Chapter 1

Introduction

In this chapter, we give the background and motivation of our investigation. We present a brief review of pattern recognition, feature extraction (dimensionality reduction) and Principal Component Analysis (PCA) to setup the context of this work. At the end of this Chapter we summarize our contributions that are to come in the following chapters.

1.1 A Brief View of Pattern Recognition

Pattern Recognition (PR) is the study of how machines can observe the environment, learn to distinguish patterns of interest from their background, and make sound and reasonable decisions about the categories of the patterns [72]. Given a pattern (e.g. a face image or a character or a fingerprint image), its recognition or classification may consist of one of the following two tasks [173]: (i) Supervised classification (e.g. Nearest Neighbour Rule), in which an input pattern is identified as
a member of a predefined class. (ii) Unsupervised classification (e.g. Clustering), in which an input pattern is assigned to an unknown class. In many of the applications, it is obvious that no single approach for classification is optimal and that multiple approaches have to be used. Consequently combining several approaches is one of the commonly used practices in pattern recognition.

Pattern Recognition systems [144] often play a vital role in machine intelligence systems and are used for both data processing and decision making. In broad terms, pattern recognition is the science that concerns the description or classification (recognition) of measurements (patterns). Pattern recognition systems overlap with other areas such as Signal Processing, Artificial Intelligence, Neural Modelling, Optimization/Estimation theory, etc. PR applications include image processing, segmentation and image analysis, computer vision, seismic analysis, radar signal classification, biometric identification (e.g. face, fingerprint), speech recognition or understanding, character recognition and handwriting analysis.

Pattern Recognition may be classified as an information reduction, information mapping or information labelling process. The structure of a typical PR system is shown in Fig. 1.1. The PR system consists of a sensor (e.g. camera), a feature extraction mechanism/algorith (e.g. Principal Component Analysis) and a classification or descriptive algorithm (e.g. k-Nearest Neighbour method). Using camera/transducer, we measure world pattern data, which may contain noise, measurement error, etc (for e.g. camera may capture a face of a person in a rainy day or sunny day or in overcast weather in a typical surveillance system). Further, the measured data captured by transducer at different scenarios is required to be preprocessed.
to make it suitable for pattern recognition tasks (for e.g. enhancing or cropping or segmenting the face image). The preprocessed pattern may be of high dimensionality. It is well known that high-dimensional data demands high computational and storage requirements. In addition, classification performance may come down with increase in dimensionality due to curse of dimensionality phenomenon. Therefore, there is a need of feature extraction algorithms, which reduce dimensionality and also extract a few salient features useful for better classification. Next, classification algorithms such as Nearest Neighbour method, Bayes classifier, Support Vector machines, Clustering algorithms, etc, make use of the salient features to classify a given pattern. In general, we assume some data is available which is already classified or described to train the system. Such data is called as training data.

Recognition depends upon the concept of a pattern. Generally, a pattern can be described as a set of measurements or observations and may be represented in vector or matrix form. The measurements could be items such as height, weight, color, etc. In addition, patterns may be converted from one representation to another. Some examples of patterns are shown in Fig. 1.2. In broad terms, features represent any extractable measurement (e.g. pixel intensities of a face image). Features may be symbolic, numeric or both (e.g. height, color, texture). Features may also be obtained from applying a feature extraction or feature selection algorithm (e.g. PCA or LDA) to the input data. Such algorithms may need significant computational effort and the extracted features may contain errors or noise. Features may be represented by continuous, discrete or discrete-binary variables. The key issue of feature extraction/selection algorithms is to extract or choose features that (i) are computationally
Figure 1.1: Typical pattern recognition system structure [144]
feasible, (ii) lead to good PR system success (i.e. in terms of high classification rate), (iii) reduce the problem data (raw measurements of patterns) into a significant amount of information without discarding vital information.

In general, we wish to have classification or recognition of a pattern that is invariant to some changes or deviations in the pattern from the ideal case. One of the causes for such deviations is noise. In many situations, a set of patterns from a single class show wide variations. For example, a character from English alphabet (Figs. 1.3-1.6). Classifying such characters involves feature analysis of each character. Some examples of pattern distortions are shown in Figs. 1.3-1.6. Therefore, a careful choice of invariant features (e.g. with respect to Rotation, Scale and Translation (RST)) and pattern structure is to be used for recognition. For example, RST-invariant moment features may be used for shape recognition application.
Figure 1.4: Scaling of pattern ‘A’

Figure 1.5: Variants of pattern ‘A’

Figure 1.6: Some occlusions of pattern ‘A’
The goal of pattern recognition or classification is to distinguish between different classes of patterns, hence it is good to find the basis for this *discriminative ability*. One obvious answer we can give is that patterns of different classes are composed of features with different numerical values. However, this may not be always true. For example, two halftone pictures, say car and bus, comprise of same black and white dots (features). However, spatial arrangements of those two pictures are different and may be used for classification.

Features are arranged in a $d$-dimensional *feature vector*, denoted by $\mathbf{x}$, which gives multi-dimensional feature space. If each feature is a real number, the feature space is $\mathbb{R}^d$ i.e. hypercube. Quite often classification is carried out by partitioning feature space into regions for each class. If dimensionality of a feature vector is large (e.g. an image vector of size $300 \times 300$), the feature extraction techniques (e.g. PCA) play a vital role in dimensionality reduction (Fig. 1.7(a)). A classifier partitions feature space into class-labelled *decision regions*. Decision regions must cover feature space and be disjointed (one exception to this is the notion of fuzzy sets). With these ideas, classification of a feature vector, $\mathbf{x}$, is done as follows: we find the decision region (in $\mathbb{R}^d$) into which $\mathbf{x}$ falls and assign $\mathbf{x}$ to this class. Although the classification principle looks simple, the determination of decision regions is a challenging task. We can also come out with a number of classifiers based on *discriminant functions*. In the $c$-class case, discriminant functions, denoted by $g_i(\mathbf{x})$, $i = 1, \ldots, c$ are used to partition $\mathbb{R}^d$ as follows.

Decision Rule: Assign $\mathbf{x}$ to class $h_m$ (Region $R_m$), where $g_m(\mathbf{x}) > g_i(\mathbf{x}) \ \forall i = 1, \ldots, c$ and $i \neq m$. 
Figure 1.7: (a) Feature extraction process in pattern recognition (b) Piecewise (Linear) decision regions. (c) Hyperbolic (Quadratic) decision regions.
It is obvious that $g_k(x) = g_l(x)$ defines a decision boundary. Fig. 1.7(b) shows some sample decision regions.

To get the best achievable performance for a PR system, it is good to use maximum amount of information available (\textit{a priori information}) such as sample patterns with known class origin. A set of typical patterns, where salient attributes or the class label of each is known, forms a training set, $X$. In general, the training set provides significant information required to associate input data with class labels. The training set is used to make the system to learn relevant information. \textit{Supervised learning} assumes a labelled training set, $X$ (i.e. patterns with class labels), where as in \textit{unsupervised learning}, the class labels of $X$ are not known and the system must determine natural groups of training data. Each group is named with a class label.

We need to use appropriate PR approaches for the application under consideration. The \textit{Statistical PR} (StatPR) approach may be used if there is an underlying and quantifiable statistical basis for the generation of patterns. In StatPR, features are assumed to be generated by a state of nature and therefore the underlying model is of a state of nature or class-conditioned set of probabilities and/or probability density functions. The \textit{Syntactic PR} (SyntPR) approach may be used if interrelationships or interconnections of features give some important information, which facilitates structural description or classification (e.g. musical pattern analysis, speech recognition, handwriting analysis, etc). However, in SyntPR we must be able to quantify and extract structural information and to assess structural similarity of patterns. In some other cases, where neither statistical basis of patterns nor structural description is available, we may use other PR approaches, for example \textit{Neural Network based PR}
may be suitable for pattern association applications. That is we can train a Neural Network to correctly associate input patterns with desired classes [144]. The main difference between neural network and other approaches to pattern recognition are that these networks have the ability to learn complex non-linear input-output relationships and use sequential training procedures. In addition, neural networks have the general characteristic of adapting themselves to the data. The most commonly used family of neural networks for pattern classification tasks is the feed forward networks such as Multi Layer Perceptron (MLP), Radial Basis Function (RBF) networks [73] and Self-Organizing Map (SOM) [86] (which is mainly used for data clustering and feature mapping). Neural networks provide unified approaches for feature extraction and classification and flexible procedures for finding good, moderately nonlinear solutions. Template matching is another simple and well known approach to pattern recognition. In template matching a ‘template’ or an ideal prototype of the pattern to be recognized is available. The pattern to be recognized is matched against the stored templates while taking into account all allowable translation, rotation and scale changes [72]. Fu introduced the notion of attributed grammars which unifies Syntactic and Statistical Pattern Recognition approaches [48].

1.2 An Overview of Feature Extraction (Dimensionality Reduction)

Feature extraction is an important stage of pattern recognition (Figs. 1.1 and 1.7(a)). Extraction of features is an important step and strongly influences classifier
design. That is, if the extracted features show significant differences from one class to another, the classifier can be designed more easily with better performance. Therefore, the extraction of features is a key issue in pattern recognition.

Feature extraction is generally considered as a process of mapping the original features (measurements) into more effective features. If the mapping is linear, the mapping function is well defined and the task is simply to find the coefficients of a linear function so as to maximize or minimize a criterion. To determine these mapping coefficients, we can use the linear algebra techniques for simple criteria and we can apply optimization techniques for complex criteria. Unfortunately, in many applications of pattern recognition, there are salient features which are non-linear mappings of original measurements. Since there is no general theory to generate such nonlinear functions systematically and find the optimum one, extraction of features becomes problem-specific.

In large multi-dimensional data sets, it is usually advantageous to discover some structure from the data. Thus we assume that the data are governed by a certain number of underlying parameters (features). The minimum number of features required to account for the observed properties of the data is called intrinsic dimensionality of the data set. Geometrically, the entire data lies on a topological hyperspace of dimensions equal to intrinsic dimensionality [49]. Some recent efforts to compute intrinsic dimensionality may be found in [124] [180] [28].

An ideal feature extraction technique yields a set of features that makes the job of the classifier trivial. The goal of the feature extraction technique is to characterize a pattern to be recognized by measurements (features) whose values are very similar
for patterns in the same class and very different for patterns in different classes. This leads to the problem of finding distinguishing features that are invariant to transformations (e.g. Rotation, Scale, Translation) of the input pattern (Figs. 1.3–1.6). In speech recognition, we want the features that are invariant to translations in time and to changes in the overall amplitude. We may also want the features that are sensitive to the duration of the word, i.e. invariant to the rate at which the pattern evolves. Rate variation is a serious problem because even the same person speaks at different rates causing the speech signal to change in complex ways. Similarly cursive handwriting also varies as the writer speeds up.

The feature extraction may be domain-dependent and thus requires knowledge of the domain. A good feature extractor for fingerprint classification, may not be useful for face recognition or palmprint recognition.

From our experience, we know that classification of patterns as done by human beings is based on a very few of the important attributes (features). Similarly we attempt to design PR systems on the basis of a few significant features characterizing the class membership of the patterns, preferably those that would be used by humans for classification. The main motivation for keeping the number of pattern dimensions to the absolute minimum is to curtail the effect of the curse of dimensionality phenomenon (i.e. the number of feature points is the exponential function of feature dimension [11]) on the complexity of the classifier. It is obvious that probability of misclassification does not increase with the number of features increased provided the class-conditional densities are completely known (or equivalently the number of training data is large and representative of underlying densities). However in prac-
tice the performance of classifier degrades as the number of features increases if the
number of features is relatively smaller than number of training samples [131][132].
The difficult task is to find out significant features for classification among available
features. This problem may be increased if we do not know the process by which
patterns are generated. To alleviate this problem, it is necessary to use some feature
extraction technique (transducer) which is able to retain as much information about
patterns as possible. Some recent work on to avoid curse of dimensionality may be
found in [3] [8] [155].

It is well known that often patterns contain a large number of features (e.g. a
face image of $300 \times 400$ size consists of 120000 features). These patterns may contain
features which are redundant or irrelevant to the classification task. Moreover, the
pattern generating mechanism and the feature extraction techniques (transducers) are
likely to introduce some distortion and noise, in addition to natural pattern variability.
Therefore the fundamental task of feature extraction technique is to extract most
useful information from the original pattern and present it in a form of a lower
dimensionality vector, whose components represent the most significant features of
input pattern. The goal of feature extraction technique is not merely dimensionality
reduction, but to remove any redundant and irrelevant features which may reduce the
classifier performance. Further role of the feature extraction process is to establish
whether it is necessary to seek additional features which would contain discriminatory
information allowing the improvement of classifier performance.

Most feature extraction techniques compress the observed information into a lower
dimensional space to facilitate its transmission or storage or classification. Here the
elimination of the irrelevant information and redundancy (using feature extraction techniques) is an integral part of the transformation which maps original pattern vector $\mathbf{x}$ into $\mathbf{y}$, the transformed vector, a new lower dimensional feature space which is given by

$$\mathbf{y} = f(\mathbf{x}) \quad (1.1)$$

The mapping $f(\ldots)$ is obtained by optimizing a criterion function $J(\ldots)$.

As the number of dimensions increases the generalization ability is likely to come down for finite training data. Feature extraction may be seen as the one which improves the generalization ability of the recognition system by keeping number of dimensions at minimum. Therefore its performance on unknown patterns is a trade-off error probability for estimation errors. The benefits of dimensionality reduction by feature extraction may be limited when we restrict the form or criterion of mapping. At times we implicitly assume an over-simplistic model of the pattern recognition system. For example, if the classes are not linearly separable and mapping of feature extractor is restricted to a linear form [35].

Some commonly used feature extraction techniques include Principal Component Analysis (PCA), Linear Discriminant Analysis [49], Independent Component Analysis [15][26][97], Projection Pursuit [47], Random Projections [33][43]. A good overview of feature extraction techniques may be found in Refs. [16][72].

Here we shall focus on PCA henceforth in this work.
1.3 Classical Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is one of the widely used techniques for
dimensionality reduction with widespread applications to data reduction, image pro-
cessing, visualization, pattern recognition, exploratory data analysis, etc, [76][35][39][75].
PCA is known by different names *Karhunen-Loeve transform, Hotelling transform, Signal Subspace or Eigenstructure approach*. In pattern recognition, PCA is used in
various forms for optimal feature extraction and data compression [35]. In image
processing PCA defines the Hotelling or KL transform that is applied in image data
compression [71]. In signal processing, a useful characterization of signals is to assume
that they roughly lie in signal subspace defined by PCA. Several methods of signal
modelling, spectrum estimation, and array processing are based on this concept [161].

PCA is concerned with explaining the *variance-covariance structure* through a
few linear combinations of the original features. Although $d$ components are required
to reproduce the total system variability, often much of this variability can be ac-
counted for by a small number, $r (< d)$, of the principal components (PCs) from the
original $d$ features. The $N$ measurements with $r$ PCs are used to replace original $N$
measurements with $d$ features [75].

PCA is used as an intermediate step in many investigations such as cluster analysis
or multiple regression. Algebraically, PCs are particular linear combinations of $d$
random variables (features), $x_1, x_2, \ldots, x_d$. Geometrically these linear combinations
represent the selection of a new coordinate system obtained by rotating the original
system with $x_1, x_2, \ldots, x_d$ as coordinate axes. The new axes represent the directions
with maximum variability and provide a simple description of covariance structure.
PCs depend solely on the covariance matrix, \( \mathbf{C} \) (or the correlation matrix, \( \rho \)) of \( x_1, x_2, \ldots, x_d \). Computation of PCs does not require a multivariate normal distribution assumption. Let the random vector \( \mathbf{x} = [x_1, x_2, \ldots, x_d]^T \) have the covariance matrix \( \mathbf{C} \) with eigenvalues \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d \geq 0 \).

Consider the linear combinations

\[
y_1 = \mathbf{l}_1^T \mathbf{x} = l_{11}x_1 + l_{12}x_2 + \ldots + l_{1d}x_d \tag{1.2}
\]

\[
y_2 = \mathbf{l}_2^T \mathbf{x} = l_{21}x_1 + l_{22}x_2 + \ldots + l_{2d}x_d \tag{1.3}
\]

\[
\vdots \quad \vdots \quad \vdots \quad \vdots 
\]

\[
y_d = \mathbf{l}_d^T \mathbf{x} = l_{d1}x_1 + l_{d2}x_2 + \ldots + l_{dd}x_d \tag{1.4}
\]

where \( \mathbf{l}_i = [l_{i1}, l_{i2}, \ldots, l_{id}]^T \) is a coefficient vector. Then the variance (\( Var \)) and covariance (\( Cov \)) of these linear combinations are given as follows. Here \( E(\ldots) \) indicates expectation function.

\[
Var(y_i) = E[\mathbf{l}_i^T \mathbf{x} \mathbf{x}^T \mathbf{l}_i] = \mathbf{l}_i^T E[\mathbf{x} \mathbf{x}^T] \mathbf{l}_i = \mathbf{l}_i^T \mathbf{C} \mathbf{l}_i; \ i = 1, 2, \ldots, d \tag{1.5}
\]

\[
Cov(y_i, y_j) = E[\mathbf{l}_i^T \mathbf{x} \mathbf{x}^T \mathbf{l}_j] = \mathbf{l}_i^T E[\mathbf{x} \mathbf{x}^T] \mathbf{l}_j = \mathbf{l}_i^T \mathbf{C} \mathbf{l}_j; \ i, j = 1, 2, \ldots, d \tag{1.6}
\]

The PCs are those uncorrelated linear combinations \( y_1, y_2, \ldots, y_d \) (eqs. 1.2-1.4) whose variances (eq. (1.5)) are as large as possible. It is clear that variance specified in eq. (1.5) can be increased by multiplying \( \mathbf{l}_i \) by some constant. Thus to eliminate this problem it is good to restrict the length of coefficient vectors (eigenvectors) to unity. Therefore we can enumerate:
First Principal Component = linear combination $\mathbf{l}_1^T \mathbf{x}$ that maximizes $\text{Var}(\mathbf{l}_1^T \mathbf{x})$ subject to $\mathbf{l}_1^T \mathbf{l}_1 = 1$,

Second Principal Component = linear combination $\mathbf{l}_2^T \mathbf{x}$ that maximizes $\text{Var}(\mathbf{l}_2^T \mathbf{x})$ subject to $\mathbf{l}_2^T \mathbf{l}_2 = 1$ and $\text{Cov}(\mathbf{l}_1^T \mathbf{x}, \mathbf{l}_2^T \mathbf{x}) = 0$, and so,

at the $i^{th}$ step,

$i^{th}$ Principal Component = linear combination $\mathbf{l}_i^T \mathbf{x}$ that maximizes $\text{Var}(\mathbf{l}_i^T \mathbf{x})$ subject to $\mathbf{l}_i^T \mathbf{l}_i = 1$ and $\text{Cov}(\mathbf{l}_i^T \mathbf{x}, \mathbf{l}_j^T \mathbf{x}) = 0$, $\forall j < i$.

A sample PC direction is shown in Fig. 1.8(a). Let $\mathbf{C}$, the covariance matrix associated with random feature vector $\mathbf{x} = [x_1, x_2, \ldots, x_d]^T$, have the eigenvalue-eigenvector pairs $(\lambda_1, \mathbf{e}_1), (\lambda_2, \mathbf{e}_2), \ldots, (\lambda_d, \mathbf{e}_d)$, where $\lambda_1 \geq \lambda_2, \ldots, \lambda_d \geq 0$. Then the
\( y_i = \mathbf{e}_i^T \cdot \mathbf{x} = e_{i1}x_1 + e_{i2}x_2 + \ldots + e_{id}x_d; \ i = 1, 2, \ldots, d \quad (1.7) \)

The variance summarized by \( i^{th} \) PC, \( y_i \), is given by

\[
\text{Var}(y_i) = \mathbb{E}[\mathbf{e}_i^T \cdot \mathbf{x} \cdot \mathbf{x}^T \cdot \mathbf{e}_i] = \mathbf{e}_i^T \cdot \mathbb{E}[\mathbf{x} \cdot \mathbf{x}^T] \cdot \mathbf{e}_i = \lambda_i; \ i = 1, 2, \ldots, d
\quad (1.8)
\]

and the covariance between PCs, \( y_i \) and \( y_j \) is given by

\[
\text{Cov}(y_i, y_j) = 0, \ \forall i, j = 1, 2, \ldots, d; i \neq j
\quad (1.9)
\]

If some eigenvalues are equal, the choices of corresponding eigenvectors, \( \mathbf{e}_i \) and hence \( y_i \), are not unique. It is also true that the sum of variance of original variables (features), \( x_1, x_2, \ldots, x_d \), is equal to the sum of eigenvalues. In other words,

\[
\sigma_{11} + \sigma_{22} + \ldots + \sigma_{dd} = \sum_{i=1}^{d} \text{Var}(x_i) = \lambda_1 + \ldots + \lambda_d = \sum_{i=1}^{d} \text{Var}(y_i)
\quad (1.10)
\]

Each component of the coefficient vector (eigenvector), \( \mathbf{e}_i = [e_{i1}, e_{i2}, \ldots, e_{id}]^T \) needs to be inspected. The magnitude of \( e_{ip} \) measures the importance of the \( p^{th} \) variable to the \( i^{th} \) PC, irrespective of other variables (features). \( e_{ip} \) is proportional to the correlation coefficient between \( y_i \) (i.e. \( i^{th} \) PC) and \( x_p \) (i.e. \( p^{th} \) variable). In other words,

\[
\rho_{y_i,x_p} = \frac{(e_{ip} \cdot \sqrt{\lambda_i})}{\sqrt{\sigma_{pp}}}; \ i, p \in \{1, 2, \ldots, d\}
\quad (1.11)
\]

Also it was observed that eigenvalues and eigenvectors obtained from covariance matrix (\( \mathbf{C} \)) are, in general, not the same as the ones derived from correlation matrix (\( \rho \)). Further, PCs derived from \( \mathbf{C} \) are not a simple function of the \( \rho \). Variables (features) should probably be standardized if they are measured on scales with widely
differing ranges or if the measurement units are not commensurable. For example, if a variable \( x_1 \) represents annual sales in the range Rupees 1 million to 10 millions and \( x_2 \) is the ratio (net annual income)/(total assets) that falls in the range 0.01 to 0.6, then the total variation will be just due to sales variable, \( x_1 \). Here first PC shows weighting heavily \( x_1 \) and weight for variable \( x_2 \) is quite negligible. Alternatively, if both variables are standardized, their subsequent magnitudes are of the same order and both the variables, \( x_1 \) and \( x_2 \), may have significant weights in the construction of the PCs. When attempting to interpret subject matter of the PCs (i.e. to interpret a PC in terms of variables), the correlations, \( \rho_{y_i,x_p} \) (eq. (1.11)) may be more reliable guides than the PC coefficients, \( e_{ip} \). The correlations allow for differences in the variances of the original variables and therefore avoid the interpretive problem caused by different measurement scales.

An unusually small eigenvalues from either covariance or correlation matrix may indicate an unnoticed linear dependency in the data set. If this is the case, one or more of the variables (features) is redundant and should be deleted. For example, let \( x_1, x_2, x_3 \) be student scores in various subjects and the total score, \( x_4 \), is the sum \( x_1 + x_2 + x_3 \). Then even though the linear combination \( e^T \cdot x = [1, 1, 1, -1] \cdot x = x_1 + x_2 + x_3 - x_4 \) is always zero, rounding error in the computation of eigenvalues may lead to a small nonzero value. If the linear expression (dependency) \( x_4 = x_1 + x_2 + x_3 \) was unnoticed, the smallest eigenvalue-eigenvector pair provide a hint to its existence. Thus although large eigenvalues and associated eigenvectors are important in PCA, eigenvalues very close to zero should not be overlooked in a routine way. The eigenvectors associated with these latter eigenvalues may reveal linear dependencies in
the data set that can cause interpretive and computational problems in a subsequent analysis [75].

1.3.1 How to Perform PCA on a Given Set of Patterns?

Consider \(X = \{X_1, X_2, \ldots, X_N\}\), the set of \(N\) training patterns of dimensionality \(d\). The steps of classical PCA are given as follows [120].

1. Perform a ‘mean subtraction’ operation upon the training data patterns.

2. Calculate the Covariance Matrix:

\[
(C)_{d \times d} = \frac{1}{N} \sum_{i=1}^{N} [X_i - \bar{X}] [X_i - \bar{X}]^T
\]  

where \(\bar{X}\) is the mean of training patterns, \(X\).

3. Determine eigenvalues, \(\lambda\) and eigenvectors, \(e\) of the covariance matrix, \(C\) such that

\[
(C)_{d \times d} \cdot (e)_{d \times 1} = (e)_{d \times 1} \cdot (\lambda)_{1 \times 1}
\]  

If the rank of \(C\) is \(d\) then one can find \(d\) eigenvalues.

4. Sort the eigenvalues and the corresponding eigenvectors in non-increasing order.

5. Choose first \(r (< d)\) column eigenvectors, (denoted by \(E_{d \times r}\)) and project the data set \(X\) on \(E_{d \times r}\) to get the reduced data \(X^r\).

\[
X^r_{N \times r} = X_{N \times d} \cdot E_{d \times r}
\]
The original data can be reconstructed with most of the variance (information) from compressed or reduced data $X_{N \times r}$ as given by

$$\hat{X}_{N \times d} = X_{N \times r} E_{r \times d}^T$$

where $\hat{X}_{N \times d}$ is reconstructed data.

### 1.3.2 Why is PCA so Popular in Dimensionality Reduction?

It is observed that the use of PCA is widespread for dimensionality reduction. Here we enumerate some reasons:

- PCA is a global scheme because it considers covariances (correlations) between every pair of original $d$ variables (features). Therefore it is highly effective in dimensionality reduction when global variations are prominent.

- PCA is an optimal linear scheme (in terms of mean squared error) for compressing a set of high dimensional vectors into a set of low dimensional vectors and for reconstructing the original vectors (Section 1.3).

- The model parameters (eigenvectors and eigenvalues) can be computed directly from the data i.e. by diagonalizing the sample covariance (eq. 1.13).

- Compression and Decompression are easy operations and they require only matrix multiplications (eqs. 1.14-1.15).

- Further, PCA minimizes representation entropy and it makes output variables (PCs) mutually uncorrelated [79].
1.3.3 Fundamental Problems/Issues with Classical PCA

Although PCA is popular in its application, there are several issues which inhibit an analyst. Here we describe a few of them based upon our study.

1. *High computational complexity.* The classical PCA methods are not suitable for high dimensional data \((d)\) or large number of data points \((N)\), because PCA needs high computational requirements for such data in particular for huge image data. Computing the sample covariance matrix or correlation matrix itself is expensive \((O(N.d^2))\). To understand the severity of the problem, consider 50 training images, each of size \(100 \times 100\) pixels, that is \(d = 100.100 = 10^4\). The time complexity to calculate covariance matrix is \(O(50.10^4.10^4)\), which is quite expensive [103][176]. The problem is multiplied with bigger images. Now the question arises whether one can avoid computation of sample covariance matrix explicitly or any other means to reduce the time complexity.

2. *Poor performance (that is less generalization ability) with data of prominent local variations.* PCA is a global feature extraction technique which may perform well when *global* variations among patterns are dominant (e.g. screaming face expression spreads across entire face), however it may not perform well when *local* variations among patterns are dominant (e.g. smile of a face is limited to mouth region) [53]. Can we find a way which extracts *both local and global features* to adapt to different scenarios, which can improve classification performance?
3. Small Sample Size (SSS) problem. PCA may not be good to reduce dimensions in the case of a Small Sample Size (SSS) problem. That is when number of training samples are less as compared to number of features or dimensions, PCA may not effectively perform dimensionality reduction or feature extraction. Therefore SSS problem has to be resolved before extraction of principal components [189]. How to resolve SSS problem? [131][132]

4. Not suitable to handle missing data. Another problem with classical PCA approaches is that it is not known how to deal properly with incomplete data set, in which some feature values are missing. The incomplete points are either discarded or completed using some interpolation methods. However, such approaches may not be useful if a significant part of measurements are unknown [18][62][101].

5. Not suitable to handle outliers data. In general, training data may contain some errors or noise from the underlying data generation method. Such data objects that distract the discovery of the underlying model are called as outliers. The classical PCA algorithm is based on the assumption that the data have not been spoiled by outliers. However in practice, outliers do exist in the data and can divert the principal components towards them and away from the direction of maximum variance, thus spoiling the projected data (Figs. 1.8(a) and 1.8(b)) [18][30].

6. Not good for non-linear data. Further, Classical PCA is suitable in applications where the underlying structure is linear. The linear PCA either needs more
principal components or unsuitable for the data sets where nonlinear structure is present [145][24].

7. *Choosing right number of principal components.* Right choice of principal components influence classifier performance as well as the total amount of variance (structure) in the reduced data [76][123]. Now the question arises as to how to choose right number of Principal Components?

In the next chapter, we discuss the state-of-the-art PCA methods existing in the literature to address the problems faced by classical PCA.

### 1.4 Summary of Contributions

Our ideas/contributions are layed out across the following chapters in this thesis. In this section we summarize all our ideas for ready reference. Our work focus on solving the problems of (i) high time complexity with high dimensional data, (ii) low generalization ability when either local variations or global variations among the patterns are dominant and (iii) SSS problem, related to classical PCA and other recent PCA methods. In addition, our approach retains the near optimality in terms of summarization of variance, which leads to near-optimal compression and reconstruction of the patterns.

First of all, we adopt a feature partitioning approach to PCA computation. Here a novel framework with a general scope is proposed for feature partitioning based PCA (FP-PCA) approaches. In this framework, each pattern is divided into $k \geq 2$ sub-patterns (blocks). Then we extract local features from these sub-patterns using
Principal Component Analysis technique. We call these features *local* because they are extracted from the sub-patterns, but not from whole patterns. Finally all these local features are combined by a systematic procedure.

Next, we analyze the generalized feature partitioning framework and bring out several fundamental issues to be addressed, which arise due to partitioning of patterns. These issues include (i) how to divide a given pattern into sub-patterns?, (ii) how to choose sub-pattern size or number of sub-patterns?, (iii) impact of feature ordering of patterns, (iv) how to combine locally-extracted features from sub-patterns?, (v) impact of overlapping features across sub-patterns, (vi) how to choose principal components from each of the sub-patterns?, (vii) impact of truncation of features in the last sub-pattern if its size is less than other sub-patterns, (viii) impact of loss of inter-sub-pattern correlations or covariances, etc. It is to be noted that any method which uses the feature partitioning framework is required to address at least some of these issues.

Further, we deduce from the general framework, a novel FP-PCA approach, SubXPCA. SubXPCA divides a given pattern by choosing a fixed number of contiguous features in the order of appearance. If a fewer number features are present in the last sub-pattern as compared to other sub-patterns, those features are truncated. However, we try to choose sub-pattern size so as to minimize the loss of features in the last sub-pattern. In SubXPCA, we form a set, $P^j$ of all $j^{th}$ sub-patterns and features are extracted (their scope is limited to sub-patterns) from every sub-pattern set using classical PCA technique. As a final step, SubXPCA combines these locally-extracted features systematically by applying PCA to extract global features (here
inter-sub-pattern covariances are exploited). SubXPCA chooses fixed number of Principal Components from every sub-pattern set (PCs may also be chosen based on some threshold).

We further investigate application of feature partitioning framework exclusively for image data by exploiting matrix structure of images. In this direction, we propose first of our FP-PCA approaches on image data, Sub-IMage Principal Component Analysis (SIMPCA), where in each image is divided vertically (number of rows remains same) into $k$ sub-images. Then local image features are extracted from each of $k$ sub-image sets using 2DPCA (also known as IMPCA) [189][187]. The features thus extracted are concatenated to form local feature vector. We improve upon SIMPCA by proposing another FP-PCA approach, FLExible Image Principal Component Analysis (FLPCA), which divides each image in the same way as SIMPCA. FLPCA uses more sophisticated way for combining local image features by utilizing inter-sub-image covariances or correlations. These correlations aid in removing redundant features across sub-images, which may improve subsequent recognition/classification.

A noteworthy observation is that SIMPCA captures local variation of image features by finding image principal components, whereas FLPCA captures global variation of locally-extracted image features by finding PCs across sub-images.

To have deeper insight of FP-PCA approaches, we perform a theoretical study of these approaches to establish general properties of them.

Further, we apply the FP-PCA approach, SubXPCA for cluster analysis. In our approach, we combine DBSCAN [40] with SubXPCA method to find clusters with correlation structures in subsets of data. Such correlation connected clusters may be
invisible when the entire data is considered. Such clusters may give some indication about feature dependencies, cause-effect relationships in some subset of the data. For example, in medical applications, one cluster may indicate dependency between age and dosage. In this feature partitioning approach to correlation connected clusters, we combine our ideas of SubXPCA with DBSCAN way of clustering to elicit correlation structures in the neighbourhood of a core data object.

As our last contribution, we investigate the relevance of feature partitioning paradigm to PCA based subspace classification. Here we use our FP-PCA approach, SubXPCA (or any such approach in principle), to compute subspace for each class (category). A test pattern is classified to the class, for which maximum norm is obtained for the pattern after projection onto corresponding PCs.

1.5 Organization of Thesis

The rest of the thesis is organized as follows. In Chapter 2 we review PCA literature in brief and state the problem. We propose a generalized feature partitioning framework and the issues arise in the framework are presented in Chapter 3. An instance of the proposed feature partitioning framework, Cross-Sub-Pattern Correlation based Principal Component Analysis (SubXPCA), its time complexity analysis, etc., is discussed in Chapter 4. Moving in this direction, we extend feature partitioning approach to image data based on Two-dimensional structure of images by proposing two approaches, Sub-IMage based Principal Component Analysis (SIM-PCA) and FLEXible Image Principal Component Analysis (FLPCA) in Chapter 5. In Chapter 6, we perform a theoretical study of FP-PCA approaches with respect to
variance-covariance structure of the data. Subsequently we show how an FP-PCA approach can be used for cluster analysis and subspace classification in Chapters 7 and 8 respectively. We conclude in Chapter 9.