,45] for image classification. From the TIMIT corpus a set of 10 distinct monothongal vowels is extracted, then each vowel is divided into three segments and each segment is used in getting Mel-frequency Cepstral coefficients with energy-delta-acceleration (MFCC_E_D_A) feature vectors [46]. A total of 9357 MFCC_E_D_A vectors of dimension 39 for training sessions and a separate set of 3222 vectors for classification are utilized. The second dataset is Sat-Image which consists of 6 distinct classes with 36 dimensions. A sum of 4435 feature vectors is used to train the classifier and a different set of 2000 vectors is used for verifying the performance of the classifier.

In the first part of the experimentation, classification accuracy is measured for all the classifiers given some fixed parameters. Here the accuracy is a function of dimension *h* and level *Q*, where Q = 1, 2, 4, 8, 16 and h = 1, 2,..4 for all the levels, except for Q = 8, where h = 1, 2,...,10. Level 8 (Q = 8) is taken at random for dimension h = 1, 2,...,10 to get a general understanding of how the dimension affects the classification accuracy if it is increased continuously.

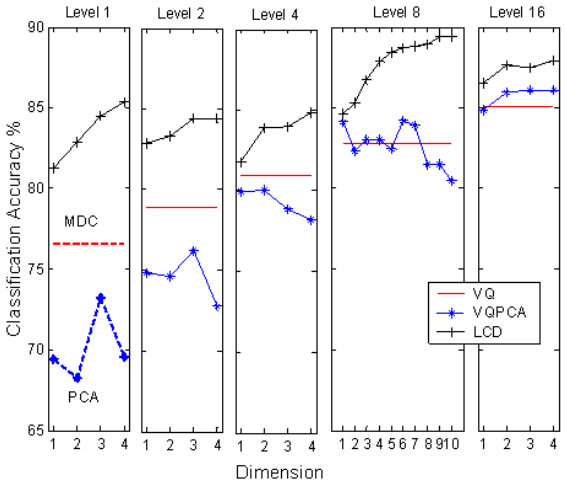

Fig. 4: Classification accuracy vs. dimensions and levels using MDC, VQ, PCA, VQPCA and LCD on image data sets

Not all the techniques depend upon both the dimension *h* and level *Q*; VQ depends upon levels, PCA depends upon the dimensions, MDC, NN and kNN depend neither upon dimensions nor on levels, only VQPCA and LCD depend upon dimensions as well as levels. Fig. 4 (image dataset) and Fig. 5 (speech dataset) illustrates the classification accuracy for MDC, VQ, PCA, VQPCA and LCD techniques and Table 1 depicts classification accuracy for NN and kNN techniques. Usually the MDC technique is a special case of VQ when $Q=1$, that's why it is represented in the column of Level 1 in Fig. 4 and 5.

Table 1: Classification accuracy for NN and kNN techniques on image and speech datasets

Technique		Classification accuracy using image dataset	Classification accuracy using speech dataset
NN		90.30	74.05
kNN	3	90.45	75.67
	5	89.70	76.82
	7	90.05	77.56
	9	90.05	78.15
	11	89.35	78.34

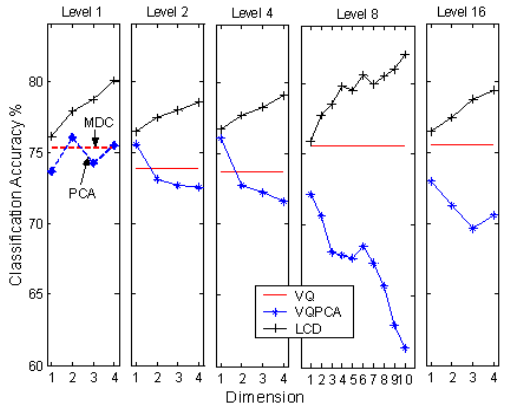

Fig. 5: Classification accuracy vs. dimensions and levels using MDC, VQ, PCA, VQPCA and LCD on speech dataset

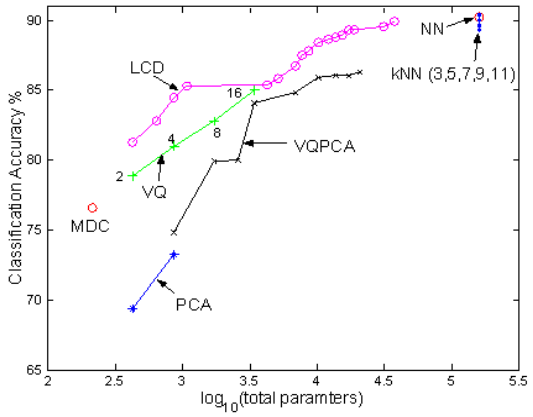

Fig. 6.1: Classification accuracy vs. $\log_{10}$ (total parameters) on image datasets

It can be observed from Fig. 4 (image datasets) that MDC is giving better classification accuracy than PCA; VQ is producing a higher classification accuracy at Level 2 and Level 4 than VQPCA, but VQPCA is showing improvement over VQ technique at level 8 and level 16. It is also clear that LCD is performing better than MDC, VQ, PCA and VQPCA at all the levels and dimensions. Increasing the dimension at any given level is improving the classification accuracy of LCD technique. At level 8 and dimension 10 the classification accuracy of an LCD is 89.2% which is very close to NN and kNN techniques. It should be noted that NN and kNN techniques produce similar classification accuracy as LCD technique but their processing time and total parameter requirement are severely expensive.

Furthermore, it can be observed from the experiment on speech data (Fig. 5) And Table 1 that MDC is giving better classification accuracy than NN technique; PCA is improving at dimension 2 over MDC technique; VQPCA is producing a better classification accuracy over VQ technique at levels 2 and 4 for dimension 1 but deteriorating at level 8 and level 16. LCD is exhibiting better performance than all the techniques including NN and kNN. The classification accuracy is improving with the increase in dimension at any given level. The classification accuracy by NN and kNN is quite poor for speech data. This may be due to the testing data not matching with their training data.

In the second part of experimentation, classification accuracy is computed as a function of total parameters and processing time. This would give 3D plot where *x* and *y* axes represent total parameters and processing time and *z*-axis represents classification accuracy. For simplicity, a 3D plot is split into two 2D plots, where one plot shows classification accuracy versus total parameters and the other plot shows classification accuracy versus processing time for the corresponding values of total parameters.

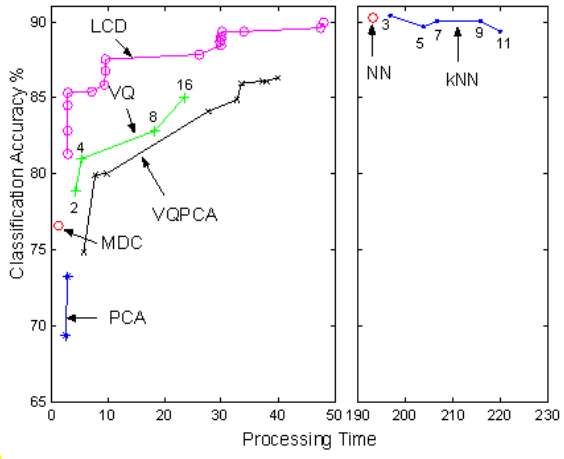

Fig. 6.2: Classification accuracy vs. Processing time on image datasets

The level is taken as Q = 1, 2, 4, 8, 16 and data session $h = 1, 2, ..., 10$ for image data set and $h = 1, 2, ..., 12$ for speech dataset. Figure 6.1 and 6.2 show classification accuracy versus total parameters in logarithmic scale and classification accuracy versus processing time respectively, using all the techniques on image dataset.

For LCD technique, as presented on the Fig. 6.1 and 6.2, the first value of classification accuracy is 81.3% at total parameter $10^{2.636}$ (Fig. 6.1) which takes processing time of 2.94 units (Fig. 6.2). The next reported value of classification accuracy in Fig. 6.1 and 6.2 is only those which provide better classification accuracy than the present value, i.e. Those values are plotted next in the figures which are given the improvement in classification accuracy compared to the previous value. This would help to describe that to achieve a certain range of classification accuracy what is the total parameter requirement and its corresponding processing time. A similar strategy is opted for VQPCA and PCA techniques. For VQ technique there are only four levels and all of them are given which are denoted by 2,4,8 and 16 in the Fig. 6.1 and 6.2. MDC and NN have only one value and kNN has got 5 values for $k = 3,5,7,9,11$ which is depicted in the same figures.

It can be observed from the Fig. 6.1 and 6.2 that the MDC has a minimal total parameter requirement and processing time but the classification accuracy is quite poorly around 76.6%. The other techniques with the same total parameter requirement but with different processing timings are PCA, VQ and LCD (at level 1). Though the processing time is very low for PCA (around 2.53 to 2.99 time units), the performance is quite poor giving classification accuracy in the range of 69.4% to 73.3% which is even lower than MDC. With the same total parameter requirement VQ gives much better performance than PCA in terms of accuracy but the processing time increases as the levels increase towards 16. The classification accuracy of VQPCA is quite poor at the beginning. As the total parameter requirement increases it gives reasonably good results but at the expense of high processing time. It is evident that LCD technique gives high classification accuracy at low total parameter requirement and processing time, for e.g. it gives 85.4% accuracy at $10^{3.033}$ total parameters using only 3.00 units processing time whereas the maximum accuracy obtained by VQ is 85.1% at $10^{3.539}$ total parameters using 23.41 units processing time and VQPCA gives 84.9% at $10^{3.840}$ using 32.81 units processing time. The maximum accuracy achieved by the LCD technique (when Q< 16 and $h \le 10$) is 90.0% at $10^{4.580}$ using 48.12the NN techniquessing time which is very close to NN KNNhnique (90.3%) and close to the maximum of kNN (for $k = 3$) technique (90.5%). However the processing time for NN and kNN techniques are 193.37 units and from 196.89 to 220.01 units (for $k = 3,5,7,9,11$) respectively and the total parameter requirement for both the techniques is $10^{5.203}$, which is quite expensive as compared to LCD and other techniques. Figure 7.1 and 7.2 show classification accuracy vs. total parameters on logarithmic scale and classification accuracy vs. Processing time respectively for all the techniques on speech dataset. The plotting scheme is similar to that applied for Fig. 6.1 and 6.2.

It is evident from Fig. 7.1 and 7.2 that LCD technique is performing better than all the other techniques including NN and kNN in terms of achieving higher classification accuracy at low total parameter requirement and low processing time. The classification accuracy of NN technique is even poorer than MDC, PCA and VQ techniques; this means that increasing total parameters does not always help in improving the classification accuracy. The maximum classification accuracy for LCD technique is 84.1% in $10^{3.670}$ using 8.74 units processing time, whereas the nearest technique in terms of accuracy is kNN which is giving 78.3% (for $k = 11$) in $10^{5.562}$ using 794.08 units processing time.

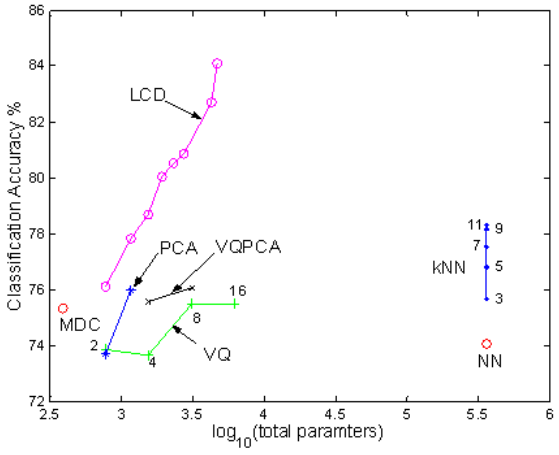

Fig. 7.1: Classification accuracy vs. $\log_{10}$ (total parameters) on speech dataset

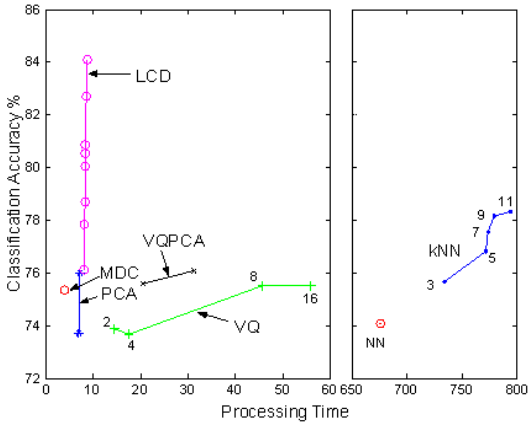

Fig. 7.2: Classification accuracy vs. Processing time on speech dataset

It can be concluded from the experiments on image data set and speech dataset that LCD technique outperforms MDC, PCA, VQ, VQPCA, NN and KNN techniques in terms of getting reasonably accepted classification accuracy and at the same time maintaining the minimal total parameter requirement and processing time. This would enable the user to classify a given object accurately and quickly with minimal implementation cost.

CONCLUSION

A survey on basic classifiers namely MDC, VQ, PCA, NN and kNN was given. Their classification procedures were illustrated. Then we looked at VQPCA technique which is normally used for representation purposes. We showed how to use VQPCA for classification purposes. However, we found that VQPCA did not give a very encouraging performance as a classifier but this gave us initiative to develop combined classifiers.

Next we presented LCD technique which is the combination of VQ and VQPCA techniques. By combining the classifiers we found that the performance improved significantly which was not possible by using either VQ or VQPCA individually. The performance of LCD technique is found to be better than all the other presented techniques. Thus it can classify a given object more accurately at very low implementation cost and processing time, which was demonstrated using speech and image datasets.

It was found that when the weighting coefficient α was close to 0.5 the LCD technique gave close to optimum performance, i.e. when VQ and VQPCA techniques contribute equally in the decision making of a test pattern then the performance is close to optimum.

REFERENCES

1. Jain, A.K., R.P.W. Duin and J. Mao, 2000. Statistical pattern recognition: a review. IEEE Trans. Pattern Anal. Machine Intelligence, 22: 4-37.
2. Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition. Academic Press Inc., Hartcourt Brace Jovanovich, Publishers.
3. Di Maio V. and F. Marciano, 2003. Automatic classification of neural spike activity: an application of minimum distance classifiers. Cybernetics and Systems, 34: 173-192.
4. Paclik, P. and R.P.W. Duin, 2003. Dissimilarity-based classification of spectra: Computational issues. Real-time Imaging, 9: 237-244.
5. Sahin, F., 2000. A radial basis function approach to a color image classification problem in a real time industrial application. PhD Thesis, State University, Virginia.
6. Datta, P. and D. Kibler, 1997. Symbolic nearest mean classifiers. Proc. of the 14th Natl. Conf. On Artificial Intelligence, San Mateo, CA, pp: 82-87.
7. Griguolo, S., 1994. Pixel-by-pixel clustering for vegetation monitoring. Intl. Conf. on “Alerte précoce et suivi de l'Environment”, Niamey, Niger.
8. Lewenstein, K. and M. Chojnacki, 2004. Minimum distance classifiers in coronary artery disease diagnosing. Modelling in Mechatronics, Kazimierz Dolny, Poland.
9. Lambrou, T., A.D. Linney, R.D. Speller and A. Todd-Pokropek, 2002. Statistical classification of digital mammograms using features from the spatial and wavelet domains. Medical Image Understanding and Anal., Portsmouth, UK.
10. Toth, D., A. Condurache and T. Aach, 2002. A two-stage-classifier for defect classification in optical media inspection. 16th Intl. Conf. On Pattern Recognition (ICPR'02), 4: 373-376.
11. Linde, Y., A. Buzo and R.M. Gray, 1980. An algorithm for vector quantization design. IEEE Trans. On Comm., COM-28, 1: 84-94.
12. Gray, R.M., 1984. Vector quantization. IEEE ASSP Magazine, pp: 4-29.
13. Wesel, R.D. and R.M. Gray, 1994. Bayes risk weighted VQ and learning VQ. Proc. Data Compression Conf. (DCC’94), UT, USA, pp: 400-409.
14. Potlapalli, H., M.Y. Jaisimha, H. Barad, A.B. Martinez, M.C. Lohrenz, J. Ryan and J. Pollard, 1989. Classification techniques for digital map compression. Proc. Of the 21st Southeastern Symp. On System Theory, Tallahassee, FL, USA, pp: 268-272.
15. Makhoul, J., S. Roucos and H. Gish, 1985. Vector quantization in speech coding. Proc. Of the IEEE, 73: 1551-1588.
16. Soong, F.K., A.E. Rosenberg and B. Juang, 1987. A vector quantization approach to speaker recognition. AT&T Technical Jnrl., 66: 14-26.
17. Dsung, T.P., 1998. Applications of unsupervised clustering algorithms for aircraft identification using high range resolution radar. Proc. IEEE National Aerospace and Electronics Conf., OH, USA, pp: 228-235.
18. Arvind, R. and A. Gersho, 1986. Low-rate image coding with finite-state vector quantization. In Proc. ICASSP pp: 137-140.
19. Oja, E., 1983. Subspace Methods of Pattern Recognition. Research Studies Press, New York.
20. Oja, E. and J. Parkkinen, 1984. On subspace clustering. Seventh Intl. Conf. On Pattern Recognition, 2: 692-695.
21. Oja, E., 1992. Principal components, minor components and linear neural networks. Neural Networks, 5: 927-935.
22. Kambhatla, N., Leen, T.K., 1997. Dimensionality reduction by local PCA. Neural Computation, 9: 1493-1516.

23. Kambhatla, N., 1995. Local models and Gaussian mixture models for statistical data processing. PhD Thesis, Oregon Graduate Inst. Of Sci. and Technology.
24. Duda, R.O. and P.E. Hart, 1973. Pattern Classification and Scene Analysis. John Wiley and Sons, New York.
25. Sharma, A., K.K. Paliwal and G.C. Onwubolu, 2006. Splitting technique initialization in local PCA. J. Computer Sci., 2: 53-58 (in print).
26. Xu, L., A. Krzyżak and C.Y. Suen, 1992. Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Trans. On Systems Man. and Cybernetics, 22: 418-435.
27. Jacobs, R.A., M.I. Jordan, S.J. Nowlan and G.E. Hinton, 1991. Adaptive mixtures of local experts. Neural Computation, 3: 79-87.
28. Ho, T.K., J.J. Hull and S.N. Srihari, 1994. Decision combination in multiple classifier systems. IEEE Trans. On Pattern Anal. and Machine Intelligence, 16: 66-75.
29. Tresp, V. and M. Taniguchi, 1995. Combining estimators using non-constant weighting functions. In G. Tesauro, D.S. Touretzky, T.K. Leen (Eds). Advances in Neural Info. Processing Systems 7, MIT press, Cambridge.
30. Woods, K., K. Bowyer and W.P. Kegelmeyer, 1996. Combination of multiple classifiers using local accuracy estimates. IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition CVPR '96, pp: 391-396.
31. Woods, K., 1997. Combination of multiple classifiers using local accuracy estimates. IEEE Trans. Pattern Anal. Machine Intelligence, 19: 405-410.
32. Zhou, L. and S. Imai, 1996. Chinese all syllables recognition using a combination of multiple classifiers. ICASSP, 6: 3494-3497.
33. Alimoglu, F. and E. Alpaydin, 1997. Combining multiple representations and classifiers for pen-based handwritten digit recognition. Intl. Conf. Document Analysis and Recognition, 2: 637-640.
34. Kittler, J., M. Hatef, R.P.W. Duin and J. Matas, 1996. On combining classifiers. Intl. Conf. Pattern Recognition, 2: 897-901.
35. Kittler, J., M. Hatef, R.P.W. Duin and J. Matas, 1998. On combining classifiers. IEEE Trans. Pattern Anal. Machine Intelligence, 20: 226-239.
36. Breukelen van, M. and R.P.W. Duin, 1998. Neural network initialization by combining classifiers. Intl. Conf. Pattern Recognition, 1: 215-218.
37. Alexandre, L.A., A.C. Campilho and M. Kamel, 2000. Combining independent and unbiased classifiers using a weighted average. Intl. Conf. Pattern Recognition, 2: 495-498.
38. Turner, K. and J. Gosh, 1999. Linear and order statistics combiners for pattern classification. In A. Sharkey, (Ed.). Combining Artificial Neural Nets, Springer-Verlag, pp: 127-161.
39. Ueda, N., 2000. The optimal linear combination of neural networks for improving classification performance. IEEE Trans. Pattern Anal. Machine Intelligence, 22: 207-215.
40. Senior, A., 2001. A Combination Fingerprint Classifier. IEEE Trans. Pattern Anal. and Machine Intelligence, 23: 1165-1174.
41. Lei, L., W. Xiao-Long and L. Bing-Quan, 2002. Combining multiple classifiers based on statistical methods for handwritten Chinese character recognition. Intl. Conf. Machine Learning and Cybernetics, 1: 252-255.
42. Yao, M., X. Pan, T. He and R. Zhang, 2002. An improved combination method of multiple classifiers based on fuzzy integrals. World Congress on Intelligent Control and Automation, 3: 2445-2447.
43. Garofalo, S.G., L.F. Lori, F.M. William, F.G. Jonathan, P.S. David and D.L. Nancy, 1986. The DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROMs. NIST.
44. Blake, C.L. and C.J. Merz, 1988. UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn, Irvine, CA, University of Calif., Dept. Of Information and Comp. Science.
45. Michie, D., D.J. Spiegelhalter and C.C. Taylor (Eds.), 1994. Machine Learning, Neural and Statistical Classification. Ellis Horwood.
46. Young, S., G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev and P. Woodland, 2002. The HTK Book Version 3.2, Cambridge, England, Cambridge University.