Chapter 8

A Feature Partitioning Approach to Subspace Classification

8.1 Introduction

Note: An initial version of the work in this chapter has been published in proceedings of IEEE TenCon 2007 International Conference.

In this Chapter, we explore the applicability of feature partitioning based PCA (FP-PCA) methods such as SubXPCA (Chapter 4), SubPCA (Section 2.2 of Chapter 2) for subspace classification. Subspace classification is one of the widely used methods for pattern recognition tasks, where a linear subspace of the Euclidean sample space is found [115]. The motivation for subspace classifiers originates from compression and optimal reconstruction of multidimensional data with linear principal

components. The use of linear subspaces as class models is based on the assumption that the vector distribution in each class lies approximately on a lower-dimensional subspace of the feature space. The subspaces representing classes are defined in terms of basis vectors that are linear combinations of the sample vectors of each class. Once the basis vectors spanning those subspaces are computed, a test data vector from an unknown class is classified based on the lengths of the projections of that sample onto each of the subspaces or, alternatively, on the distances of the test vector from these subspaces. Even though this linearity assumption may not be valid in all the cases, acceptable classification accuracies can be achieved if the input vector dimensionality is large enough [93].

Subspace methods in data analysis date back to the 1930s by Hotelling [64]. The value of the subspace methods in data compression and optimal reproduction was observed in the 1950s by Kramer and Mathews [89]. Later, Watanabe et al [172] published the first application in pattern classification. Learning subspace methods gained popularity after the pioneering work of Kohonen et al [87] in 1970s. These methods aimed for classification instead of optimal compression or reproduction since their inception. The guiding idea in the learning methods is to modify the bases of the subspaces in order to diminish the number of misclassifications. The nature of the modifications varies in different learning algorithms.

The remainder of the Chapter is organized as follows. In section 8.2 we review some classical PCA based subspace classification methods. We propose a novel Sub-XPCA based Feature Partitioning approach to Subspace Classification (FP-SC) in section 8.3. Time complexity analysis of subspace methods is done in section 8.4. We
demonstrate our approach by using experimentation on UCI repository of Machine Learning data sets in section 8.5.

8.2 Review of Classical Subspace Methods

In this section, we discuss some of classical PCA based subspace methods in brief. A useful review of subspace classification may be found in [93].

8.2.1 Class-Featuring Information Compression (CLAFIC)

Watanabe et al applied Principal Component Analysis (PCA), or the Karhunen-Loeve Transform (KLT), in classification which is known as CLAFIC algorithm [172]. CLAFIC simply forms the base vectors for the classifier subspaces from the eigenvectors of the covariance matrix or correlation matrix of each class. For each class \( h_q \), we compute the covariance matrix \( C_q = E[\mathbf{x}\mathbf{x}^T | \mathbf{x} \in h_q] \). Then we find first \( r \) eigenvectors of \( C_q \), \( \{e_1^q, e_2^q, \ldots, e_r^q\} \), in the order of decreasing eigenvalues \( \lambda_i^q \), \( i = 1, 2, \ldots, r \) and used as columns of the basis eigenvector matrix \( E_q \), which is given by

\[
E_q = \{e_i^q | (C_q \cdot e_i^q = \lambda_i^q \cdot e_i^q; \quad \lambda_i^q \geq \lambda_{i+1}^q, i = 1, 2, \ldots, r\} \tag{8.1}
\]

A test vector \( \mathbf{L} \) is classified according to the maximal similarity value using the function

\[
B(\mathbf{L}) = \arg \max_{q=1, \ldots, c} \| [E_q]^T \cdot \mathbf{L} \|^2 \tag{8.2}
\]
8.2.2 Multiple Similarity Method (MSM)

One way to generalize the classification function (8.2) is to introduce individual weights for all the basis vectors. Iijima et al [69] have selected to weight each basis vector with the corresponding eigenvalue in their Multiple Similarity Method (MSM).

\[
D(L) = \arg\max_{q=1, \ldots, c} \sum_{i=1}^{r^q} \frac{\lambda_q^i}{\lambda_1^i} (L^T e_q^i)^2 \tag{8.3}
\]

This emphasizes the effect of the most prominent directions, for which \(\frac{\lambda_q^i}{\lambda_1^i} \approx 1\). The selection of the subspace dimension \(r^q\) is, therefore, less important because the influence of the less prominent eigenvectors, which have multipliers \(\frac{\lambda_q^i}{\lambda_1^i} \approx 0\), becomes trivial. Thus the influence of the eigenvectors which have small eigenvalues and are created by additive noise is thus cancelled out.

One of the problems with classical PCA based subspace methods is large computational complexity especially for high-dimensional data because of high computational time to calculate covariance matrix. The covariance matrix is subsequently used to compute eigenvectors and eigenvalues. Another problem is the classical PCA based subspace methods may not yield good classification especially if local variations are dominant (i.e. variations restricted to a subset of original features). In the next section we propose a promising novel approach to reduce time complexity as well as to improve classification rate, which is based on feature partitioning framework. For information on feature partitioning framework and approaches please see Chapters 3-5.
8.3 Feature Partitioning (SubXPCA based) Approach to Subspace Classification (FP-SC)

In this section, we present our proposed approach to subspace classification based on feature partitioning framework.

8.3.1 FP-SC Algorithm

Consider $X = \{ (X_1)^1, (X_2)^1, \ldots, (X_{N_1})^1, \ldots, (X_1)^c, (X_2)^c, \ldots, (X_{N_c})^c \}$, the set of $N = N_1 + N_2 + \ldots + N_c$ patterns of size $d$. Here, $(X_i)^q$ denotes $i^{th}$ pattern of class $h_q$, $N_q$ indicates the number of data items which belongs to class $h_q$, $q \in \{1, 2, \ldots, c\}$ and $c$ is the number of classes.

Step 1: Computing subspace for each class, $h_q$, using SubXPCA (See Chapter 4 for SubXPCA approach)

(A) Partitioning:
For each class $h_q, q \in \{1, 2, \ldots, c\}$, we divide each pattern, $(X_i)^q$ into $k (\geq 2)$ equally-sized sub-patterns, $\{(X_1)^q, (X_2)^q, \ldots, (X_k)^q\}$. Each sub-pattern is of size $u$, where $u = \lfloor \frac{d}{k} \rfloor$ and let $P_{qj}^q$ be the set of $j^{th}$ sub-patterns of $\{(X_i)^q\}; i = 1, 2, \ldots, N_q$ and is given by

$$ (P_{qj}^q)_{N_q \times u} = [(X_1)^q (X_2)^q \ldots (X_{N_q})^q]^T \quad (8.4) $$

(B) Local feature extraction:
For each sub-pattern set, $P_{qj}^q, j \in \{1, 2, \ldots, k\}$: (i) Compute the sub-covariance matrix $(C_{j})_{u \times u}$, then (ii) Choose $r (\leq u)$ local column eigenvectors, $(E_{j})_{u \times r}$, corresponding to
highest eigenvalues computed from the sub-covariance matrix, \( C^q_j \). (iii) Subsequently extract \( r \) local features from \( P^q_j \) by projecting \( P^q_j \) onto \( E^q_j \) as given by

\[
(R^q_j)_{N_q \times r} = (P^q_j)_{N_q \times u} (E^q_j)_{u \times r} = [(Y^q_1)^q \quad (Y^q_2)^q \quad \ldots \quad (Y^q_N)^q]^T
\]

\( (Y^q_j)_{r \times 1} \) is the set of extracted \( r \) local features corresponding to \( (X^q_j)_{u \times 1} \) from \( P^q_j \) for class \( h_q \).

(C) Global feature extraction:

(i) Concatenate \( r \) local features (extracted in the preceding step) corresponding to the same pattern \( (X^q_i) \) as given by

\[
(Y^q_i)_{k \times 1} = [(Y^q_1)^T \quad (Y^q_2)^T \quad \ldots \quad (Y^q_k)^T]^T
\]

(ii) Compute final covariance matrix \( (C^g)^q_{k \times r} \) from locally-reduced patterns, \( (Y^q_1) \), \( (Y^q_2) \), \ldots , \( (Y^q_N) \), then (iii) Compute \( w \) (\( \leq k \)) global eigenvectors, \( (E^g)^q_{k \times w} \), corresponding to \( w \) highest eigenvalues.

**Step 2: Classification.** For a test pattern, \( L \) of \( d \) features : (i) Divide \( L \) into \( k \) (\( \geq 2 \)) equally-sized sub-patterns as done in Step-1. Each sub-pattern, \( L_j \) \((j = 1, 2, \ldots , k)\) is of size \( u \), where \( u = \left\lfloor \frac{d}{k} \right\rfloor \). \( L_1 \) contains first \( u \) features of \( L \), \( L_2 \) contains next \( u \) features and so on. (ii) Project each \( (L_j)_{u \times 1} \); \( j = 1, 2, \ldots , k \) onto \( (E^q_j)_{u \times r} \) of class \( h_q \) to get \( k \) \( r \) local principal component features, denoted by \( (L^q_j)_{k \times r \times 1} \), as described in the previous step. (iii) Subsequently we classify the test pattern, \( L \) using the following function.

\[
F(L) = \arg\max_{q=1,2,\ldots ,c} ||(E^q)^T \cdot L^q||^2
\]

where \( (E^g)^q_{k \times r \times w} \) is the set of \( w \) column global eigenvectors for class \( h_q \) obtained in the previous step.
8.4 Time Complexities of Classical and Feature Partitioning based Subspace Classification Methods

We categorize subspace methods such as CLAFIC, MSM, etc as Classical PCA based subspace methods. In PCA based subspace methods, the computation of covariance matrices consumes a large amount of computational time for high-dimensional data in particular, and a relatively insignificant amount of time for other tasks such as finding eigenvalues, etc. Hence, here we focus our study on time complexity of covariance matrices as being computed by classical PCA based subspace methods and FP-SC.

From Chapter 4, we know that the time complexity to calculate a $d \times d$ covariance matrix by classical PCA based subspace methods for a class $h_q$, $T^q_C$, is given as

$$T^q_C = O(N_q.d^2 + d^3)$$ (8.8)

and the time complexity to calculate all covariance matrices by FP-SC for a class $h_q$, $T^q_F$, is given by

$$T^q_F = O(k.N_q.u^2 + k.u^3 + N_q.k^2.r^2 + k^3.r^3)$$ (8.9)

The total time complexity of a subspace method is the sum of time complexities with respect to all the classes.

**Theorem 32** For a class with a label $h_q$, $T^q_F < T^q_C$, $\forall r < u.\sqrt{\frac{(k-1)}{k}}$, where $2 \leq k \leq \frac{d}{2}$, is the number of sub-patterns per pattern, $r$ is the number of chosen projection (eigen) vectors (PVs) per sub-pattern set and $u$ is the sub-pattern size.
Proof 32 The Theorem directly follows on the similar lines of Theorem 3 of Chapter 4.

Theorem 33 \( \lim_{r \to 1, k \to 2} [T_F^q \approx (\frac{1}{k}).T_C^q], \) where \( 2 \leq k \leq \frac{d}{2} \), is the number of sub-patterns per pattern and \( r \) is the number of chosen projection (eigen)vectors (PVs) per sub-pattern set.

Proof 33 The Theorem directly follows on the similar lines of Theorem 4 of Chapter 4.

By Theorem 33, \( T_F^q \approx (\frac{1}{k}).T_C^q \) is true for smaller values of \( k \) and \( r \). However, in practice, \( r \) may not be chosen as 1 (i.e. smallest possible value), especially when \( k \) is small, since the classification rate may get reduced due to less number of eigenvectors \( (r) \). Hence some trade-off between \( r \) and \( k \) is required to achieve good classification rate and time efficiency.

8.5 Experimental Results and Discussion

In this section, we compare FP-SC method (i.e. SubXPCA based subspace classifier) with PCA based and SubPCA based subspace classifiers. SubPCA [21] is another FP-PCA method and the description on SubPCA method may be found in section 2.2 of Chapter 2.

8.5.1 UCI Data Sets

We considered 2 publicly available databases from UCI repository of Machine Learning [165] for our experiments. (1) Waveform data (21 features, 3 classes with
labels (0,1,2), 5000 Patterns, 50 patterns each class, for training, rest of them for testing). (2) Musk data (166 features, 2 classes with labels (0,1), 6598 patterns, 500 patterns each class, for training, rest of them for testing).

8.5.2 Experimental Setup

For each class, an experiment is conducted as follows: We choose the number of sub-patterns, $k$, to minimize the truncation of last features as far as possible. For each class, $h_q$, $w$ projection eigenvectors are found using FP-SC algorithm (Section 8.3.1). For SubPCA based subspace classification, Steps-1(C)(ii)-(iii) are omitted. Each test data is classified into a class for which the norm of projection of the test data item is maximum. The classification is done based on subspace classification rule as given in eq. (8.7) by FP-SC method. For SubPCA based subspace classification, we use $\| L^q \|^2$ instead of $\| [(E^q)^q]^T L^q \|^2$ in eq. (8.7) of Step-2(iii). The experiment is repeated 10 times, each iteration with different training and testing data sets keeping the same values of $k$ and $w$. Further (a) the average of these 10 classification rates and (b) best of these 10 classification rates are calculated for the given values of $k$ and $w$. Now repeat the procedure by varying $k$ and $w$.

For different data sets, the number of blocks, $k$ is varied as follows: For musk data, we consider $k = 2, 3, 5, 11, 15, 33$ and for waveform data, we consider $k = 2, 3, 4, 5, 7$. In the case of PCA based subspace classification, $k = 1$ for all the data sets.

We plot the results as follows: (i) Among the experiments with varying number of sub-patterns (blocks) we choose the case (i.e. $k = 11$ for SubPCA and SubXPCA with respect to musk data; $k = 2$ for SubPCA and $k = 3$ for SubXPCA with respect
to waveform data) with relative good *average classification performance*. Here we take the average of results of 10 iterations, each iteration with different training and testing data. The average classification results and the corresponding total computational time for 10 iterations, thus obtained are plotted in Figs. 8.1-8.2 for musk data and in Figs. 8.8-8.9 for waveform data.

(ii) Among the experiments with varying number of sub-patterns (blocks) we choose the case (i.e. $k = 5$ for SubPCA and $k = 11$ for SubXPCA with respect to musk data; $k = 2$ for SubPCA and $k = 5$ for SubXPCA with respect to waveform data) with relative *best classification performance*. Here we take the best of classification results of 10 iterations, each iteration with different training and testing data. The best classification results thus obtained are plotted in Fig. 8.3 for musk data and in Fig. 8.10 for waveform data.

The comparison of PCA, SubPCA and SubXPCA based subspace classifiers with respect to both average classification rate and computational time is shown in Fig. 8.4 for musk data.

(iii) The classification performance by *varying the number of sub-patterns* ($k$) is shown in Figs. 8.5, 8.6, 8.11 and 8.12. For each $k$ (number of sub-patterns): (a) we find maximum average classification rate (of average classification rates obtained with varied number of projection eigenvectors) and is plotted in Fig. 8.5 for musk data and in Fig. 8.11 for waveform data, (b) we find maximum best (of 10 iterations) classification rate (of best classification rates obtained with varied number of projection eigenvectors) and is plotted in Fig. 8.6 for musk data and in Fig. 8.12 for waveform data.
(iv) The comparison of total computational time of 10 iterations for PCA and Sub-PCA (with different $k$ values) and SubXPCA (with different $k$ values) based subspace classification is shown in Fig. 8.7 for musk data and in Fig. 8.13 for waveform data.

We used Pentium 4 based system with 2.4 GHz CPU clock speed, 256 MB RAM and Fedora Core 5 Linux running on it, to obtain experimental results. We used C language built-in time functions for recording time and procedures, viz. `tqli`, `tredt`, `eigensrt`, to find eigenvectors, eigenvalues and for sorting them from [127].

### 8.5.3 Discussion of Experimental Results

For UCI musk data:

From Fig. 8.1, it is clear that SubXPCA based subspace classifier (FP-SC) outperforms both PCA based and SubPCA-based subspace classifiers. It is observed that SubXPCA based method (FP-SC) shows (i) 7% higher classification rate as compared to PCA based classifier and (ii) 2.5% higher classification rate as compared to SubPCA based classifier. Here SubXPCA based method uses 1 eigenvector from each of sub-pattern sets. Fig. 8.2 shows the computational efficiency of SubXPCA based feature partitioning subspace classifier as compared to PCA and SubPCA based subspace classifiers. A novel plot between average classification rate and computational time (Fig. 8.4) allows one to conclude as to the method that gives good classification at less computational requirements. Such method forms a cluster of its points at top-left corner of the plot. Interestingly, SubXPCA based subspace classifier (FP-SC) forms such a cluster at top-left corner, which shows its superiority over other
methods in terms of classification at less computational time. Please note that other two methods (SubPCA and PCA based subspace classifiers) have their points moved away from top-left corner (except one point of SubPCA), which implies that those methods either show lower classification rate or more computational time or both.

Another way to analyze the subspace methods is in terms of best individual classification rate. From Fig. 8.3, it is seen that SubXPCA based FP-SC method outperforms both PCA and SubPCA based subspace methods. That is, SubXPCA based method (FP-SC) shows (i) 12.4% higher classification rate than PCA based subspace method and (ii) 8.4% higher classification rate than SubPCA based subspace method.

Another interesting analysis is to see the performance of these methods with varying number of sub-patterns or blocks ($k$). Figs. 8.5 and 8.6 reveal that SubXPCA based subspace classifier (FP-SC) consistently shows better classification as compared to SubPCA based subspace method with varying number of blocks. Note that SubPCA based subspace method shows decreasing performance with increased number of sub-patterns or blocks because more noisy features are added up with increased number of blocks. SubXPCA based method (FP-SC) is able to remove such noisy features effectively by using inter-block correlations or dependencies among these local features. It is also observed that FP-SC shows superior classification as compared to PCA based subspace classifier.

Finally we compare the computational time of all these methods with respect to varying number of sub-patterns or blocks as shown in Fig. 8.7. From the Fig. 8.7 it is clear that both feature partitioning based subspace classifiers (SubPCA and SubXPCA based) show decreasing computational time with increased number of blocks.
However, it is to be noted that PCA shows high computational requirements as compared to feature partitioning based methods (SubXPCA and SubPCA based). Also SubXPCA based subspace classifier shows better computational time as compared to SubPCA based method because SubXPCA is more effective in summarizing most of the variance in less number of principal components.

In a nutshell, our experimentation on UCI musk data shows that, SubXPCA based subspace classifier (FP-SC) outperforms both PCA based and SubPCA based subspace classifiers in terms of classification rate and computational time. Please note that SubPCA based subspace classifier shows better classification rate and computational time as compared to PCA based method. This is perhaps due to dominant local variations in waveform data. In this case (where local variations are dominant), SubXPCA shows much better performance as compared to other two methods.

*For UCI Waveform data:*

From Fig. 8.8 and Fig. 8.10 it is observed that SubXPCA based subspace classifier (FP-SC) shows slightly improved classification rate as compared to SubPCA and PCA based subspace classifiers with respect to maximum classification values. SubXPCA based subspace classifier shows computational superiority as compared to PCA based and SubPCA based subspace classifiers as shown in Figs. 8.9 and 8.13. Please note that PCA based method shows better classification as compared to SubPCA based method, however PCA takes more computational time as compared to SubPCA based method. With varying number of sub-patterns (blocks), SubXPCA based subspace classifier (FP-SC) shows consistently good classification rate as compared to SubPCA based subspace classifier. It is also observed that FP-SC method shows
slightly improved classification as compared to PCA based subspace classifier (Figs. 8.11 and 8.12).

In a nutshell, from our experimentation on UCI waveform data it is clear that Sub-XPCA based method (FP-SC) shows slightly improved classification rate as compared to PCA based method. SubPCA based subspace classifier shows lower performance as compared to FP-SC method with varying number of blocks. This is perhaps due to more global variations than local variations in the data. However, feature partitioning based subspace classifiers (SubXPCA and SubPCA) remain computationally superior as compared to PCA based subspace classifier.

8.6 Explaining Possible Reason Why FP-SC is better than Other Methods?

From Theorem 32 and Theorem 33, it is clear that FP-SC (SubXPCA based) method can improve its computational time ideally upto $\frac{1}{k}$ times that of classical PCA based subspace methods. From our experimentation it is well known that FP-SC (i.e. SubXPCA based subspace classifier) is more efficient in terms of classification because it considers local structure as well as global structure in computing subspace (Step-1 of algorithm in section 8.3.1). In contrast to FP-SC (SubXPCA based), (i) classical PCA based subspace methods consider only global structure to compute subspace, which may not capture local variations (i.e. variations limited to subset of original features) and (ii) other FP-PCA based subspace classifiers (e.g. SubPCA and similar methods) capture only local structure, but fail to capture global structure.
Chapter 8: A Feature Partitioning Approach to Subspace Classification

Figure 8.1: Comparison of average classification rates for UCI Musk data. SubXPCA based subspace classifier (FP-SC) outperforms both PCA based and SubPCA-based subspace classifiers. It is clear that SubXPCA based method (FP-SC) shows (i) 7% higher classification rate as compared to PCA based subspace classifier and (ii) 2.5% higher classification rate by using less number of projection eigenvectors as compared to SubPCA based subspace classifier. SubXPCA based method uses 1 eigenvector from each of sub-pattern sets.
Figure 8.2: *Comparison of computational time for UCI Musk data.* SubXPCA based subspace classifier (FP-SC) shows less computational time as compared to PCA and SubPCA based subspace classifiers.
Figure 8.3: *Comparison of best classification rates for UCI Musk data.* SubXPCA based subspace classifier (FP-SC) outperforms both PCA based and SubPCA-based subspace classifiers. It is clear that SubXPCA based method (FP-SC) shows (i) 12.4% higher classification rate as compared to PCA based subspace classifier and (ii) 8.4% higher classification rate by using less number of projection eigenvectors as compared to SubPCA based subspace classifier. SubXPCA based method uses 1 eigenvector from each of sub-pattern sets.
Figure 8.4: Comparison of PCA, SubPCA and SubXPCA based subspace classifiers with respect to both computational time and classification rate for UCI Musk data. SubXPCA based method (FP-SC) forms all its points at the top-left corner of the plot, which is the indication of high classification rate at less computational time. Other two methods have the points concentrated away from top-left corner which indicates that both PCA and SubPCA based classifiers either show lower classification rate or high computational time or both.
Figure 8.5: Comparison of average classification rates with varied number of sub-patterns (blocks) for UCI Musk data. SubXPCA based subspace classifier (FP-SC) consistently shows good performance as compared to SubPCA based subspace classifier with different number of blocks. Please note that PCA based classifier shows lower classification rate as compared to (i) FP-SC classifier (with all $k$ values) and (ii) SubPCA based classifier (except for $k = 15, 33$).
Chapter 8: A Feature Partitioning Approach to Subspace Classification

Figure 8.6: Comparison of best classification rates with varied number of sub-patterns (blocks) for UCI Musk data. SubXPCA based subspace classifier (FP-SC) consistently shows good performance as compared to SubPCA based subspace classifier with different number of blocks. Please note that PCA based classifier shows lower classification rate as compared to (i) FP-SC classifier (with all $k$ values) and (ii) SubPCA based classifier (except for $k = 15, 33$).
Figure 8.7: Comparison of computational time with different number of blocks for UCI Musk data. It is to be noted that SubXPCA based subspace classifier (FP-SC) is computationally more efficient as compared to other two methods. Also SubPCA based method shows less computational time over PCA based subspace classifier.
Figure 8.8: Comparison of average classification rates for UCI Waveform data. SubXPCA based subspace classifier shows slight improvement over PCA based method with respect to its maximum of plotted classification rates. However, SubXPCA based method shows nearly 2% higher classification as compared to SubPCA based subspace classifier with respect to its maximum of plotted classification rates. SubXPCA uses 1 and 2 projection eigen vectors (PVs) per sub-pattern set.
Figure 8.9: *Comparison of computational time for UCI Waveform data.* SubXPCA based subspace classifier (FP-SC) shows less computational time as compared to PCA and SubPCA based subspace classifiers.
Figure 8.10: *Comparison of best classification rates for UCI Waveform data.* SubXPCA based subspace classifier shows slight improvement over PCA and SubPCA based methods with respect to its maximum of plotted classification rates. SubXPCA uses 1 and 2 projection eigen vectors (PVs) per sub-pattern set.
Figure 8.11: *Comparison of average classification rates with varied number of sub-patterns (blocks) for UCI Waveform data.* SubXPCA based subspace classifier (FP-SC) consistently shows good performance as compared to SubPCA based subspace classifier with different number of blocks. SubXPCA based method shows slight improvement over PCA based subspace classifier. It is clear that SubPCA based classifier shows lower performance as compared to other two methods.
Figure 8.12: Comparison of best classification rates with varied number of sub-patterns (blocks) for UCI Waveform data. SubXPCA based subspace classifier (FP-SC) consistently shows good performance as compared to SubPCA based subspace classifier with different number of blocks. SubXPCA based method shows slight improvement over PCA based subspace classifier. It is clear that SubPCA based classifier shows lower performance as compared to other two methods.
Figure 8.13: Comparison of computational time with different number of blocks for UCI Waveform data. It is to be noted that SubXPCA based subspace classifier (FP-SC) is computationally more efficient as compared to other two methods. Also SubPCA shows less computational time over PCA based subspace classifier.
However, FP-SC (SubXPCA based) overcomes these problems, by capturing local information in addition to global structure. Therefore, SubXPCA based subspace classifier (FP-SC) shows its superiority by taking advantage of merits of both local and global PCA based feature extraction methods.

8.7 Summary

We proposed a feature partitioning approach to subspace classification, FP-SC (SubXPCA based subspace classifier). FP-SC outperforms other classical PCA based subspace methods in terms of time complexity and classification. Also, FP-SC shows superiority in terms of classification as compared to SubPCA (an existing FP-PCA method) based subspace classifier. In addition, FP-SC classifier performs well in terms of computational time as compared to SubPCA based subspace classifier if FP-SC uses less number of principal components than SubPCA based method. Classical PCA based subspace methods use classical PCA to compute subspace, which may take large amount of time if the dimensionality of the data is high. Unlike classical PCA based subspace methods, FP-SC uses feature partitioning approach to compute subspace, where it reduces the time complexity enormously and improves the classification rate as well. The proposed method may be extensively used for face recognition, palmprint recognition, OCR applications, etc.