CHAPTER 4

PROPOSED IMPROVED KERNEL COMMON VECTOR (IKCV) METHOD

4.1 INTRODUCTION

From the literature survey, it is known that Kernel Common Vector (KCV) method is a non linear subspace classifier for pattern recognition. The recognition performance of KCV method is superior to other kernel based methods for applications like image segmentation, Fourier coefficients, pixel average database and the wine data base. In these applications the number of samples in each class is much larger than the dimensionality of the sample space i.e., LSS case. Moreover, this method gives better recognition rate in the above mentioned applications. This method of classifier is also suitable for SSS case. However, the performance of KCV method is reduced due to the limitations which are mentioned in section 1.4. These limitations can be solved by introducing new scatter matrix in KCV method. Moreover if there is any mislabel in the training set this method fails to classify a given test sample and acts as a weak classifier. These problems will be solved by the proposed Improved Kernel Common Vector (IKCV) method. This chapter describes about the procedure of developing IKCV method.

4.2 IMPROVED KERNEL COMMON VECTOR METHOD (IKCV)

4.2.1 Feature Extraction Procedure

The KCV method followed, the class scatter matrix and total scatter matrix for pattern recognition tasks. Since the classes are modeled as separate subspaces a class subspace may interfere with other class’ subspaces. Thus the consideration of between class scatter matrix is required. This can be solved by introducing a between class scatter matrix in this method.
The between-class scatter matrix is defined as the consideration of scatter
discrimination of classes in the feature space F. The pair wise scatter discrimination
described in section 2.3.1 is used to calculate the scatter information between the
classes. The between class scatter matrix in equation (2.2) is considered to solve the
limitations of KCV method. The introduced between class scatter matrix is defined as,

$$S^o_{n} = \sum_{i,j=1}^{C} (pd)_{i,j}^o N_i N_j (\mu_i^o - \mu_j^o) (\mu_i^o - \mu_j^o)^T.$$  \hspace{1cm} (4.1)

This expression considers the pair wise class discriminatory information between
class i with other classes which in turn considers the overlapping between class i
and other classes also. The between class scatter expression, $S^o_{n}$ in equation (4.1)
just sums up the pair wise scatter $(\mu_i^o - \mu_j^o)(\mu_i^o - \mu_j^o)^T$. The contribution from
class pair $(m,n)$ can compete with the contribution from class pair $(i,j)$. If
$(\mu_m^o - \mu_n^o)$ is much smaller compared with $(\mu_i^o - \mu_j^o)$, then classes m and n are close
to each other. Moreover, the mean difference of class’s m and n $(\mu_m^o - \mu_n^o)$ is
negligible. Thus there may be loss of discriminatory information between classes m
and n. This may affect the recognition performance. This will be avoided by
showing variations on the pair wise scatter parameter $(pd)_{i,j}^o$ such that the classes
that are close to each other are more heavily weighted than the classes far from each
other. Thus the between class scatter expression sums up the different weight values
based on the distance between the classes. By including this weighted pair wise
scatter parameter in (4.1) the between class scatter matrix is defined as

$$S^o_{y} = \sum_{i,j=1}^{C} w(pd)_{i,j}^o N_i N_j (\mu_i^o - \mu_j^o) (\mu_i^o - \mu_j^o)^T$$ \hspace{1cm} (4.2)

In this the $w(pd)_{i,j}^o$ is defined as $w(pd)_{i,j}^o \propto \frac{1}{(\mu_i^o - \mu_j^o)}$  \hspace{1cm} (4.3)
Thus using this, the classes which are closer are heavily weighted than the classes far from each other. Next, the procedure to extract the common vector of each class is described.

Let the training set be composed of C classes, where the \(i^{th}\) class contains \(N_i\) samples, and let \(x_k^i\) be a \(d\)-dimensional column vector, which denotes the \(k^{th}\) sample from \(i^{th}\) class. Thus there are total of \(N = \sum_{i=1}^{C} N_i\) samples in the training set.

In kernel methods, the samples are transformed into an implicit higher dimensional feature space \(F\) by a nonlinear mapping function \(\phi(x)\). The training set samples are mapped into a higher dimensional feature space as \(\phi : R^d \rightarrow F\). In implementation, the implicit feature vector in \(F\) does not need to be computed explicitly, and it is done by computing the inner product of two vectors in \(F\) with a kernel function,

\[
K(x, y) = \langle \phi(x), \phi(y) \rangle
\]  

(4.4)

In this research work, the polynomial kernel function is adopted as \(K(x, y) = (x^T y + 1)^d\) with the degree \(d = 2\).

Let \(X_\phi = \{\phi(x_1^1), \cdots, \phi(x_{N_i}^1), \phi(x_1^2), \cdots, \phi(x_{N_i}^2), \cdots, \phi(x_1^C), \cdots, \phi(x_{N_i}^C)\}\) represent the sample matrix whose columns are the transformed training samples in \(F\). The scatter matrix \(S_i^\phi\) of each class, the proposed between class scatter matrix \(S_B^\phi\) and the total scatter matrix \(S_r^\phi\) in \(F\) are defined as

\[
S_i^\phi = \sum_{k=1}^{N_i} (\phi(x_k^i) - \mu_i^\phi)(\phi(x_k^i) - \mu_i^\phi)^T
\]  

(4.5)

\[
S_B^\phi = \sum_{i,j} w(pd)_{i,j} N_i N_j (\mu_i^\phi - \mu_j^\phi)(\mu_i^\phi - \mu_j^\phi)^T
\]  

(4.6)

and

\[
S_r^\phi = S_i^\phi + S_B^\phi
\]  

(4.7)
where $\mu_i^\Phi$ is the mean of mapped samples in the $i^{th}$ class, and $\mu^\Phi$ is the mean of all mapped samples. It is proved that the optimal discriminant subspace of a class is the intersection of the range space of the total scatter matrix and the null space of class scatter matrix. Thus a new class subspace is defined as the basis vectors of the intersection subspaces $N(S_i^\Phi) \cap R(S_i^\Phi), i = 1, \ldots, C$ where $N(S_i^\Phi)$ is the null space of $i^{th}$ class and $R(S_i^\Phi)$ is the range space of total scatter matrix in the mapped space.

To find these basis vectors, transform all training samples onto $R(S_i^\Phi)$ and then find the vectors spanning the null spaces of the scatter matrix of each class in the transformed space. In this, transformation of all training samples onto $R(S_i^\Phi)$ will be implemented by employing the difference subspace of the total scatter matrix since the eigen values of $S_i^\Phi$ need not be computed explicitly. The algorithm to extract the common features of classes is described as follows.

**Step 1**: Construct the difference subspace $B_i^\Phi$ of class $i$ as

$$B_i^\Phi = \text{span}\{\phi(b_i^1), \ldots, \phi(b_{N-1}^i)\} \text{, where } \phi(b_i^k) = \phi(x_i^k) - \phi(x_i^1), k = 1, \ldots, N_i$$

Then the complete difference subspace $B^\Phi$ is expressed as $B^\Phi = B_1^\Phi + \cdots + B_C^\Phi$ which is equal to $\text{span}\{\phi(b_1^1), \ldots, \phi(b_{N-1}^1), \phi(b_1^2), \ldots, \phi(b_{N-1}^2), \ldots, \phi(b_1^C), \ldots, \phi(b_{N-1}^C)\}$. (4.8)

Thus, in transformed feature space $F$, the vectors which form the complete difference subspace $B^\Phi$ are independent. The orthonormal basis vectors of the subspace $B^\Phi$ are obtained using Cholesky decomposition method which is described in Appendix II. Suppose all orthonormal basis vectors of the subspace $B^\Phi$ form a matrix $Q_B$, then the complete difference subspace $B^\Phi$ is defined as

$$B^\Phi = Q_B R_B.$$ (4.9)
The orthonormal basis vectors of $B_\phi$ is obtained as

$$B_\phi R_\beta^{-1} = Q_\beta$$  \hspace{1cm} (4.10)

where $R_\beta$ is an upper triangular matrix which is obtained by Cholesky decomposition of $K_\beta$. The kernel matrix $K_\beta$ is defined as

$$K_\beta = B_\phi^T B_\phi = R_\beta^T Q_\beta^T Q_\beta R_\beta = R_\beta^T R_\beta$$  \hspace{1cm} (4.11)

$R_\beta$ is used to obtain the orthonormal basis vectors $Q_\beta$ in equation (4.9).

**Step 2:** Compute the scatter matrix of each class in the difference subspace $B_\phi$.

The new scatter matrix $S_\phi^\diamond$ of each class in the difference subspace is computed by using the kernel matrices $K^{(i)}$ and $K^{(ij)}$. The kernel matrix $K \in \mathbb{R}^{N \times N}$ is given by $K = X_\phi^T X_\phi = (K^{(i)})_{i=1,\ldots,C}$. Similarly, the matrix $K^{(ij)} \in \mathbb{R}^{N \times N_j}$ is given by $K^{(ij)} = X_\phi^T X_\phi^{(ij)} = (K^{(ij)})_{j=1,\ldots,C}$ and each sub matrix $K^{(ij)}$ is defined as

$$K^{(i)} = \langle \phi(x^\diamond_i), \phi(x^\diamond_j) \rangle$$

**Step 3:** For each class, the null space of new class scatter matrix $N(S_\phi^\diamond)$ is obtained by an eigen decomposition of $S_\phi^\diamond$. The normalized eigen vectors corresponding to the zero eigen values of $S_\phi^\diamond$ form an orthonormal basis for the null space of $S_\phi^\diamond$. Let $V^{(i)}$ be a matrix whose columns are the computed eigenvectors corresponding to the zero eigenvalues, such that

$$V^{(i)^T} S_\phi^\diamond V^{(i)} = 0, i = 1, \ldots, C$$  \hspace{1cm} (4.12)

**Step 4:** The matrix of basis vectors $W^i$ whose columns span the optimal intersection subspace of each class. It is obtained by using the column vectors $V^{(i)}$ in the difference subspace $B_\phi$ as,

$$S_\gamma^\diamond = S_\gamma^\phi + S_\beta^\phi$$  \hspace{1cm} (4.13)
Step 5: Obtain the common vector $CV$ of each class by projecting any sample of a class on to the matrix of basis vectors $W^i$ as

$$CV_{con}^i = W^{(i)^T} \phi(x_k^i) \text{ where } i = 1, \cdots, C \text{ and } k = 1, \cdots, N_i.$$  

(4.14)

The common vector is independent of the sample index $k$, and hence, any sample can be selected from a particular class to obtain the corresponding common vector. Thus by applying the above 5 steps on the training set, the common vector of each class $CV_{con}^i$ is obtained. The common vector of a class represents the common features of all samples of a class.

When all the samples in the training set are assigned to correct labels, the given test image is recognized correctly. The recognition procedure of given test image in this case is as follows.

Step 6: The feature vector $CV_{test}^i$ of a given test sample is obtained by projecting the test sample on to the matrix of basis vectors $W^i$ as

$$CV_{test}^i = W^{(i)^T} \phi(x_{test})$$  

(4.15)

Step 7: The test image $x_{test}$ is assigned to one of the classes by finding the minimum Euclidean distance between $CV_{con}^i$ and $CV_{test}^i$. And assign the test sample to the class that minimizes this distance, i.e,

$$C_i = \min_{j \in \{1, \cdots, C\}} \|CV_{test}^i - CV_{con}^j\|_2,$$

(4.16)

The above mentioned procedure works well in the case SSS. However, this procedure can be applied when the number of samples in the training set is larger than the dimension of the sample space i.e., in LSS case. In LSS case, the common vector of each class is obtained by projecting the average vector of a class onto the matrix of basis vectors $W^i$ as,
\[ CV_{con}^i = W^{(i)} (\phi(x_{ave}) = W^{(i)} \left( \frac{1}{N_i} \sum_{k=1}^{N_i} \phi(x_k^i) \right), \text{where} \quad i = 1, \ldots, C \quad \text{and} \quad k = 1, \ldots, N_i \]  

(4.17)

The obtained results of this procedure on face recognition task are shown in Chapter 5. This procedure increases the recognition rate due to the following appealing properties.

- This method employs the nonlinear subspaces for representing each class, since the mapped space is nonlinearily related to the original sample space.

- Transformation of training samples onto \( R(S^S_\gamma) \) by following the difference subspace of the total scatter matrix reduces the computational complexity.

- By extracting the common vector of each class from the intersection of the range space of the total scatter matrix and the null space of each individual class scatter matrix increases the discriminative information and numerical stability.

- In this method, there is no limitation on the dimensionality of the sample space and the training set size i.e. this method can be applied for SSS case and for LSS case also.

### 4.2.2 IKCV Procedure

In the previous section the discriminative features of each class is extracted by following the assumption that the training set samples are assigned to correct labels. If there is any mislabel in the training set, the above procedure fails to extract the correct discriminative features of classes. Then, this procedure acts like a weak classifier in this case. To overcome this problem and to make this weak classifier as strong classifier, boosting technique is employed.
Boosting is a general machine learning meta algorithm for improving the accuracy of any given learning algorithm. One of the effective boosting algorithms, referred to as AdaBoost, can be used for multi-class problems. The multi-class extensions of AdaBoost algorithm are AdaBoost.M1 and AdaBoost.M2. For multi-class problems AdaBoost.M2 outperforms AdaBoost.M1. AdaBoost.M2 focus the learner not only on hard to classify samples but on incorrect labels. This is the reason for selecting the AdaBoost.M2 algorithm in this research work to construct a strong classifier. This strong classifier combines the strength of the AdaBoost algorithm and IKCV techniques.

In this method, pair wise class discriminatory information is calculated based on the mislabel distribution of samples in the training set. In this, the mislabel distribution \( \Gamma'(\ast, \ast) \) measures the extent of difficulty of discriminating the sample \( \ast \) from the improper label \( \ast \) on the basis of previous boosting results. By this, at the \( t \)th iteration in AdaBoost.M2 algorithm any pair wise class discriminant distribution \( ((pd)_{i,j}^{\ast \ast})' \) between classes \( C_i \) and \( C_j \) can be calculated as follows:

\[
((pd)_{i,j}^{\ast \ast})' = \begin{cases} 
\frac{1}{2} \left( \sum_{i=1}^{N_i} \Gamma'(x_i^j, j) + \sum_{i=1}^{N_j} \Gamma'(x_i^i, i) \right) & \text{if} \quad i \neq j \\
0 & \text{otherwise}
\end{cases} \tag{4.18}
\]

A larger value of \( ((pd)_{i,j}^{\ast \ast})' \) indicates the worse separability between class \( i \) and class \( j \) and it also show that the class \( i \) and \( j \) are closer together in \( F \) i.e., the classes which are closer have larger value of \( ((pd)_{i,j}^{\ast \ast})' \) at \( t \)th iteration. From the above concept, it is clear that the pair wise class scatter parameter \( ((pd)_{i,j}^{\ast \ast}) \) acts as weight value i.e., the classes which are closer should be more heavily weighted than the classes far from each other.
In AdaBoost.M2 algorithm, some sampling procedures are employed to artificially weaken the corresponding discriminant technique. This can be introduced by selecting the hardest samples from each class based on the parameter $q^t_{it_k}$. These hardest samples of a class vary from iteration to iteration. The parameter $q^t_{it_k}$ measures the difficulty of separating a $k^{th}$ sample of class $i$ from other classes at $t^{th}$ iteration and can be calculated by using

$$ q^t_{i \, i_k} = \left( \sum_{j \neq i} \Gamma^t(x^t_{i_k}, j) \right) $$

(4.19)

where $k = 1, \cdots, N_i$ and $i, j = 1, \cdots, C$. A larger value of $q^t_{it_k}$ shows the worse separability of $k^{th}$ sample of class $i$ from other classes at $t^{th}$ iteration. The harder samples in a class are selected by using the sample separability parameter $q^t_{it_k}$. This sample separability parameter $q^t_{it_k}$ is incorporated in within class scatter expression to enhance the characteristics of $S^\circ_i$. By this, the parameter $q^t_{it_k}$ emphasizes the harder samples in class $C_i$. Thus the new within class scatter matrix is defined as

$$ S^\circ_i = \sum_{j=1}^{C} \sum_{k=1}^{N_i} q^t_{it_k} (\phi(x^t_{i_k}) - \mu^\circ_i)(\phi(x^t_{i_k}) - \mu^\circ_i)^t $$

(4.20)

### IKCV Procedure

**Input**

(iii) A set of training examples $X^\circ$

$$ X^\circ = \left\{ \phi(x^t_{i_k}) \right\}_{x^t_{i_k} \in \mathbb{R}^d, i = 1 \cdots C, k = 1 \cdots N_i} $$

(iv) A set of all mislabels $M$

$$ M = \{(i, \phi(x^t_{i_k})), j, i, j \in \{1, \cdots, C\}, k \in \{1, \cdots, N_i\}, i \neq j\} $$

(iii) The initial Mislabeled Distribution on $M$ is,

$$ \Gamma^t(x^t_{i_k}, j) = \frac{1}{N(C-1)}, \text{ a small constant } \varepsilon. $$
Procedure

For $t = 1$ to $T_{\text{max}}$ do

(i) Calculate the boosting parameters, the pair wise class discriminatory information $((pd)^{(i)})'$, and the sample separability parameter $q_{si}$ by using equations (4.18) and (4.19) respectively.

(ii) Select $S$ hardest samples from each class based on $q_{si}$ to form a training subset $T_s \subset X_\phi$.

(iii) Apply feature extraction procedure described in section 4.2.1 (step 1 to step 5) on $T_s$ and constitute the IKCV-based feature extraction technique, denoted by $IKCV_i$.

(iv) Apply $IKCV_i$ on $X_\phi$ to obtain $\{CV_i\}_i \in [1, \cdots, C]$ where $\{CV_i\}$ represents the common vector of all the classes at $i^{th}$ iteration.

(v) Build the hypotheses $h_i(x, \bullet) \in [0,1]$ on the subset $CV_i'$ of $\{CV_i\}_i \in [1, \cdots, C]$ corresponding to $T_s$.

(vi) Calculate the pseudo loss $\omega'$ based on $h_i$ as

$$\omega' = \sum_{(i,k), (j,l) \in M} \frac{\Gamma'(\phi(x_k^i), j)(1 + h_i(CV_i', j) - h_i(CV_i', i))}{2}$$

(vii) Set $\beta_i = \omega_i (1 - \omega_i)$ and if $\beta_i \leq \epsilon$ then $T_{\text{max}} = t - 1$ and break.

(viii) Update the mislabel distribution $\Gamma'$:

$$\Gamma^{t+1}(\phi(x_k^i), j) = \Gamma'(\phi(x_k^i), j) \beta_i^{(1 + h_i(CV_i', j) - h_i(CV_i', i))}/2$$

end For
Output

The final hypothesis \( h_f(\phi(x_{test})) \) is:

\[
h_f(\phi(x_{test})) = \arg \max_{i \in \{1, \ldots, c\}} \left\{ \sum_{t=1}^{T} \left( \log \frac{1}{\beta_t} \right) h_t(CV', i) \right\}
\]

For a given test sample \( x_{test} \), \( CV' \) is the corresponding kernel common vector extracted by IKCV and maximum response of class \( i \) is the class label for a given test sample \( x_{test} \).

In this method, the feature extraction procedure described in section 4.2.1 is involved in the IKCV algorithm to calculate the optimal common vectors of classes in the transformed space. The detailed calculation procedure is described in section 4.2.1. In addition, this procedure is a strong feature extraction technique for classification. As a result, the boosting process cannot go forward due to the very small pseudo-loss \( \omega_r \). In general, some sampling procedures are employed to artificially weaken the corresponding discriminant technique, and, in IKCV some samples in each class are selected based on \( q'_{lj} \) to focus on the hardest samples in each class. In order to be consistent with the AdaBoost algorithm, the hypothesis \( h_t(\star, \star) \) between sample \( \star \) and class label \( \star \) is built easily based on the Euclidean distance method. The experimental results and its analysis are discussed in Chapter 5.