Variants of Orthogonal Neighborhood Preserving Projections for Image Recognition

by

PURVI AMRUTLAL KORINGA
201321010

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

November, 2018

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Doctor of Philosophy at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Purvi Amrutlal Koringa

Certificate

This is to certify that the thesis work entitled VARIANTS OF ORTHOGONAL NEIGHBORHOOD PRESERVING PROJECTIONS FOR IMAGE RECOGNITION has been carried out by PURVI AMRUTLAL KORINGA for the degree of Doctor of Philosophy at *Dhirubhai Ambani Institute of Information and Communication Technology* under my supervision.

Suman K Mitra
Thesis Supervisor

Acknowledgments

As my journey as a Ph.D. scholar at DA-IICT is ending, I would like to use this opportunity to thank all those who have contributed to my work' directly or indirectly (while attempting to acknowledge everyone, my sincere apologies for missing anyone out).

First and above all, I want to convey my sincere gratitude to my supervisor Prof. Suman K Mitra for not only guiding me in my research but for teaching me many invaluable life lessons. He has been very supportive and encouraging throughout my tenure as a Ph.D. scholar. He closely supervised my work, providing critical observations and suggestions as and when needed, giving me enough freedom to work at my pace and still ensuring that I stay focused on the goal. I am eternally grateful for everything you have taught me.

My special thanks go to Prof. V. Sunitha and Prof. Aditya Tatu for contributing in many ways to my thesis. Thanks for all the insightfull suggestions and interesting observations made during my research progress seminar. A special thanks to Prof. K. S. Dasgupta (Director, DA-IICT), Prof. Ranendu Ghosh and Prof. Manish Narwaria for refining my work by their valuable suggestions during the synopsis. I am grateful to the reviewers Prof. Balasubramanian Raman (Associate Professor, IIT-Roorkee) and Prof. Farzin Deravi (Professor, University of Kent, UK) for spending their valuable time to carefully review the thesis and their useful suggestions.

Special thanks to BRNS, DAE-India for partially funding this research work. I would like to thank Prof. Vijayan K. Asari (Professor, University of Dayton, Ohio) for his collaboration. I would also like to convey thanks to Dr. Anil K Roy for teaching me many skills apart from research during my tenure at DA-IICT. I wish

to acknowledge all the professors of DA-IICT who have inspired me directly or indirectly.

A very warm thanks to my senior PhD scholars Dr. Bhavesh Dharmani, Dr. Mukesh Goswami, Dr. Gitam Shikkenawis, Dr. Ashish Phophaliya, Dr. Milind Padalkar, Dr. Ramnaresh and Dr. Vandana Ravindran for their motivation and support. A special thanks to fellow researchers Prashant, Sujata, Radhika, Anjali, Nidhi, Miral, Parth and Pranav for being amazing tea-buddies. Thanks to DA-IICT staff for all the support.

I cannot express in words the gratitude I have for my parents *Amrutlal* and *Heena*. Thank you maa and papa for having unwavering faith in me. I am eternally grateful to my brother *Dennis* for always being there for me. A very special thanks to my sister-in-law *Mansi*. A special thanks to my parents-in-law for all the support. Finally, A simple thanks will not do justice to my husband *Deep*, I'm so thankful for having you in my life, for standing by my side and being my strength during hard times.

Contents

Abstract

With the increase in the resolution of image capturing sensors and data storage capacity, a huge increase in image data is seen in past decades. This information upsurge has created a huge challenge for machines to perform tasks such as image recognition, image reconstruction etc. In image data, each observation or a pixel can be considered as a feature or a dimension, thus an image can be represented as a data point in the very high-dimensional space. Most of these high-dimensional images lie on or near a low-dimensional manifold. Performing machine learning algorithms on this high-dimensional data is computationally expensive and usually generates undesired results because of the redundancy present in the image data. Dimensionality Reduction (DR) methods exploit this redundancy within the high-dimensional image space and explore the underlying low-dimensional manifold structure based on some criteria or image properties such as correlation, similarity, pair-wise distances or neighborhood structure.

This study focuses on variants of one such DR technique, Orthogonal Neighborhood Preserving Projections (ONPP). ONPP searches for a low-dimensional representation that preserves the local neighborhood structure of high-dimensional space. This thesis studies and addresses some of the issues with the existing method and provides the solution for the same. ONPP is a three-step procedure, in which the first step defines a local neighborhood followed by the second step which defines locally linear neighborhood relationship in high-dimensional space, the third step seeks a lower-dimensional subspace that preserved the relationship sought in the second step.

The major issues with existing ONPP technique are local linearity assumption even with varying size of the neighborhood, strict distance based or class membership based neighborhood selection rule, non-normalized projections or susceptibility to the presence of outliers in the data. This study proposes vari-

ants of ONPP by suggesting modification in each of these steps to tackle above mentioned problems that better suit image recognition application. This thesis also proposes a 2-dimensional variant that overcomes the limitation of Neighborhood Preserving Projections (NPP) and Orthogonal Neighborhood Preserving Projections (ONPP) while performing image reconstruction. All the new proposals are tested on benchmark data-sets of face recognition and handwritten numerals recognition. In all cases, the new proposals outperform the conventional method in terms of recognition accuracy with reduced subspace dimensions.

Keywords: Dimensionality Reduction, manifold learning, embeddings, Neighborhood Preserving Projection (NPP), Orthogonal Neighborhood Preserving Projections (ONPP), image recognition, face recognition, text recognition, image reconstruction

List of Tables

List of Figures

CHAPTER 1

Introduction

Recognition or identification is widely used application in the field of machine learning, where a test data is to be labeled based on the samples present in the training dataset. In last few decades, there has been a huge increase in image data, increasing need for image recognition. Many methods have been developed for image recognition task including appearance-based methods which are widely studied in last decade. While working with the images in these methods, one is often confronted with *curse of dimensionality* [1] because these images can be thought of as points in high dimensional vector space, with each input dimension corresponding to the brightness of one pixel in an image. Thus, the vector representation of an $m \times n$ image leads to an mn-dimensional vector, which does not allow fast computation and demands high computational facility [2, 3]. The redundancy of image data has already exploited in many image processing tasks such as compression [4], denoising [5] etc.

Although the input dimensionality may be quite high, the perceptually meaningful intrinsic structure resembles to a low-dimensional linear or non-linear manifold [6, 7]. This leads to the development of data Dimensionality Reduction techniques to remove the curse of dimensionality and reducing undesired and redundant information present in higher dimensional space. As discussed in [8, 9, 10, 11] dimensionality reduction is now not limited to applications related to computer vision, it is being employed to other machine learning tasks, too.

1.1 Overview of Dimensionality Reduction

The fundamental idea of Dimensionality Reduction (DR) techniques is to seek a linear or non-linear transformation to map the high dimensional data to a lower dimensional subspace which facilitates, among others, classification, recognition or compression of high-dimensional data [12, 13]. In last two decades, many linear and non-linear dimensionality reduction techniques have been proposed [14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]. In next two section we will discuss popular linear and non-linear dimensionality reduction techniques in brief.

1.2 Linear Dimensionality Reduction Techniques

Linear dimensionality reduction techniques basically learn a transformation matrix (or a set of bases vectors) to map higher dimensional data to lower dimensional data points [28]. Some of the most popular examples of linear dimensionality reduction methods are Principal Component Analysis (PCA) [29, 14], Independent Component Analysis (ICA) [30], Linear Discriminant Analysis (LDA) [31], these techniques search low dimensional representation of data assuming data lies on or near a linear manifold. PCA finds subspace from available data such that preserves the direction of maximum covariance. LDA seeks a subspace where within class scatter is minimized and between-class variance is maximizes making classification tasks more accurate. ICA tries to make components as independent of each other as possible.

A generic Linear DR problem can be defined as follows. Given a set of N data points $\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}$ in $\mathcal{R}^{mn}$, find a transformation matrix $\mathbf{V}$ which maps these N data points to a set of points $\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_N}$ in $\mathcal{R}^{d}(d << mn)$ where $\mathbf{y_i}$ is a low dimensional representation of $\mathbf{x_i}$ that can be obtained by a matrix-vector multiplication, $\mathbf{y_i} = \mathbf{V^T x_i}$.

1.2.1 Principal Component Analysis (PCA)

Principal Component Analysis [14] performs dimensionality reduction by projecting the original mn-dimensional data vector on d-dimensional ($d << mn$) linear subspace by the strongest d eigen-vectors of the data covariance matrix. PCA projection transforms correlated features of high-dimensional space into uncorrelated features in low-dimensional space. The objective function of PCA is given by a maximization problem

$$\arg\max \sum_{i=1}^{N} \parallel \mathbf{y_i} - \bar{\mathbf{y}} \parallel^2 \tag{1.1}$$
$$\text{here, } \bar{\mathbf{y}} = \sum_{i=1}^{N} \mathbf{y_i}$$

Solving the Eq. (1.1) leads to the closed form solution as an eigen-value problem $\mathbf{CV} = \lambda\mathbf{V}$. Here, $\mathbf{C}$ is a covariance matrix of given data matrix $\mathbf{X}$. Matrix $\mathbf{V}$ is a projection matrix having eigen-vectors of $\mathbf{C}$ corresponding to highest d eigen-values as d columns.

While performing PCA on image data, an image is converted into a mn-dimensional vector thus may loose inter-pixel dependency of an image in at least one direction. To overcome this limitation of PCA, 2D variant of PCA has been proposed in [32, 33] that handle an image as a matrix of size $m \times n$. In [34] 2-dimensional 2D-PCA is proposed that considers row and column directions of images to find projection matrix and proved to be better than 2D-PCA and PCA.

PCA preserves global structure of data, but fails to understand the underlying non-linear manifold. To handle this problem to some extent, a non-linear variant of PCA using kernel was proposed in [35] which improves performance when linear classifiers are used. Later a generalized framework for learning algorithms was proposed in [36] based on kernel PCA. Another disadvantage of PCA is its poor performance when data have prominent variations, ModPCA (Modular PCA) proposed in [37] divides each of the patterns present in the data into sub-patterns and extracts local PCs from these sub-patterns. ModPCA proved to

be better at classification task.

PCA has been successfully employed in image recognition problems such as face recognition [29], palmprint recognition [38], image classification [39]. However, PCA is an unsupervised DR model, which does not incorporate any information about class membership of training data. To get an advantage of class knowledge, LDA [31] was introduced.

1.2.2 Linear Discriminant Analysis (LDA)

LDA [31] tries to maximize the separability between different classes, simultaneously finds a compact representation of data from the same class using class label knowledge. The objective function is designed using within-class scatter and between-class scatter as

$$\arg\max \frac{\mathbf{V^T S_B V}}{\mathbf{V^T S_W V}} \tag{1.2}$$

$$\mathbf{S_B} = \sum_{i=1}^{C} n_i(\boldsymbol{\mu}^i - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}^i - \bar{\boldsymbol{\mu}})^T$$

$$\mathbf{S_W} = \sum_{i=1}^{C}(\sum_{j=1}^{n_j}(\mathbf{x_j^i} - \boldsymbol{\mu}^i)(\mathbf{x_j^i} - \boldsymbol{\mu}^i)^T)$$

Here, $\mathbf{S_W}$ is the within-class scatter matrix, $\mathbf{S_B}$ is the between-class scatter matrix, C is the number of classes in the given dataset, n_i is the number of data points in class i, $\boldsymbol{\mu}$ is the mean vector of all data points, $\boldsymbol{\mu}^i$ is the mean vector of i^{th} class and $\mathbf{x_j^i}$ is the j^{th} data point of class i. The projection basis for LDA can be obtained by solving generalized eigen-value problem $\mathbf{S_B V} = \lambda \mathbf{S_W V}$

Due to the supervised approach of LDA, LDA is proved to be better at image recognition tasks [40]. 2D variants are also proposed in [41, 42] that performs better than LDA. Another variant Marginal Fisher Analysis (MFA) [43] extends LDA by characterizing the intraclass compactness and interclass separability.

1.2.3 Indepent Component Analysis (ICA)

Another widely used linear dimensionality reduction method is Independent Component Analysis (ICA) [15]. This technique reveals hidden features of underlying sets of random variables by minimizing the statistical dependence of these variables. ICA is related to PCA and factor analysis but it is more powerful technique for finding factors or sources of information. ICA representation is found by making the features of observed data as independent as possible, which is given by

$$\mathbf{Y} = \mathbf{V}\mathbf{X} \tag{1.3}$$

here, $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}]$ is matrix of observed data, $\mathbf{V}$ is a transformation matrix and $\mathbf{Y} = [\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_N}]$ are the maximally independent components. Computational complexity of ICA is much higher than that of PCA [30].

The above mentioned approaches have a common assumption that the underlying data manifold is linear in nature, which may not be true. On the contrary it is often proved that the data like face images etc. lies on or near a non-linear manifold [18, 19, 20]. In last decade, many non-linear manifold learning methods have been proposed to overcome the problem arises with linearity assumption.

1.3 Non-linear Dimensionality Reduction Techniques

Nonlinear dimensionality techniques such as Isometric MAP (ISOMAP) [18], Locally Linear Embeddings (LLE) [19] and Laplacian Eigenmaps (LE) [20] were developed to search underlying low dimensional manifold when data lie in a non-linear manifold where linear techniques do not perform well [18, 44]. Such manifold learning based methods have drawn considerable interests in recent years.

1.3.1 ISOmetric MAP (ISOMAP)

Isometric Map (ISOMAP) [18] is a non-linear dimensionality reduction technique which finds a d-dimensional Euclidean space which preserves the geodesic dis-

tance between data points in high-dimensional space. In a case, where data is lying on or near a non-linear manifold $\mathcal{M}$, for two arbitrary points, the Euclidean distance between them may not accurately reflect their intrinsic similarity as compared to their geodesic distance.

To find actual geometry of the data, pairwise geodesic distances are used. The objective function of ISOMAP is

$$\arg\min \parallel \tau(\mathbf{D_G}) - \tau(\mathbf{D_y}) \parallel^2 \tag{1.4}$$

here, $\mathbf{D_G}$ is the matrix of pairwise geodesic distance in the original high-dimensional space and $\mathbf{D_y}$ is the matrix of pairwise Euclidean distance between embeddings $\mathbf{y}$ in the projection space. Multidimensional Scaling can be used to find the bases of newly found space, the eigen-vectors corresponding to smallest d eigen-values can be used to project data on the linear subspace.

1.3.2 Laplacian Eigenmaps (LE)

The Laplacian Eigenmap (LE)[20] uses a graph embedding approach. An undirected neighborhood graph is formed, where each data point is a vertex. Data points $\mathbf{x_i}$ and $\mathbf{x_j}$ are connected by an edge with weight $w_{ij} = 1$ if $\mathbf{x_j}$ is among the k nearest neighbors of $\mathbf{x_i}$, otherwise the edge weight is set to zero. This simple weighting method has been found to work well in practice.

To find a low-dimensional embedding of the graph, the algorithm tries to embed points that are connected in the graph as close to each other as possible and does not care what happens to the other points. The objective function is given as

$$\arg\min_{i,j} \parallel \mathbf{y_i} - \mathbf{y_j} \parallel^2 \mathbf{W_{ij}} \tag{1.5}$$

here, $\mathbf{W}$ is a weight matrix representing the neighborhood graph. $\mathbf{Y}$ are low dimensional embeddings of data $\mathbf{X}$.

The low-dimensional embeddings are eigen-vectors corresponding smallest d

eigen-values of the generalized eigen-value problem $\mathbf{LY} = \lambda\mathbf{DY}$. Here $\mathbf{L} = \mathbf{DW}$ is the graph Laplacian and D is a diagonal matrix with elements $\mathbf{D_{ii}} = \sum_j \mathbf{W_{ij}}$. However, this cost function has an undesirable trivial solution: putting all points in the same position would minimize the cost. This can be avoided by adding suitable constraints. In practice, the smallest eigen-value corresponds to this trivial solution, but the eigen-vectors corresponding to the next smallest eigen-values give the Laplacian Eigenmap embeddings [45]. In [46], the author proposes a mechanism, named Laplacian score, to choose the best features in projected subspace in the unsupervised environment.

1.3.3 Locally Linear Embedding (LLE)

Locally Linear Embedding [19] is a scheme that finds global low-dimensional coordinates when data lies on (or very near) a linear manifold embedded in a high-dimensional space. LLE considers a small neighborhood patch near a data points and assumes it to be a linear patch and finds low-dimensional coordinates that best preserves this linearity.

Firstly, A neighborhood graph for data point $\mathbf{x_i}$ is constructed based on Euclidean distance, and weights w_{ij} are sought such that the linear combination $\sum w_{ij}\mathbf{x_j}$ best represents the data point $\mathbf{x_i}$. The weight $w_{ij} = 0$, if $\mathbf{x_j}$ is not a neighbor of $\mathbf{x_i}$. In the following step, low dimensional embeddings $\mathbf{y_i}$ are sought such that the neighborhood relationship between the data point and its neighbors can be represented with the same weight computed in the first step. The objective function is given by

$$\arg\min \sum_{i=1}^{N} \parallel \mathbf{y_i} - \sum_{j=1}^{N} w_{ij}\mathbf{y_j} \parallel^2 \tag{1.6}$$

In the first step, LLE learns local geometry by considering a small patch of data and its neighbors. In the second step, LLE stitches these local neighborhood patches together to explore non-linear nature of the manifold [47, 48]. Being a non-linear method LLE does not give an embedding for out-of-sample data points, making it useless for tasks such as recognition. LLE gives an embedding in lower-

dimensional space, but there is no mechanism to choose best features, in [49] LLE score is proposed that evaluates the LLE embeddings and assigns a score to each feature that best represents the objective of LLE.

Various variants have been proposed to improve the performance of LLE. [50] proposes Incremental LLE that provides a mechanism to include an out-of-sample data point in the already learned manifold. However, the procedure is computationally expensive. In [51], an enhanced scheme is suggested for neighborhood selection in the supervised mode.

1.4 Linear DR with Non-linear Properties

Being a non-linear DR, LE and LLE provide non-explicit mappings, these methods do not have any mechanism to handle out-of-sample data points [52]. Inclusion or exclusion of any data point forces these DR to learn entirely new embeddings.

1.4.1 Locality Preserving Projections (LPP)

Unlike PCA which preserves the global structure of data by preserving directions of maximum variances, LPP [6, 53] projects data to optimally preserve local structure of data. LPP achieves these by preserving the pairwise Euclidean distances in the local non-linear neighborhood when projected in low dimensional representation. The LPP yields linear maps whose properties are similar to the non linear maps yielded by the eigen-vectors of the true graph Laplacian also known as LE [20].

LPP is a linear approximation of non-linear LE. The local neighborhood is described using adjacency graph. The neighborhood selection steps of LPP is same as LE, where neighbors are chosen based on Euclidean distance between datapoints, if $\mathbf{x_i}$ is neighbor of $\mathbf{x_j}$ then $\mathbf{x_j}$ is also a neighbor of $\mathbf{x_i}$.

In the second step, weights are assigned to these links either by binary scheme or using heat kernel. In binary scheme, $w_{ij} = 1$ if and only if vertices representing data points $\mathbf{x_i}$ and $\mathbf{x_j}$ are connected. In an alternate scheme of weight assignment, euclidean distance is used to assign weights to the link connecting to vertices

representing data points $\mathbf{x_i}$ and $\mathbf{x_j}$, $\mathbf{w_{ij}} = e^{-Dist(\mathbf{x_i},\mathbf{x_j})/t}$. Here, *Dist(.)* is a Euclidean distance. Let $\mathbf{W}$ be the weight matrix, whose ij^{th} element is a weight for link connecting to vertices i and j.

The objective function to find projection bases of LPP is given by

$$\sum_{i,j=1}^{N} \parallel \mathbf{y_i} - \mathbf{y_j} \parallel^2 w_{ij} \tag{1.7}$$

under appropriate constraint.

The optimization problem leads to the closed form solution as generalized eigen-value problem:

$$\mathbf{XLX^T V} = \lambda \mathbf{XDX^T V} \tag{1.8}$$

Here, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix. $\mathbf{D}$ is a diagonal matrix, $\mathbf{D_{ii}} = \sum_i \mathbf{W_{ji}}$. projection matrix $\mathbf{V}$ have d eigen-vectors corresponding to smallest eigen-values as its columns.

Various variants of LPP have been proposed so far. In [54], an enhanced and supervised LPP is proposed that uses path based similarity weights and [55] uses Pearson's coefficient instead of depending on the free parameter of heat kernel which considers Euclidean distance only.

In [56], LPP bases are learned using face database to define Laplacian-faces and used for face recognition application, as shown in [56] Laplacian-faces proved to be more reliable for face recognition as compared to eigen-faces and fisher-faces. LPP gives non-orthogonal bases which makes it difficult to reconstruct the data. The problem is addressed in [57] by orthogonal Laplacian-faces that outperforms Laplacian-faces in face recognition task. [58, 59, 60] give some more solution schemes based on LPP to use for face recognition. 2-dimensional variants of LPP that deals with image as matrix of dimensions $m \times n$ are proposed in [61, 62].

1.4.2 Neighborhood Preserving Embedding (NPE)

NPP or NPE [7] is a linear extension of LLE, the basic idea of NPE is to approximate a data point by a weighted linear combination of its neighbors assuming that they lie on or near a locally linear manifold, and seek transformation matrix for such a low dimensional representation that best preserves the linear combination of all data points.

The first step of NPP involves finding neighbors of a data point $\mathbf{x_i}$, usually performed using grouping methods such as k-Nearest Neighbors (k-NN) or ε-Neighbors (ε-NN). Let $\mathcal{N}_{x_i}$ be the set of k neighbors of $\mathbf{x_i}$, found using either of these techniques. Euclidean distance works well with image data [63].

In the second step, data point $\mathbf{x_i}$ is expressed as a linear combination of its neighbors as $\sum_{j=1}^{k} w_{ij}\mathbf{x_j}$ where, $\mathbf{x_j} \in \mathcal{N}_{x_i}$. The weight w_{ij} are calculated by minimizing the reconstruction errors i.e. error between $\mathbf{x_i}$ and a linear combination of $\mathbf{x_j} \in \mathcal{N}_{x_i}$ as,

$$\arg\min \mathcal{E}(W) = \sum_{i=1}^{n} \parallel \mathbf{x_i} - \sum_{j=1}^{N} w_{ij}\mathbf{x_j} \parallel^2 \tag{1.9}$$

subject to $\sum_{j=1}^{k} w_{ij} = 1$

The problem stated in Eq. (1.9) can be solved for each $\mathbf{x_i}$ individually by solving the least square problem $(\mathbf{X_{N_i}} - \mathbf{x_i}\mathbf{e^T})\mathbf{w_i} = \mathbf{0}$ with a constraint $\mathbf{e^T}\mathbf{w_i} = \mathbf{1}$.

Once the reconstruction weights for high-dimensional data points are found, it is assumed that the same weights along with same neighbors will construct the lower-dimensional representation $\mathbf{y_i}s$. The objective function to find bases of such low-dimensional subspace can be written as

$$\arg\min \mathcal{F}(\mathbf{Y}) = \arg\min \sum_{i=1}^{N} \left\| \mathbf{y_i} - \sum_{j=1}^{N} w_{ij}\mathbf{y_j} \right\|^2 \tag{1.10}$$

with a normalization constraint, to remove arbitrary scaling factor in the projection space.

As discussed earlier, most dimensionality reduction methods work with mn-

dimensional data vector, but it is advantageous to work with $m \times n$ matrix when data is an image. In [64], 2D-NPP is proposed which works with image data and shows that the image recognition results are better than NPE. Even though NPE and 2D-NPP are implemented in supervised mode, only the data points of the same class were used. In NPDE [65], authors used data points from other classes to add discriminative information in learned low-dimensional space that increases the recognition results even more.

In addition to LPP and NPE, there are other techniques to handle non-linearity known as kernel trick. In kernel methods data is projected on high-dimensional space using some standard kernels and then dimensionality reduction methods are applied to the projected data. Various kernel variants of PCA, LDA, LPP are proposed in [35, 66, 67, 68, 36, 69, 70].

For applications like image recognition, data classification etc. adding discriminant constraint can be helpful [71]. Variants incorporating such constraint are proposed in [72, 73, 74]. These dimensionality reduction methods are implemented as supervised methods where class knowledge is available. However, training data having known labels are limited, but in this digital age lots of unlabeled data is available which can be used to find more consistent manifold as proposed in [75, 76]. These methods are categorized as semi-supervised methods. An interesting work is proposed in [77] to estimate the optimal dimensionality of underlying data manifold for discriminative methods. Some of the recent works on DR include use of Deep Neural networks and Generalized Auto Encoders to achieve low-dimensional embeddings of high-dimensional data [78, 79].

Technique and Author	Property	Description and Comment
PCA (Jolliffe et al.)	Variance preservation	Uncorrelated features, Unsupervised technique
LDA (Fisher et al.)	minimizes within-class variance & maximizes between-class variance	Supervised technique. Includes discriminative information
ISOMAP (Tenenbaum et.al.)	Isometric Mapping	Considers geodesic distance and uses MDS. Computationally expensive.
Locally Linear Embedding (Roweis et al.)	Preserves local structural linearity	Computes the reconstruction weights for each point, and then minimizes the embedding cost by solving an eigenvalue problem.
Laplacian Eigenmaps (Belkin et al.)	Locality Preserving	Minimizing the squared gradient of an embedding map is equal to finding eigen-functions of the Laplace-Beltrami operator.
Locality Preserving Projections (X He et al.)	Locality Preserving	Linearization of LE, embeds out-of-sample data
Neighborhood Preserving Projection (X He et al.)	Local linearity preserving	Linearization of LLE, embeds out-of-sample data points.

Table 1.1: Comprehensive summary of well-used DR methods

1.5 Motivation of the Work

With the rapid development in the data capturing methods and increased capacity of data storage caused a huge surge in the dimensionality of datasets. An observation of an object is often represented as a real-valued vector having each feature as an element in a very high dimensional space, where each feature represents one dimension. Generally, distribution of such observations in high dimensional space contains a huge amount of redundant information and thus, an underlying low dimensional subspace can be sought for such observations where only meaningful information is preserved. In applications such as image recognition, object classification and data mining, high dimensions of data poses two major problems - *curse of dimensionality* in statistical pattern recognition which causes very huge computation burden and *small sample size* problem where the number of observation is very few compared to the large number of features. Most of the learning algorithm fails when only limited training data with the large number of features is available. The excessive dimension of the data space often brings the learning algorithms into dimensionality dilemma. So dimensionality reduction is very important for the modeling of small-sample size and high-dimensional data.

A straightforward and widely used approach is PCA [29], where a low dimensional representation is learned such that information variance is maximized, but this approach is unsupervised where the information about data class is not embedded into low dimensional representation. Thus, discriminating approaches such as LDA [31] were proposed that also tries to embed discriminating information between the classes into a low dimensional representation. These linear approaches work under the assumption that the high dimensional space is linear, which may not be the case with real data. LPP [6] and NPE [7] are DR techniques which work with the local non-linearity present in the data. These techniques work in the small neighborhood patches and build a global geometry thus preserving the global geometry of data as well as the local relationship in small neighborhoods. Their orthogonal variants OLPP [57] and ONPP [80] are proved to be useful for image recognition tasks such as text recognition or face recogni-

tion.

Face images or text images are represented as high-dimensional pixel arrays. Due to the high correlation between the neighboring pixel values in real images, they often belong to an intrinsically lower dimensional manifold. Image recognition is one of the most widely used applications where dimensionality reduction is applied prior to the recognition task [7, 80, 20, 64]. Images provides most challenging data for dimensionality reduction. For example in face recognition task, because of high variations in face images such as pose, expressions, illumination conditions, change in appearance etc, they may lie on a nonlinear manifold. Same is true for hand-written texts, where huge variations in orientation, stroke width and curves present huge challenge.

In this thesis, we are focusing on a dimensionality reduction scheme based on neighborhood preserving property. We have addressed some of the issues in the existing ONPP [80] technique and provided the solution for them keeping image recognition task in the focus. Being a linear and an orthogonal variant of LLE, ONPP carries same assumption of local linearity. ONPP defines a local neighborhood for a data point and assigns weights that holds this linearity true but this assumption of local linearity may and may not be true for a larger neighborhood. Very small or a very large number of neighbors adversely affects the learned manifold. This observation inspired us to introduce a non-linear weighing scheme to handle local non-linearity that comes with larger neighborhood well. In image recognition tasks knowledge of class label of training data is always available, thus DR methods are generally implemented in supervised mode. Training data belonging to the same class are considered neighbors of each other, regardless of their similarity or dissimilarity. Such hard decision rules does not help in learning low-dimensional manifold effectively, this motivated the proposal of new neighborhood selection rule that takes into consideration of actual similarity of data pairs based on their Euclidean distance as well as their class membership.

NPE [7] is a linear extension of LLE which gives normalized projection on non-orthogonal basis, whereas ONPP gives orthogonal projection basis but relaxes the normalization constraint on the projections. For applications where image recon-

struction is required, Normalization and orthogonalization both are desired properties but bringing both constraint together increases the optimization difficulty, many iterative solution have been proposed to solve such optimization problem which randomly initiates a projection basis and optimize it iteratively, this continues on ortho-complemented space until desired number of projection bases are achieved. This inspired an alternative solution scheme that iteratively solves for a projection basis that are orthogonal to the previously sought projection bases without random initialization. ONPP also inherits some of the shortcomings of LLE such as susceptibility to outliers, because same as LLE, ONPP also use $l2$-norm in the cost function. The effect of outliers gets magnified due to the use of $l2$-norm. Low-dimensional manifold gets distorted in the presence of the outliers which inspired the design of variant of ONPP which handles outliers present in the data.

1.6 Scope and Accomplishments of the Thesis

The scope of the thesis is to study and improve the performance of DR technique based on neighborhood preservation. ONPP [80] is a linear DR technique that tries to exploit local linearity present in the high-dimensional space and finds its lower dimensional representation by preserving this local neighborhood relationship. Conventional ONPP is based on the hypothesis that the local neighborhood is linear, thus it is very sensitive to the choice of this local neighborhood. Thus, the varying number of neighbors, the presence of outliers etc. affects the performance of ONPP. A few extensions [74, 64, 81] of ONPP have been proposed in the literature. This thesis is an attempt to address few issues with the conventional ONPP approach and provides solutions to achieve more robust and meaningful low dimensional representation, reduce computational complexity to utilize in various pattern recognition tasks.

1. In conventional ONPP algorithm the neighborhood is assumed to be a linear patch, but when the number of neighbors is large, the assumption of linearity could be invalid and it affects the lower dimensional representa-

tion adversely. A piece-wise non-linear weighing scheme is applied to overcome this limitation of ONPP. This Modified ONPP (MONPP) handles the varying number of neighbors better than ONPP. In particular piece-wise linearity, within the small neighborhood is applied which gives rise to a more compact data representation that could be utilized for recognition. The proposed scheme is implemented on synthetic as well as real data. Suitability of the proposal is tested on a set of face images and a significant improvement in recognition is observed. We have also adopted two pre-processing techniques to handle illumination variation in face images which improves recognition performance as well.

2. In supervised settings, conventional ONPP uses knowledge of class label to identify the neighbors of data points, but when data points are closely placed or the classes are overlapping, such hard decision rule may not help to find a good low dimensional representation. To overcome this limitation of neighborhood finding rule, we are proposing a novel neighborhood rule, where low dimensional representation is used with Logistic Regression (LR) to find the probability of every data point to belong into a particular class. Based on these probabilities a new distance measure - Class similarity based distance is defined and used to find nearest neighbors of data points. It is observed that class similarity based ONPP very well represents the relationship of neighbors in low dimensional representation. The proposed scheme used to recognize face images as well as handwritten numerals images, the proposed neighborhood scheme improves recognition performance significantly as compared to conventional ONPP.

3. All DR techniques deal with optimizing a cost function based on some criteria imposed on either projection of data or on the basis of projection space. NPP and ONPP are such linear methods that preserve local linear relationship within the neighborhood, with two different constraints, the normalized projection and the orthogonal basis of subspace, respectively. In this thesis, we proposed a new variant - ONPPn that finds a subspace which satisfies both the constraints. The thesis also provides its two-dimensional vari-

ant. Experiments show that ONPPn outperforms NPP and ONPP versions in image recognition tasks, whereas 2D-ONPPn overcomes the limitation of 2D-ONPP but does not perform as good as 2D-NPP for recognition. 2D-NPP as well as 2D-ONPP are not suitable for image reconstruction task, proposed method 2D-ONPPn overcomes drawbacks of existing methods and is best suited for image reconstruction.

4. Generally, most of the DR techniques involve optimizing a cost function in L2-norm and thus they are susceptible to outliers. However, recently, due to the capability of handling outliers, L1-norm optimization is drawing the attention of researchers. The work documented here is the first attempt towards the same goal where Orthogonal Neighborhood Preserving Projection (ONPP) technique is performed using optimization in terms of L1-norm to handle data having outliers. In particular, the relationship between ONPP and PCA is established theoretically in the light of L2-norm and then ONPP is optimized using an already proposed mechanism of PCA-L1. Extensive experiments are performed on synthetic as well as real data for applications like classification and recognition. It has been observed that when a larger number of training data is available L1-ONPP outperforms its counterpart L2-ONPP.

1.7 Organization of Thesis Chapters

We discuss in detail each of our above mentioned contributory works in the following sections. chapter 2 discusses a modified weighing approach to handle inherent non-linearity present in neighborhood. chapter 3 discusses a class similarity based technique to find neighbors of data point. chapter 4 discusses a new ONPP technique with normalization constraint and its 2D variant. chapter 5 documents a variant of ONPP namely, L1-ONPP to handle outliers present in data. The work uses the fact that l1-norm distance measures handle outlying data or out-lying features better than $l2$-norm distance measures. Basically, in this chapter we establish a relation between PCA and ONPP to achieve L1-ONPP basis.

To analyze the usability of the proposed variants in comparison to the conventional ONPP the recognition experiments are performed on widely used face databases and handwritten numerals databases. It is to be noted that in this thesis, the emphasis is on the development of robust variants of Linear Dimensionality Reduction technique based on Orthogonal Neighborhood preserving projections for image recognition and image reconstruction application. Thus, the performance of these variants is not compared to other state-of-the-art face recognition and text recognition methods. Overall conclusion of the thesis is given in the chapter 6 with possible directions to work further on the same problem.

CHAPTER 2

Modified Orthogonal Neighborhood Preserving Projections

Orthogonal Neighborhood Preserving Projections (ONPP) [80] is a linear extension of Locally Linear Embedding (LLE), which is a nonlinear dimensionality reduction technique that embeds high-dimensional data samples on lower dimensional subspace. This mapping is not explicit in the sense that embedding is data dependent. In LLE, intrinsic data manifold changes with the inclusion or exclusion of data points. Hence, on the inclusion of a new data point, embeddings of all existing data points changes. This prevented subspace based recognition of unknown sample point, as this unknown sample point cannot be embedded into the existing lower dimensional subspace. Lack of explicit mapping thus makes LLE not suitable for recognition. ONPP resolves this problem and finds the explicit mapping of the data in lower dimensional subspace through a linear orthogonal projection matrix. In the presence of this orthogonal projection matrix, a new data point can be embedded into the lower dimensional subspace. The following section documents ONPP and some noticeable observations on ONPP.

2.1 Orthogonal Neighborhood Preserving Projections (ONPP)

As discussed in Chapter 1, an image $\mathbf{x_i}$ of $m \times n$ size is considered a data point in mn dimensional space, let $\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}$ be given images in the mn-dimensional space. So the data matrix is $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}] \in \mathcal{R}^{mn \times N}$. The basic task of sub-

space based methods is to find an orthogonal/non-orthogonal projection matrix $\mathbf{V}^{mn\times d}$ such that $\mathbf{Y} = \mathbf{V}^T\mathbf{X}$, where $\mathbf{Y} \in \mathbf{R}^{d\times n}$ is the embedding of $\mathbf{X}$ in lower dimension as d is assumed to be less than m. Even though the data is sampled in mn dimensional space, data points are assumed to lie on or near a manifold having intrinsic dimensionality d, where $d << mn$.

ONPP is a three-step algorithm where, in the first step a local neighborhood is sought for each data point, followed by the representation of each data point as a linear combination of its neighbors. In the third step, the data compactness is achieved through a minimization problem.

Step 1: Finding Nearest Neighbors: The first step of ONPP involves finding neighbors of a data point $\mathbf{x_i}$, usually performed using grouping methods such as k-Nearest Neighbors (k-NN) or ε-Neighbors (ε-NN).

- k-Nearest Neighbors: In this method, k neighbors are chosen based on some distance measure, where $k \in \mathbb{N}$ is suitably chosen parameter. Here, Euclidean distance is considered while choosing neighbors.
- ε Neighbors: In this technique, neighbors are selected which are enclosed within a sphere of $\varepsilon \in \mathbb{R}$ radius of the data point.

Let $\mathcal{N}_{x_i}$ be the set of k neighbors of $\mathbf{x_i}$, found using either of these techniques.

Step 2: Calculating Reconstruction Weight: The data point $\mathbf{x_i}$ is expressed as a linear combination of its neighbors as $\sum_{j=1}^{k} w_{ij}\mathbf{x_j}$ where, $\mathbf{x_j} \in \mathcal{N}_{x_i}$. The weight w_{ij} are calculated by minimizing the reconstruction errors i.e. error between $\mathbf{x_i}$ and linear combination of $\mathbf{x_j} \in \mathcal{N}_{x_i}$ as,

$$\arg\min \mathcal{E}(W) = \arg\min \sum_{i=1}^{n} \parallel \mathbf{x_i} - \sum_{j=1}^{k} w_{ij}\mathbf{x_j} \parallel^2 \tag{2.1}$$

subject to $\sum_{j=1}^{k} w_{ij} = 1$ i.e. $\mathbf{e^T w_i} = \mathbf{1}$ where, $\mathbf{e} = [1, 1, \ldots 1]$

The problem stated in Eq. (2.1) can be solved for each $\mathbf{x_i}$ individually. For $\mathbf{x_i}$, let $\mathbf{X_{N_i}}$ be a matrix having $\mathbf{x_j}$ as its columns, where $\mathbf{x_j} \in N_{x_i}$. Hence $\mathbf{X_{N_i}}$ is a $m \times k$ matrix. Now by solving the least square problem $(\mathbf{X_{N_i}} - \mathbf{x_i e^T})\mathbf{w_i} = \mathbf{0}$ with

a constraint $\mathbf{e^T w_i} = \mathbf{1}$, a closed form solution, as shown in Eq. (2.2)), is evolved for $\mathbf{w_i}$. Here, $\mathbf{e}$ and $\mathbf{w_i}$ are vectors of dimension $k \times 1$.

$$\mathbf{w_i} = \frac{\mathbf{G_i}^{-1}\mathbf{e}}{\mathbf{e}^T\mathbf{G_i}^{-1}\mathbf{e}} \tag{2.2}$$

where, $\mathbf{G_i}$ is Gramiam matrix of dimension $k \times k$ for data point $\mathbf{x_i}$. Each entry of $\mathbf{G_i}$ is calculated as $\mathbf{g_{pl}} = (\mathbf{x_i} - \mathbf{x_p})^T(\mathbf{x_i} - \mathbf{x_l}),\ for\, \forall\, \mathbf{x_p}, \mathbf{x_l}\ \in \mathcal{N}_{\mathbf{x_i}}$

$$\mathbf{G_i} = \begin{bmatrix} < \mathbf{x_{n_{i1}}} - \mathbf{x_i}, \mathbf{x_{n_{i1}}} - \mathbf{x_i} > & \dots & < \mathbf{x_{n_{i1}}} - \mathbf{x_i}, \mathbf{x_{n_{ik}}} - \mathbf{x_i} > \\ \vdots & \ddots & \vdots \\ < \mathbf{x_{n_{ik}}} - \mathbf{x_i}, \mathbf{x_{n_{i1}}} - \mathbf{x_i} > & \dots & < \mathbf{x_{n_{ik}}} - \mathbf{x_i}, \mathbf{x_{n_{ik}}} - \mathbf{x_i} > \end{bmatrix} \tag{2.3}$$

Step 3 : Finding Projection Matrix : In ONPP, final step is dimensionality reduction or finding the projection matrix $\mathbf{V}$ as stated earlier. This method basically seeks the lower dimensional projection $\mathbf{y_i} \in \mathcal{R}^d$ of the data point $\mathbf{x_i} \in \mathcal{R}^{mn}$ ($d << mn$) such that the linear combination of neighbors $\mathbf{x_j}$ which reconstruct the data point $\mathbf{x_i}$ in higher dimension would also reconstruct $\mathbf{y_i}$ in lower dimension with corresponding neighbors $\mathbf{y_j}$ along with same weight as in higher dimensional space. The constraint imposed here is the orthogonality of the basis vectors $\mathbf{v_i}$s of d dimensional ONPP subspace.

Such embedding is obtained by minimizing the sum of squares of reconstruction errors in the lower dimensional subspace. Hence, the objective function is,

$$\begin{aligned} \arg\min \mathcal{F}(\mathbf{Y}) &= \arg\min \sum_{i=1}^{N} \parallel \mathbf{y_i} - \sum_{j=1}^{N} w_{ij}\mathbf{y_j} \parallel^2 \\ \arg\min \mathcal{F}(\mathbf{V}) &= \arg\min tr[\mathbf{V^T X M X^T V}] \end{aligned} \tag{2.4}$$

$$\tag{2.5}$$

with an orthogonality constraint

$$\mathbf{V^T V} = \mathbf{I_d} \tag{2.6}$$

Lagrange equation of ONPP optimization problem for finding the projection

matrix $\mathbf{V}$ which gives orthogonal bases for the subspace, can be written using orthogonality constraint,

$$\mathcal{L}(\mathbf{V}) = \mathbf{V^T X M X^T V} - \lambda(\mathbf{V^T V} - \mathbf{I_d}) \tag{2.7}$$

Differentiating Eq. (2.7) with respect to λ and equating to zero will give

$$\frac{\delta \mathcal{L}(\mathbf{V})}{\delta \lambda} = 0 \implies \mathbf{V^T V} = \mathbf{I_d} \tag{2.8}$$

Differentiating Eq. (2.7) with respect to $\mathbf{V}$ and equating to zero will give

$$\begin{aligned} 2\mathbf{X M X^T V} - 2\lambda \mathbf{V} &= 0 \\ \mathbf{X M X^T V} &= \lambda \mathbf{V} \end{aligned} \tag{2.9}$$

This optimization problem results in ordinary eigenvalue problem i.e. computing eigen-vectors corresponding to the smallest d eigenvalues of matrix $\mathbf{XMX^T}$. ONPP also explicitly maps $\mathbf{X}$ to $\mathbf{Y}$, which is of the form $\mathbf{Y} = \mathbf{V^T X}$, the only difference from NPP is that ONPP subspace is an orthogonal subspace, but projection $\mathbf{Y}$ is not normalized.

2.1.1 Observations From ONPP

The noticeable aspects of the ONPP algorithm are:

- The core algorithm is simple and involves a few local computation and an eigenvalue problem. However, it involves neighborhood search in high-dimensional space. Note that there are several efficient algorithms for faster neighborhood search [82].

- The basic assumption of ONPP is the local linearity within a local neighborhood, which is not always true for a larger neighborhood. This assumption limits the performance of ONPP on non-linear data manifolds. The neighborhood is defined with parameter k, thus the performance of ONPP highly depends on the choice of k.

The ONPP algorithm is implemented on two synthetic data-sets *viz Swiss-roll* (Fig. 2.1(a)) and *S-curve* (Fig. 2.1(d)) to visualize their two dimensional representation with the varying number of k. These two data-sets reveal a linear relationship within the class as well as between the classes when unfolded. Linear dimensionality reduction techniques such as PCA when applied to these type of data, fail to capture this intrinsic linearity. However, dimensionality reduction techniques such as LPP [6], NPE [7] try to capture the local geometry and retain it into the projection space are expected to perform better. These algorithms give rise to a compact representation of the data without much distorting the shape of the data. To implement existing ONPP on these non-linear data, 1000 data points are randomly sampled from continuous 3D manifold (Fig. 2.1(b), Fig. 2.1(e)) to build the orthogonal transformation matrix $\mathbf{V}$. To explore how ONPP works with varying values of k, experiments have been conducted for $k = 4, 6, 8, 10$. 2D projections using two strongest ONPP dimensions for each experiment are shown in Fig. 2.1(c) and Fig. 2.1(f). It is observed that projection using ONPP algorithm depends on k, variation in k results in huge variation in its lower dimensional representation.

2.2 Modified Orthogonal Neighborhood Preserving Projections

Orthogonal NPP builds a linear relationship within a small neighborhood of the data and then assumes its validity in the lower dimension space. However, the assumption of linearity could be invalid in some applications. With this aim in mind, we introduce an approximate non-linearity in weighing scheme. In particular piece-wise linearity, within the small neighborhood which gives rise to a more compact data representation that could be utilized for recognition. The proposed scheme is implemented on synthetic as well as real data. Suitability of the proposal is tested on a set of face images and a significant improvement in recognition is observed.

Orthogonal Neighborhood Preserving Projection (ONPP) is based on two basic assumptions. First, it assumes that a linear relation exists in a local neigh-

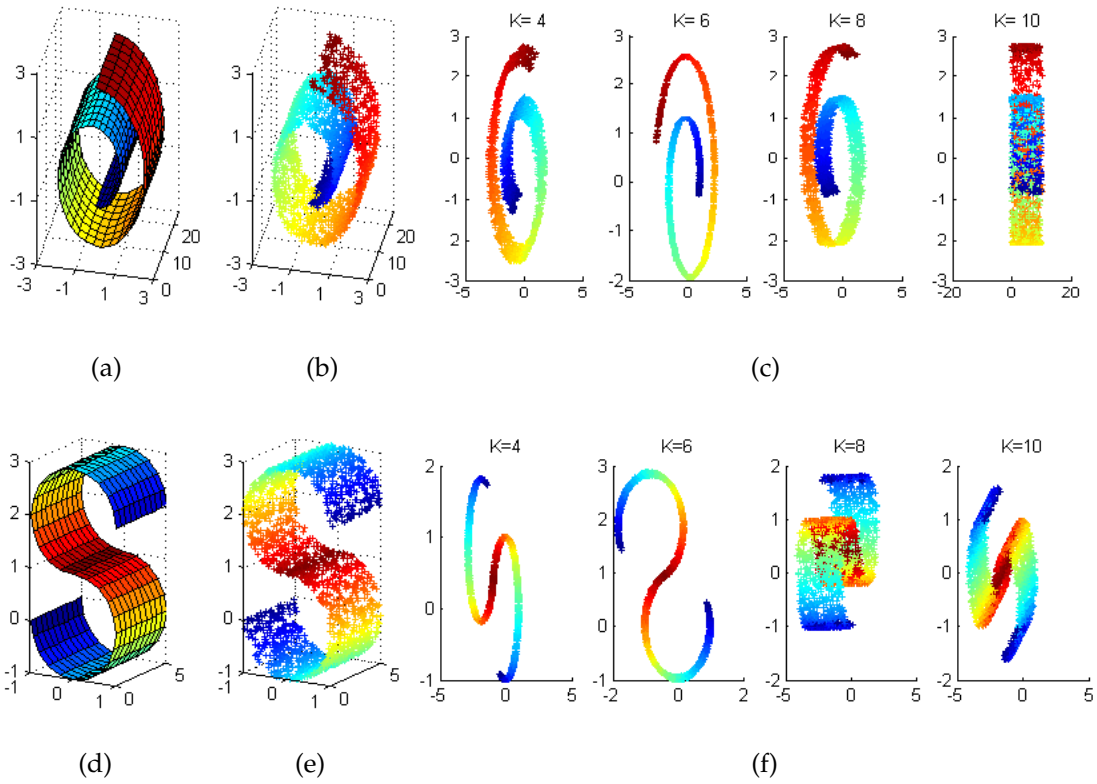


Figure 2.1: 2D ONPP projection with varying value of k (number of NN) for synthetic data (a) Swiss roll: 3D continuous manifold (b) sampled 3D data points (c) 2D projections using two strongest ONPP bases with $k = 4$, $k = 6$, $k = 8$ and $k = 10$. (d) S-curve: 3D continuous manifold (e) sampled 3D data points (f) 2D projections using two strongest ONPP bases with $k = 4$, $k = 6$, $k = 8$ and $k = 10$

borhood and hence any data point can be represented as a linear combination of its neighbors. Secondly, it assumes that this linear relationship also exists in the projection space. The same assumption gives rise to a compact representation of the data that can enhance the classification performance. The data compactness would be more visible in case the first assumption is strongly valid. While experimenting with synthetic data, as shown in Fig. 2.3, it has been observed that data compactness is not clearly revealed. The main drawback could be the strict local linearity assumption. Focusing on this, we are trying to incorporate some kind of non-linear relationship of a data point with its neighbor. The proposed algorithm is termed as Modified ONPP (MONPP).

In this proposed modification, a Z-shaped function is used to assign weights to nearest neighbors in the first stage of ONPP. Note that in ONPP, the weight

matrix **W** is calculated by minimizing the cost function in Eq. (2.1), which is a least square solution given in Eq. (2.2). In the least square solution, weights of neighbors are inversely proportional to the distance of the neighbors from the point of interest. We are looking for a situation where the neighbors closest to the point of interest would get maximum weight and thereafter the weights will be adjusted non-linearly (through Z-shaped function) as the distance increases. After a certain distance, the weights will be very low. In particular, instead of assuming a linear relationship, a piece-wise linear relationship is incorporated through the z-shaped function. This piece-wise linear relationship is leading towards some kind of non-linear relationship given by

$$\mathcal{Z}(x;a,b) = \begin{cases} 1 & \text{if } x \leq a \\ 1 - 2\left(\frac{x-a}{a-b}\right)^2 & \text{if } a \leq x \leq \frac{a+b}{2} \\ 2\left(\frac{x-b}{a-b}\right)^2 & \text{if } \frac{a+b}{2} \leq x \leq b \\ 0 & \text{Otherwise} \end{cases} \tag{2.10}$$

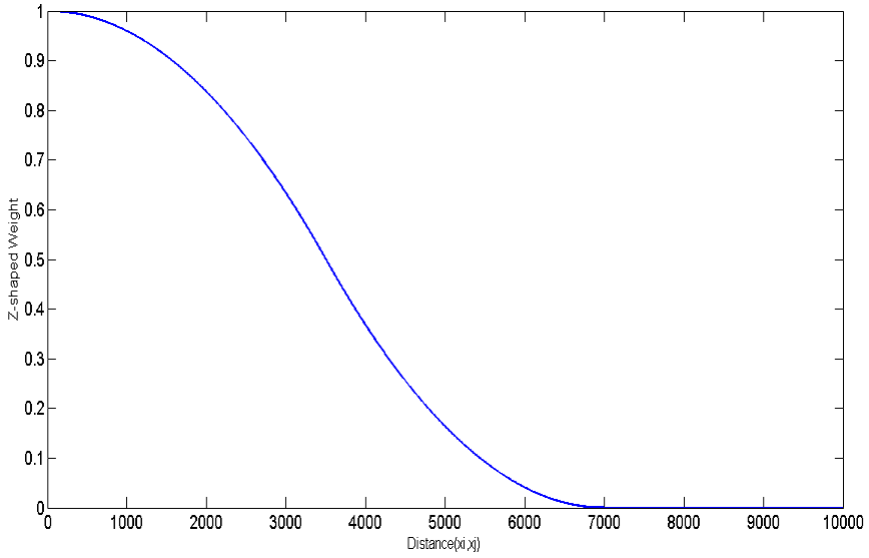


Figure 2.2: Z-shaped weight function for Range [0, Maximum within class distance], illustrated for max distance of 7000 unit

Parameters a and b locate the extremes of the sloped portion of the curve and can be set to 0 and maximum within-class distance (i.e. maximum pairwise distance between data samples belonging to the same class) respectively, as shown in

Fig. 2.2. In case of unsupervised mode, a k-NN algorithm could be implemented before assigning the weights and hence the parameters a and b of Eq. (2.10) can be adjusted.

Finally, Eq. (2.11) is used to assign weight to each neighbor $\mathbf{x_j}$ corresponding to $\mathbf{x_i}$. Note that this equation is same as Eq. (2.2), where $\mathbf{G}^{-1}$ is replaced by $\mathbf{Z}$. The new weights are

$$\mathbf{w_i} = \frac{\mathbf{Ze}}{\mathbf{e}^T\mathbf{Ze}} \tag{2.11}$$

where, elements of this $\mathbf{Z}$ matrix are defined as

$$Z_{pl} = \mathcal{Z}(d_p; a, b) + \mathcal{Z}(d_l; a, b) \; for, \; \forall \, \mathbf{x_p}, \mathbf{x_l} \; \in \mathcal{N}_{\mathbf{x_i}} \tag{2.12}$$

here, $\mathcal{Z}(d_k; a, b)$ is calculated using Eq. (2.10). d_k is the Euclidean distance between $\mathbf{x_i}$ and it's neighbor $\mathbf{x_k}$. Parameters a and b are obtained as discussed earlier.

Next step computes the projection matrix $\mathbf{V} \in \mathcal{R}^{mn \times d}$ whose column vectors are eigen-vectors of the matrix $\mathbf{M} = \mathbf{X}(\mathbf{I} - \mathbf{W})(\mathbf{I} - \mathbf{W^T})\mathbf{X^T}$ corresponding to smallest d eigenvalues. Embedding of $\mathbf{X}$ on lower dimension $\mathbf{Y}$ is achieved by $\mathbf{Y} = \mathbf{V^T X}$.

The proposed Modified ONPP (MONPP) algorithm is summarized below:

Table 2.1: Modified ONPP Algorithm

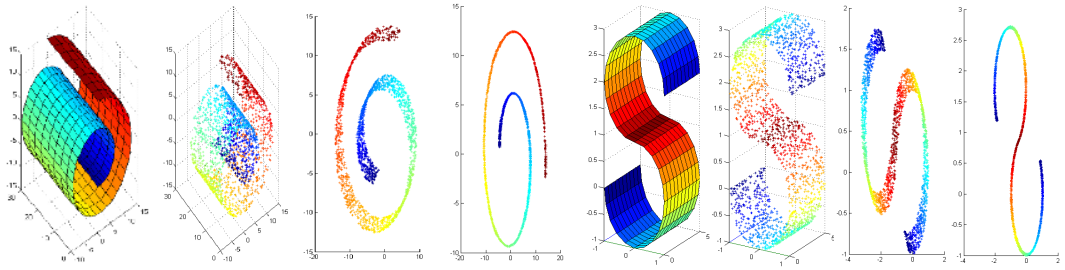
Input:	Dataset $\mathbf{X} \in \mathcal{R}^{mn \times N}$and number of reduced dimension d
Output:	Lower dimension projection $\mathbf{Y} \in \mathcal{R}^{d \times N}$
1.	Project data $\mathbf{X}$ on lower dimension space using PCA
2.	Search neighborhood $\mathcal{N}_{x_i}$ for each data point $\mathbf{x_i}$ with class information
3.	Compute the weight $\mathbf{W}$ for each $\mathbf{x_j} \in \mathcal{N}_{x_i}$ as given in Eq. (2.11)
4.	Compute Projection matrix $\mathbf{V} \in \mathbf{R}^{mn \times d}$ whose column vectors are smallest d eigen-vectors of matrix $\mathbf{M} = \mathbf{X}(\mathbf{I} - \mathbf{W})(\mathbf{I} - \mathbf{W^T})\mathbf{X^T}$
5.	Compute Embeddings of lower subspace dimension by $\mathbf{Y} = \mathbf{V^T X}$

2.3 Experiments and Results

The proposed Modified ONPP (MONPP) is used for two well-known synthetic data-sets along with a digit data [83], a low dimensional projection of these data sets is compared with ONPP. MONPP has also been implemented extensively for various well-known face databases and handwritten numerals databases the results are compared with that of the existing ONPP algorithm.

2.3.1 Synthetic data

The Modified ONPP (MONPP) algorithm is implemented on two synthetic data-sets *viz Swiss-roll* (Fig. 2.3(a)) and *S-curve* (Fig. 2.3(e)) to visualize their two dimensional plot. To implement existing ONPP and proposed Modified ONPP, 1000 data points are randomly sampled from three dimensional manifold (Fig. 2.3(b), Fig. 2.3(f)) to build the orthogonal transformation matrix ($\mathbf{V}$). From Fig. 2.3(c),(d) and Fig. 2.3(g),(h), it is clear that the 2D representations of both *Swissroll* and *S-curve* seem to be much better for MONPP.

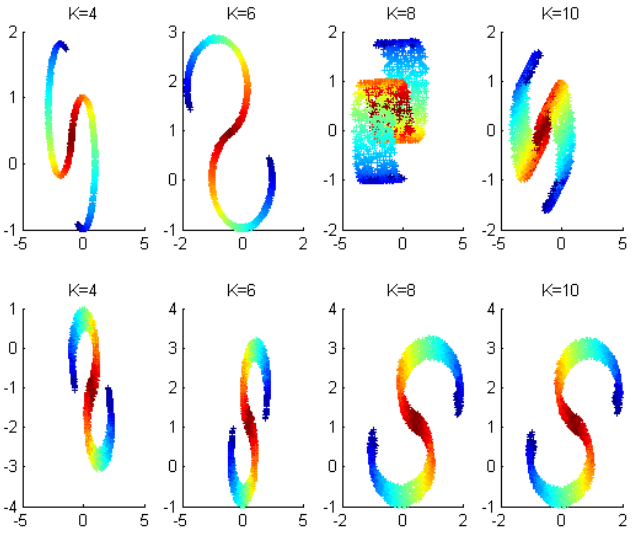


(a) (b) (c) (d) (e) (f) (g) (h)

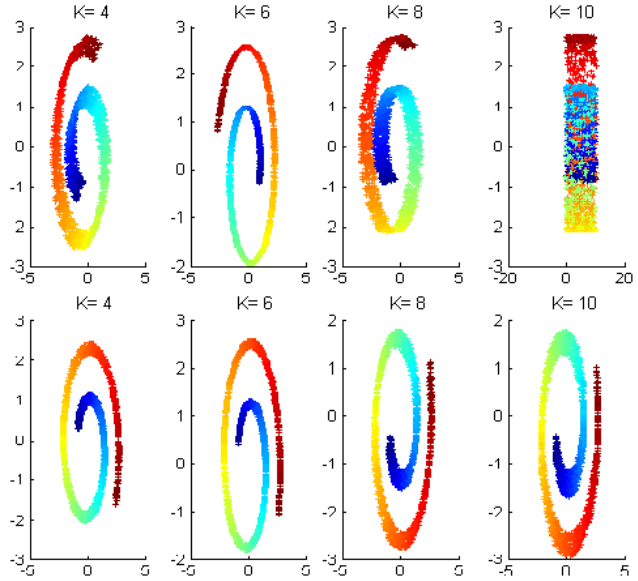
Figure 2.3: Swissroll: (a) original 3D continuous manifold, (b) sampled data, (c) 2D projection obtained by ONPP and (d) MONPP. S-curve: (e) original 3D continuous manifold, (f) sampled data, (g) 2D projection obtained by ONPP and (h) MONPP. k(number of NN) is set to 6.

To explore how ONPP and MONPP work with varied values of k, experiments have been conducted and results are shown in Fig. 2.4. Note that repeated experiments with a fixed k may not guarantee to generate same results. It is observed that projection using ONPP algorithm depends on k, variation in k results in huge variation in its lower dimensional representation. However, projection

using MONPP is more stable with varying values of k. Larger values of k imply a larger area of the local neighborhood. It is possible that larger local area does not possess linearity. The linearity assumption of ONPP thus is invalid here. So the non-linearity present in the moderately large local area is well-captured in MONPP and is reflected in the results.



(a)



(b)

Figure 2.4: 2D projection of (a) S-curve and (b) Swissroll with various k (number of NN) values with ONPP (top) and MONPP (bottom)

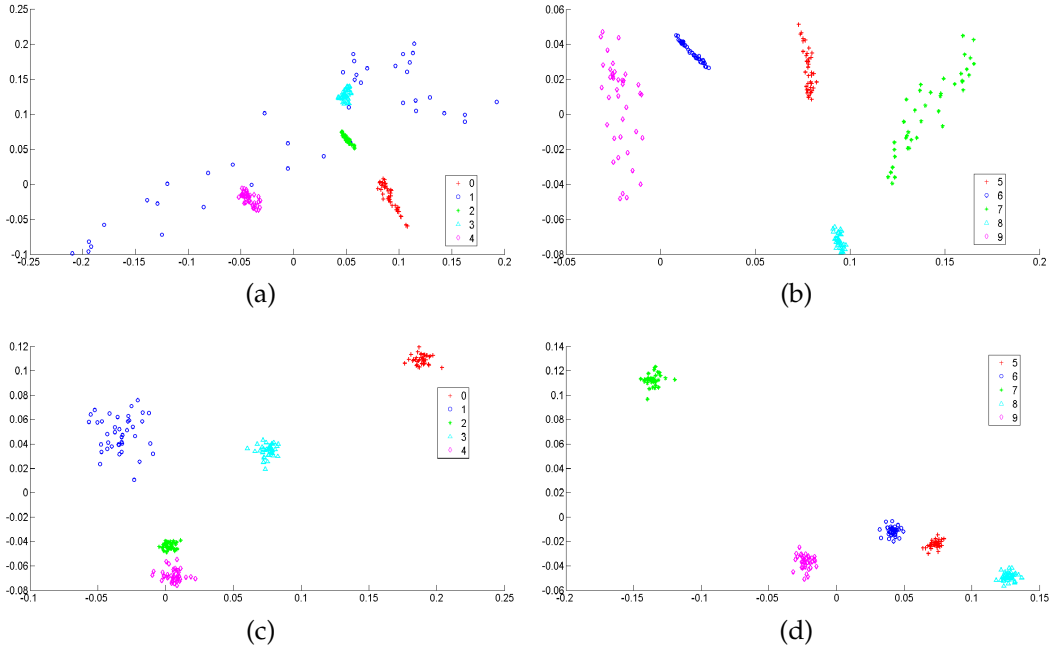


Figure 2.5: 2D projection of MNIST handwritten numerals data using ONPP and MONPP: (a) and (b) shows performance of ONPP algorithm, where, $'+'$ denotes 0, $'o'$ denotes 1, $'*'$ denotes 2, $'\Delta'$ denotes 3, $'\diamond'$ denotes 4. (c) and (d) shows performance of MONPP algorithm, where, $'+'$ denotes 5, $'o'$ denotes 6, $'*'$ denotes 7, $'\Delta'$ denotes 8, $'\diamond'$ denotes 9.

2.3.2 Digit data

The MNIST database [83] of handwritten digits is used to compare data visualization of both the algorithms. Randomly 40 data samples from each class (digit) are taken and projected on 2-D plane using ONPP and proposed Modified ONPP. The results are shown in Fig. 2.5, it can be clearly observed that the data is compact and well separated when MONPP is applied. It seems that there is a wide range of variations in digit '1' and that is reflected in Fig. 2.5 (a). But the same digit '1' is more compact in the 2D representation of MONPP (Fig. 2.5 (c)). A similar argument is true for digits '7 ' and '9'. Overall, better compactness is evident for all digits in case of MONPP.

2.3.3 Text Recognition

Real data like handwritten text or numerals have huge variations in terms of stroke width, shape, patterns, curves and orientations, thus poses a real challenge

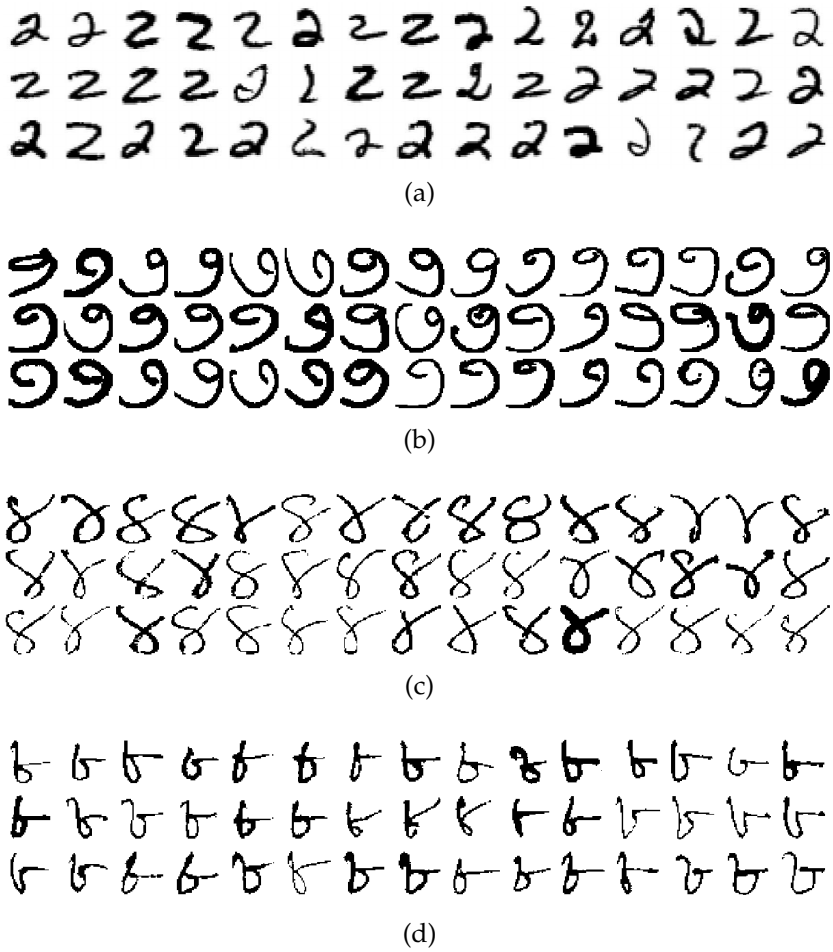


Figure 2.6: Some samples from handwritten numerals databases (a) digit '2' from MNIST database. (b) digit '7' from Gujarati database. (c) digit '4' from Devnagari database. (d) digit '8' from Bangla database. Notice the very diverse shape, stroke width, curves, orientation and pattern of different samples.

in pattern recognition task. To compare the performance of ONPPn and MONPPn with other four methods, handwritten numeral databases in English and three different Indian languages *wiz* Gujarati, Bangla and Devnagari are used. The MNIST digit database [83] contains nearly 68,000 images of digits of size 28×28, some samples from digit '2' are shown in Fig. 2.6 (a). The Gujarati Numerals database [84] has nearly 1300 images for each class, Fig. 2.6 (b) shows some images of digit '7' from the database. Another database for Indian script Devnagari was used for recognition experiment. Handwritten Devnagari database [85] have approximately 1800 images for each class. Fig. 2.6 (c) shows few random samples of digit '4' taken from the data. For this experiment, images are resized to 20×20 and nearly 100 images out of each class were selected randomly as training set and

remaining images are used for testing. Unlike other three numerals databases used here, Bangla [85] have small number of samples accounting to 50 for each class. Fig. 2.6 (d) shows some samples of digit '8' randomly chosen from the data. Out of 50 images of each class 25 images were randomly chosen for training, remaining images were used for testing. Fig. 2.7 shows recognition performance of ONPP and MONPP with varying number of subspace dimensions. It is observed in all four databases that performance of MONPP surpasses that of ONPP with a good margin. Average and best recognition rates are reported in Table 2.2. For classification Nearest Neighbor (NN) is used because of its simplicity.

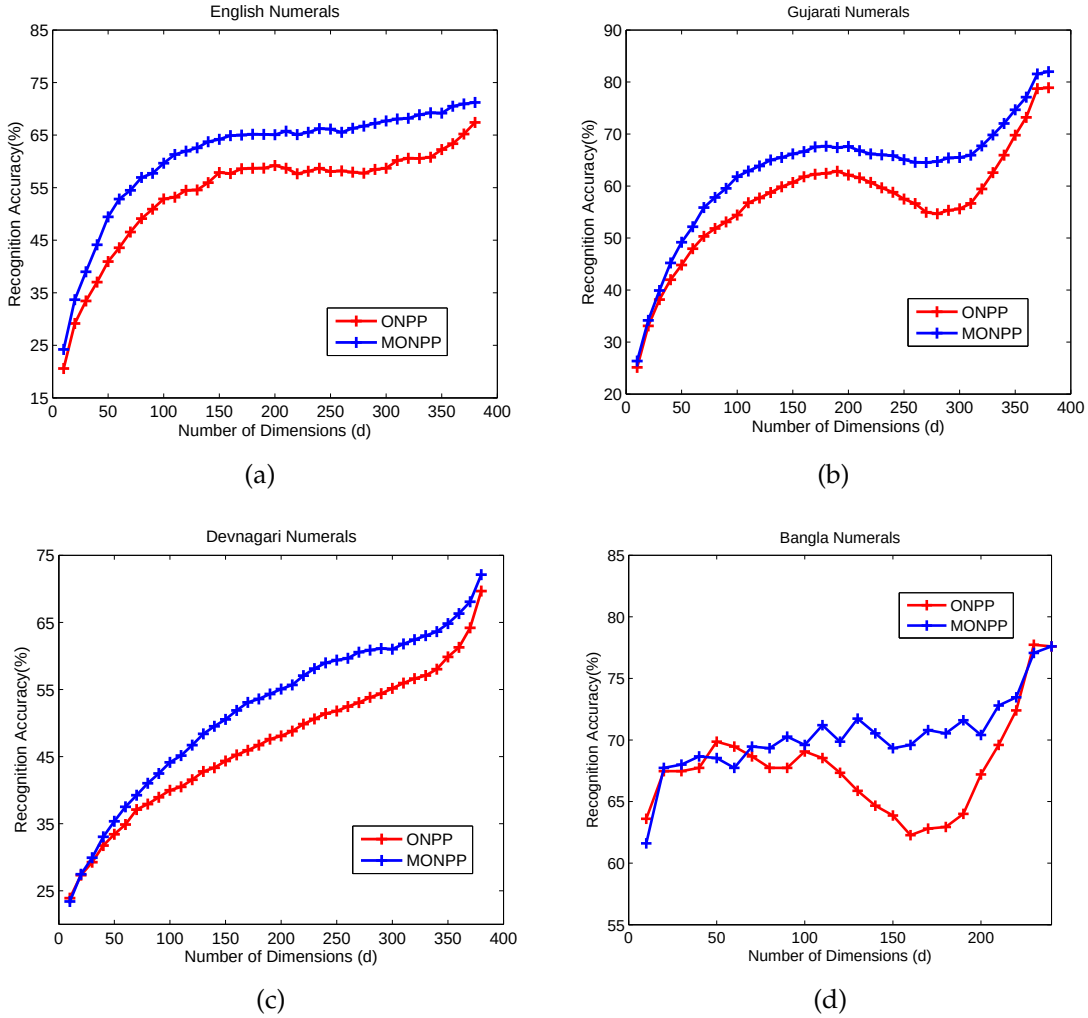


Figure 2.7: Comparison of average performance of ONPP and MONPP with varying number of subspace dimensions d on MNIST database, Gujarati database, Devnagari database and Bangla database

Table 2.2: Comparison of performance of ONPP and MONPP on handwritten numerals databases in the light of recognition score (in %)

	ONPP		MONPP	
Database	Average	Best (at dim d)	Average	Best (at dim d)
English	67.41	70.24 (380)	71.31	74.83 (380)
Gujarati	78.72	82.74 (370)	81.55	84.48 (370)
Devnagari	69.67	71.79 (380)	72.12	74.05 (380)
Bangla	77.80	79.20 (230)	77.20	80.40 (230)

2.3.4 Face Recognition

The algorithm is also tested on three different face databases *viz* AR [86], ORL [87], and UMIST [88]. AR database contains over 4000 color images of 126 individuals featuring frontal views with different facial expressions, illumination conditions, and occlusions. The UMIST face database contains 564 images of 20 individuals, with poses varying from profile to frontal views. ORL database contains a set of 10 different face images for 40 distinct subjects with varying the lighting and facial expressions. Moderate tolerance for some side movement is also allowed. This makes the face recognition more challenging.

To maintain the uniformity, face images of all databases are resized to 38×31 pixels, thus each image is considered as a point in 1178 dimensional space. For each database, randomly 50% of face images are selected as training samples and remaining are used for testing. The training samples are used to find the lower dimensional projection matrix $\mathbf{V}$. The test samples are then projected on this subspace and are recognized using a *NN-classifier*. The main intention of these experiments is to check the suitability of MONPP based image representation for face recognition and hence a simple classifier such as NN is used. To ensure that the results achieved are not biased to the randomized selection of training-testing data, the experiments are repeated twenty times with different randomization. Experiments are also conducted for different values of d (dimension of reduced space) ranging from 10 to 160 (at an interval of 10). The best, as well as average recognition rates, are reported here for all databases.

Average recognition results for AR, UMIST, ORL and The Extended Yale- B databases using ONPP and MONPP are shown in Fig. 2.8. It can be observed

that MONPP performs better than ONPP across almost all values of d. Average recognition accuracy and best recognition accuracy along with the corresponding dimensions using ONPP and MONPP for all three databases are reported in Table 2.3.

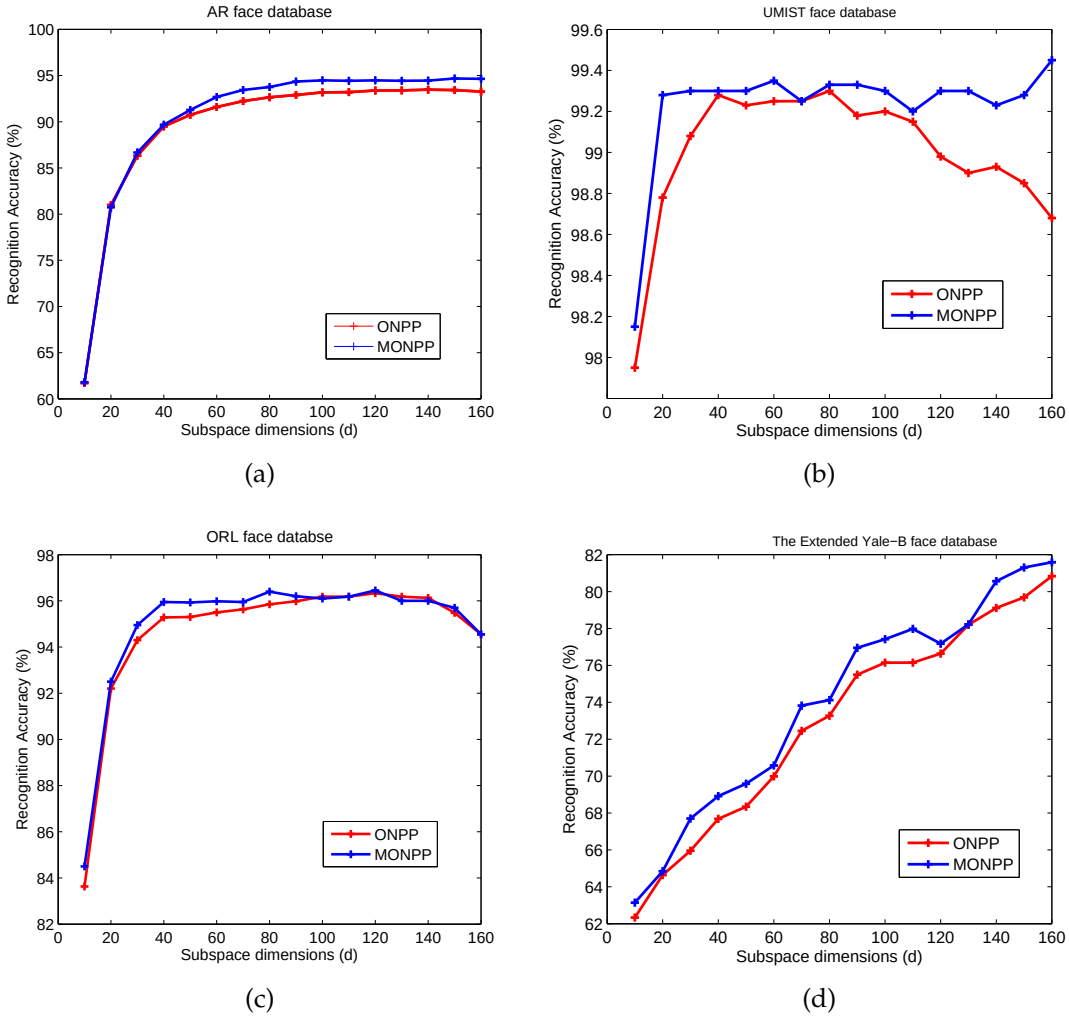


Figure 2.8: Comparison of average performance of ONPP and MONPP with varying number of subspace dimensions d on AR database, UMIST database, ORL database and The Extended Yale-B database

Although face recognition algorithms based on dimensionality reduction techniques are performing exceptionally well under controlled illumination environments, it is still a major challenge to reliably recognize a face under pose, expression, age and illumination variations. As seen the performance on The Extended Yale-B database shown in 2.8 (d) is very poor due to illimination variations present in the images. Illumination variation is most common while capturing face im-

Table 2.3: Comparison of performance of ONPP and MONPP on face databases in the light of recognition score (in %)

	ONPP		MONPP	
Database	Average	Best (at dim d)	Average	Best (at dim d)
AR	93.50	96.25 (100)	94.13	97.50 (110)
UMIST	98.95	99.05 (30)	98.00	99.50 (20)
ORL	93.06	98.00 (120)	95.90	99.00 (60)
Yale-B	92.51	93.42 (140)	94.07	95.80 (100)

ages. Lighting condition, camera position, face position all lead to change in illumination. Such illumination changes result in a major source for recognition errors, especially for appearance-based techniques. The task of face recognition algorithm is to identify an individual accurately despite such illumination variations. Two face images of the same person can seem visually very different under various illumination intensities and directions. In [89], it has been shown that the variations in two face images of the same person captured under different illumination conditions are larger than the face images of two different person, which makes face recognition under illumination variation a difficult task. To handle such cases, several approaches are used such as pre-processing and normalization, invariant feature extraction or face modeling [90]. In pre-processing based methods, several image processing techniques are performed on the image to nullify illumination effects to some extent. Gamma correction, histogram equalization [91, 92] and logarithm transforms [93] are some of these image processing techniques.

2.4 Handling Illumination Variations in Face Images

For better face recognition under uncontrolled and varying lighting conditions, the features useful for discrimination between two different faces need to be preserved. The shadows created in face images due to different lighting directions result in loss of facial features useful for recognition. A pre-processing method must increase the intensity in the areas those are under-exposed (poorly illuminated) and lower the intensity in the areas those are over-exposed (highly illuminated) simultaneously while keeping the moderately illuminated area intact. Follow-

ing two subsections discuss two different pre-processing techniques. In the following section, the recognition performance is compared using ONPP and Modified ONPP (MONPP). Detailed experiments of pre-processing to nullify the illumination variation for face recognition have been performed on various benchmark face databases having illumination variations such as The extended Yale-B database [94] and CMU PIE face database [95]. Face recognition results of ONPP and MONPP are compared and presented here.

2.4.1 Locally Tuned Inverse Sine Nonlinear(LTISN)

An enhancement technique for color images proposed in [96] takes care of such extreme illumination conditions using a series of operations and a nonlinear intensity transformation performed on images to enhance a color image for better visual perception. The intensity transformation function is based on previous research suggested in [97].

In this work, we have tested the nonlinear intensity transformation based on inverse sine function enhancement on gray-scale face images having high illumination irregularities for better recognition. This nonlinear enhancement technique is a pixel by pixel approach where the enhanced intensity value is computed using the inverse sine function with a locally tunable parameter based on the neighborhood pixel values. The intensity range of the image is re-scaled to [0 1], followed by a nonlinear transfer function given in Eq. (2.13).

$$I_{enh}(x,y) = \frac{2}{\pi} sin^{-1}(I_n(x,y)^{\frac{q}{2}}) \tag{2.13}$$

where, $I_n(x,y)$ is the normalized intensity value at pixel location (x,y) and q is the locally tunable control parameter. In the darker area where intensity needs to be increased, the value of q should be less than 1, and the over-bright area where intensity needs to be suppressed, the value of q should be greater than 1. Fig. 2.9 shows the transformation function with the value of q ranging from 0.2 to 5 for intensity range [0 1]. The red curve shows transformation for q equal to 1, green curves show q less than 1, which enhances darker region of image and blue curve

shows q greater than 1, which suppresses higher intensity in the over-exposed region of an image. The curve of the transformation function used for a pixel is decided by the value of q based on its neighborhood.

The value of q is decided by the tangent function based on mean normalized intensity values, which is determined by averaging three Gaussian filtered smooth images. These smooth images are found using three different Gaussian kernels of size $M_i \times N_i$. Normalized Gaussian kernels are created as below:

$$kernel_i(n_1, n_2) = \frac{h_g(n_1, n_2)}{\sum_{n_1} \sum_{n_2} h_g(n_1, n_2)} \tag{2.14}$$

$$h_g(n_1, n_2) = e^{\frac{-(n_1^2+n_2^2)}{2\sigma^2}} \tag{2.15}$$

where, ranges of n_1 and n_2 are $[-\lfloor \frac{M_i}{2} \rfloor, \lfloor \frac{M_i}{2} \rfloor]$ and $[-\lfloor \frac{N_i}{2} \rfloor, \lfloor \frac{N_i}{2} \rfloor]$ respectively. Here, window size $M_i \times N_i$ is set to 6×6, 10×10 and 14×14, experimentally. The subscript i indicates which Gaussian kernel is being used and σ is set to be $0.3(\frac{M_i}{2} - 1) + 0.8$.

The Gaussian mean intensity at pixel (x,y) is calculated using

$$I_{M,i}(x, y) = \sum_{m=-\frac{M_i}{2}}^{\frac{M_i}{2}} \sum_{n=-\frac{N_i}{2}}^{\frac{N_i}{2}} I(m, n) kernel_i(m + x, n + y) \tag{2.16}$$

The mean intensity image I_{M_n} is then obtained by averaging these three filtered images. The mean intensity value is normalized to range [0 1] and based on intensity value in I_{M_n} at location (x, y), the tunable parameter q is determined using

$$q = \begin{cases} tan(\frac{\pi}{C_1} I_{M_n}(x, y)) + C_2 & I_{M_n}(x, y) \geq 0.3 \\ \frac{1}{C_3} ln(\frac{1}{0.3} I_{M_n}(x, y)) + C_4 & I_{M_n}(x, y) < 0.3 \end{cases} \tag{2.17}$$

where C_1, C_2, C_3 and C_4 are determined experimentally [96].

As C_1 approaches to 2, q approaches to infinity and for values of C_1 greater than 2, q is inversely proportional to C_1. From Eq. (2.13), it is clear that for brighter pixels having intensity values nearer to 1, q must be high to bring it down. (ex-

perimentally observed that when $I_{M_n} = 1$ q should be between 5 and 6.) This limitation on q helps to decide the value of C_1, which is set to 2.255 thus, the maximum value of q can be $5.57 + C_2$. The value of C_2 is found by setting I_{M_n} to 0.5 and q to 1 to ensure that pixel with intensity value 0.5 and Gaussian mean intensity value 0.5 remains unchanged in the output image.

For the pixel having smaller values close to 0, noise becomes a serious issue while enhancement, thus for I_{M_n} less than 0.3, a logarithmic function is used to determine the value of q. The parameter C_4 is determined such that the transfer function given in Eq. (2.13) remains continuous at $I_{M_n} = 0.3$ and is set to 0.6. The value for C_3 is proportional to the minimum value of q and set experimentally equal to 60 to balance enhancement and noise reduction.

Transformation function for different values of q is presented in Fig. 2.9. As it can be seen from Fig. 2.9, extreme bright pixels having high mean intensity value nearer to 1, will be suppressed using high q values and dampen the high-intensity pixels. Extreme dark pixel with mean values nearer to zero will be enhanced with small q values and it will positively boost the low-intensity values.

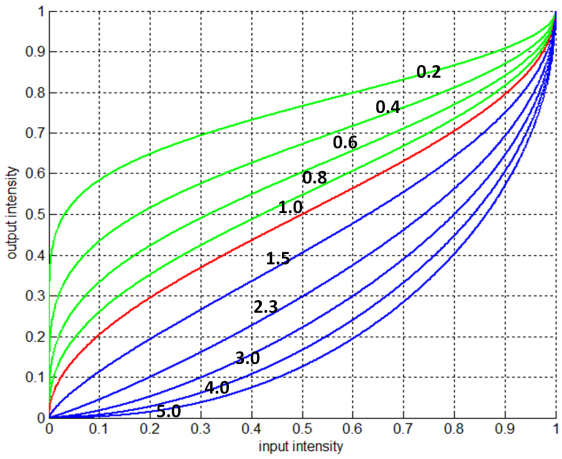


Figure 2.9: Nonlinear Inverse Sine Transformation with parameter q varying from 0.2 to 5 for intensity range [0 1]

2.4.2 DoG based Enhancement

This Preprocessing technique employs a series of operations including Gamma correction, Difference of Gaussian filtering and an equalization to enhance the

over-lit or under-lit area of an image. Gamma correction is a nonlinear transformation that replaces gray-level i with i^{γ}, for $\gamma > 0$. It enhances the local dynamic range of the image in dark regions due to under-lit conditions, at the same time suppressing the range of bright regions and highlights due to over-lit conditions.

Gamma correction alone is not capable to remove the influence of all intensity variations, such as shadowing effects. Shading induced due to the facial surface structure is useful information for recognition, whereas the spurious edges generated due to shading introduce false information for recognition. Band-pass filtering can help to retain useful information in face images while get rid of unwanted information or misleading spurious edge like features due to shadows. DoG filtering is a suitable way to achieve such a bandpass behavior. As DoG name suggests, it is basically a difference of $2D$ Gaussian filters G_{σ_1} and G_{σ_2} having different variances (the outer mask is normally $2-3$ times broader than the inner mask). The inner Gaussian G_{σ_2} is typically quite narrow (usually variance $\sigma_2 \leq 1$ pixel essentially works as high pass filter), while the outer Gaussian G_{σ_1} is $2-4$ pixels wider, depending on the spatial frequency at which low-frequency information becomes misleading rather than informative. Values for σ_1 and σ_2 are set to 1 and 2 respectively based on experiments carried out on face databases[98].

The DoG as an operator or convolution kernel defined as

$$DoG \cong G_{\sigma_1} - G_{\sigma_2} = \frac{1}{\sqrt{2\pi}}\left(\frac{1}{\sigma_1}e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{\sigma_2}e^{-\frac{x^2+y^2}{2\sigma_2^2}}\right)$$

The resulting image still typically contains extreme values produced by highlights, small dark regions etc. Following approximation is used to re-scale the gray values present in the pre-processed image.

$$I(x,y) \leftarrow \frac{I(x,y)}{(mean(min(\tau, I(x,y))^{\alpha})^{\frac{1}{\alpha}}} \tag{2.18}$$

here, $I(x,y)$ is image intensity at (x,y) location, α is a strongly compressive exponent that reduces the influence of large values, τ is a threshold used to truncate large values after the first phase of normalization and the mean is average intensity value of the image. By default, values of α and τ are set experimentally as

0.1 and 10 respectively. The resulting image is well-scaled, but it may still contain some extreme values, to reduce its effect a nonlinear mapping is done using $I(x,y) \leftarrow \tau \times \tanh(I(x,y)/\tau)$, it scales whole image in the range of $[-\tau, \tau]$.

2.4.3 Experiments and Results

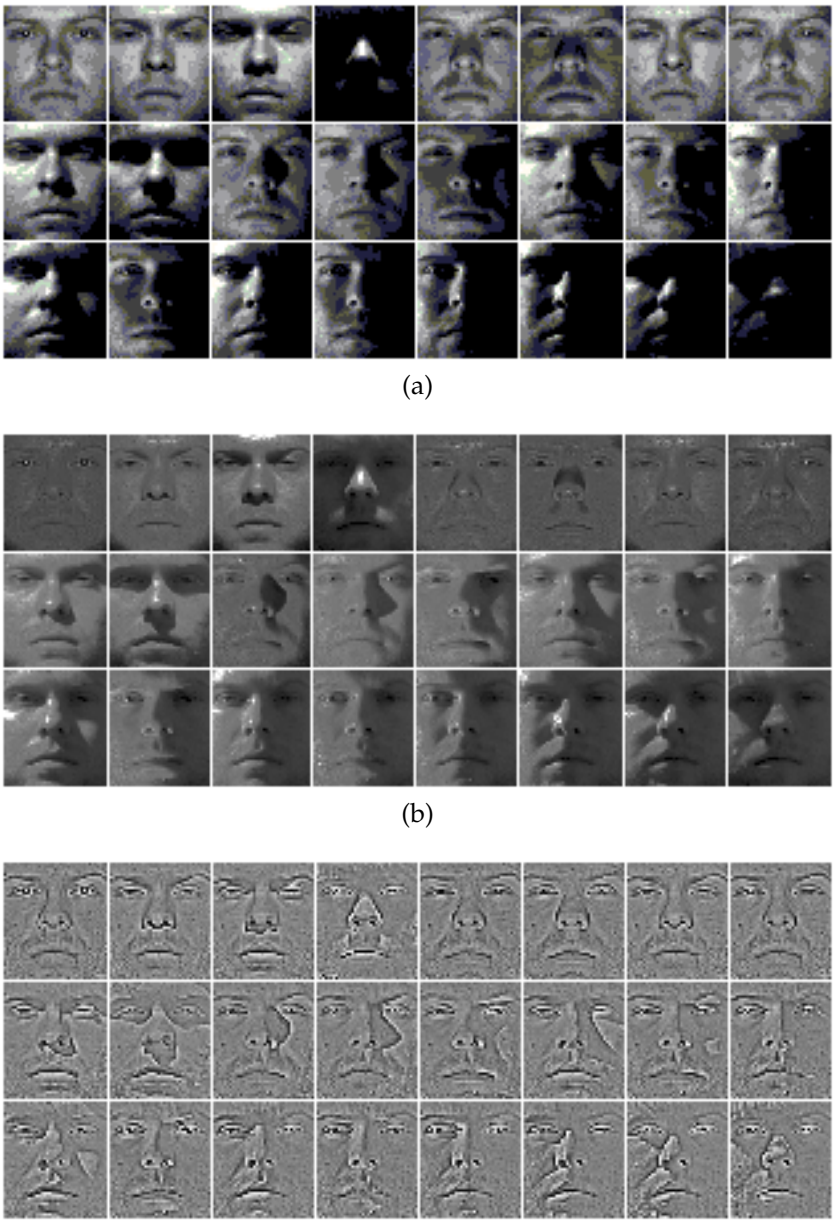
The effect of LTISN and DoG based enhancement on recognition rates using ONPP and MONPP is compared with the recognition rates without any pre-processing and reported in this section. For unbiased results, the experiments are carried out on 10 different realizations from Extended Yale-B [94] and CMU-PIE face database [95].

Extended YALE-B Face Database The experiment is performed on 2432 frontal face images of 28 subjects each with 64 illumination condition. Images are resized to 60×40 to reduce computation. Fig. 2.10 shows face images of a person with 24 different illumination direction along with pre-processed images using LTISN enhancement and DoG enhancement respectively. Fig. 2.11 (left) and Fig. 2.11 (right) shows average recognition result of LTISN and DoG based enhancement techniques respectively, combined with ONPP and MONPP with varying number of nearest-neighbors(k) values 10, 15 and 20. The best recognition result achieved with MONPP+LNIST is 99.84% at 110 dimension as listed in Table 2.4.

CMU-PIE face database The experiment is performed on 42 frontal face images of 68 subjects with varying illumination. Fig. 2.12 (left) and Fig. 2.12 (right) shows average recognition result of LTISN and DoG based enhancement techniques respectively, combined with ONPP and MONPP with varying number of nearest-neighbors(k) values 10, 15 and 20. MONPP+LNIST gives best recognition with 100% accuracy at 90 dimension as given in Table 2.4.

Table 2.4: Comparison of performance of pre-processing techniques in the light of recognition score (in %) of ONPP and MONPP

Database	Extended Yale-B		CMU PIE	
	Average	Best (at dim)	Average	Best (at dim)
ONPP	92.51	93.42(140)	94.33	96.62(130)
ONPP+LTISN	97.84	99.10(250)	98.99	100(90)
ONPP+DoG	96.67	99.34(250)	98.63	99.95(160)
MONPP	94.07	95.80(120)	95.19	97.04(110)
MONPP+LTISN	**99.53**	**99.84(110)**	**99.95**	100(70)
MONPP+DoG	99.51	99.75(40)	99.07	**100(50)**

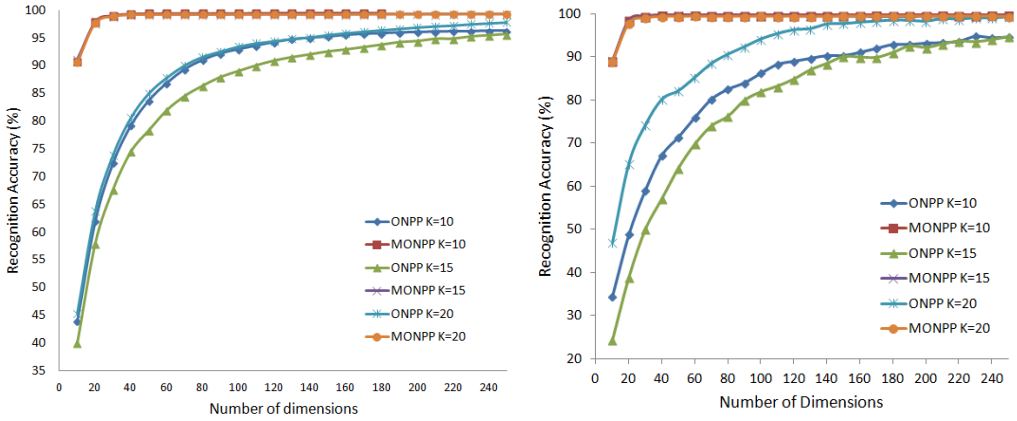


(a)

(b)

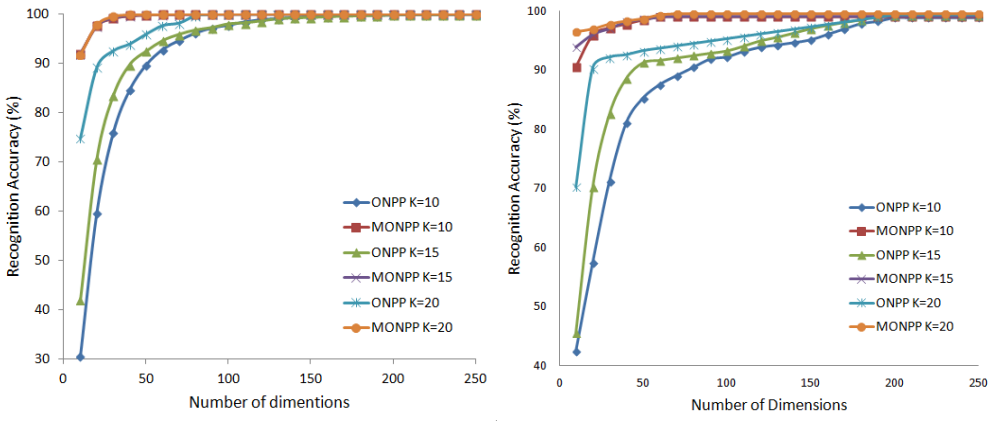
(c)

Figure 2.10: (a) Face images form Yale-B database, (b) enhanced images using LTISN, (C) enhanced images using DoG



(a) (b)

Figure 2.11: Results of recognition accuracy (in %) using (a) LNIST and (b) DoG with ONPP and MONPP on Extended Yale-B



(a) (b)

Figure 2.12: Results of recognition accuracy (in %) using (a) LNIST and (b) DoG with ONPP and MONPP on CMU-PIE

2.5 Conclusion

Subspace based methods for face recognition have been a major area of research and already proven to be more efficient. In this regard, Orthogonal Neighborhood Preserving Projection (ONPP) is assumed to handle the intrinsic non-linearity of the data manifold. The first step of ONPP deals with a linear model building within local neighborhoods. This linearity assumption may not be valid for a moderately large neighborhood. In the present work, this linear model is thus replaced by a notion of non-linearity where a piece-wise nonlinear model (z-shaped) is used instead. The suitability of the proposal is tested on non-linear synthetic data as well as a few benchmark face databases. Significant and consistent improvement in data compactness is observed for synthetic data where manifold is surely nonlinear. This signifies the suitability of the present proposal to handle non-linear manifold of the data. On the other hand, noticeable improvement is obtained for the face recognition problem. The modification suggested over existing ONPP though very simple but overall improvement in handwritten numerals recognition and face recognition results is very encouraging.

To handle illumination variations present in face images captured under uncontrolled environment, a robust pre-processing technique is highly sought. In this chapter, we adopt Locally Tuned Inverse Sine Nonlinear (LTISN) transformation and modify it to be used on gray-scale face images to nullify the illumination variations present in the face database to improve the recognition rate. The result of recognition along with LTISN as pre-processing is compared with that of another pre-processing technique called Difference of Gaussian (DoG). It is established that MONPP performs better than ONPP for face recognition. It is also observed that LTISN based enhancement followed by MONPP outperforms ONPP with or without both DoG and LTISN enhancement techniques.

CHAPTER 3

Class Similarity based Orthogonal Neighborhood Preserving Projections

In supervised settings, conventional Orthogonal Neighborhood Preserving Projections (ONPP) uses knowledge of class labels to identify the neighbors of data points. But when data points are closely placed or the classes are overlapping, such hard decision rule may not help finding a good low dimensional representation. In this chapter, we are proposing a novel neighborhood selection rule based on class similarity of data points to overcome the shortcomings of class label based strict neighborhood selection rule. It is observed that Class Similarity based ONPP (CS-ONPP) very well represents the relationship between neighbors in low dimensional subspace. The proposed scheme CS-ONPP is used to recognize face images as well as handwritten numerals images. Class Similarity based neighborhood scheme achieves same recognition performance with significantly less number of subspace dimensions as compared to conventional ONPP.

3.1 Introduction

For tasks such as face recognition, character recognition; supervised mode is better suited where class labels of training data are available and class of test data is to be predicted. In supervised ONPP, data points $\mathbf{x_i}$ and $\mathbf{x_j}$ belonging to the same class are considered neighbors to each other thus the parameter k (number of nearest neighbors) need not to be specified manually, it is automatically set to number of data samples in particular class. The definition of neighborhood and

the distance used to define this neighborhood have the paramount effect on the learned manifold. Various approaches to define this neighborhood is attempted in the past. It is proved in [99] that combining different distance matrices or dissimilarity representations can often increase the performance of individual ones. In [100], authors have proposed Supervised LLE (SLLE) that uses knowledge of class label to modify the pair-wise distances that defines neighborhood. It blindly adds a constant to the Euclidean distance, if the data points under consideration are form two different class. This approach seems to work well with high dimensional data, but performs poorly on moderately low dimensional data. In [101, 102], k-means clustering based approach is proposed to find neighbors of a data point which is again an unsupervised approach that do not consider class label knowledge to define neighborhood. Work documented in [103] proposes an adaptive neighborhood of varying size based on local linearity.

We are proposing a novel neighborhood rule which takes advantage of preprocessing stage of all dimensionality reduction techniques where high dimensional data is projected on PCA subspace. Low dimensional PCA representation of data is used with Logistic Regression (LR) [104] to find the probability of every data point to belong into a particular class. Based on these probabilities a new distance measure - Class Similarity based distance is defined and used to find nearest neighbors of data points.

3.2 Class Similarity based ONPP (CS-ONPP)

To incorporate underlying similarity between data points regardless of their class label many works have been done, in [51] authors have proposed an Enhanced Supervised LLE (ESLLE) where the Euclidean distance is simply modified by adding a constant for the pairs of data that belongs to different class, keeping others unchanged regardless of their similarity to other data points. Modified distance Δ' is given by

$$\Delta'(\mathbf{x_i}, \mathbf{x_j}) \quad = \left\|\mathbf{x_i} - \mathbf{x_j}\right\| + \alpha \max(\Delta)\mathcal{S}(i,j) \tag{3.1}$$

where, $\alpha \in [0,1]$ is a tuning parameter. $max(\Delta)$ indicates largest pair-wise distance. Class similarity function for ESLLE is only based on class labels given as

$$\mathcal{S}(i,j) = \begin{cases} 0; & \mathbf{c}(\mathbf{x_i}) = \mathbf{c}(\mathbf{x_j}) \\ 1; & \mathbf{c}(\mathbf{x_i}) \neq \mathbf{c}(\mathbf{x_j}) \end{cases} \tag{3.2}$$

Essentially, ESLLE uses knowledge of class label to identify the neighbors of data points and only to enlarge distance between data points those belong to different classes. The distance is modified with a constant increment of $\alpha \max(\Delta)$ but the scheme does not consider similarity between within class data and between class data. When data points are similar they are closely placed in the high dimensional space and thus, the classes are overlapping, such hard decision rule based on class label may not help finding a good low dimensional representation. To overcome this limitation of neighborhood finding rule, we are proposing a novel neighborhood rule inspired by [105].

Instead of claiming a data point $\mathbf{x_i}$ belonging to an unique class and modifying distance accordingly, it makes much more sense that a data point belongs to a class c with a certain probability p_c and adjust distance based on class similarity. If the data set has C classes, then for each data point we can build a C-dimensional probability vector $\mathbf{p}(\mathbf{x_i}) = [p_1(\mathbf{x_i}), p_2(\mathbf{x_i}), ..., p_C(\mathbf{x_i})]^T$ where, $p_c(\mathbf{x_i})$ is a probability of data point $\mathbf{x_i}$ belonging to c^{th} class.

We used Logistic Discrimination (LD) to find probability of each data point $\mathbf{x_i}$ belonging to class c. For given data matrix $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}]$ with known class label, the LD assumes that the logit of the probability $p_c\mathbf{x_i}$ i.e. probability of data point $\mathbf{x_i}$ belonging to a certain class c is a linear combination of features of $\mathbf{x_i}$, that can be given by

$$\log(\frac{\pi(\mathbf{x_i})}{1-\pi(\mathbf{x_i})}) = \alpha + \beta^T \mathbf{x_i}$$

or specifically for class c,

$$\pi(\mathbf{x_i}; \alpha_c, \beta_c) = \frac{\exp(\alpha_c + {\beta_c}^T \mathbf{x_i})}{1 + \exp(\alpha_c + {\beta_c}^T \mathbf{x_i})}, \quad c = 1, ..., C. \tag{3.3}$$

where $\alpha_c \in \mathcal{R}$ and $\beta_c \in \mathcal{R}^{mn}$ are parameters for class c learned on training data with class knowledge using maximum likelihood estimation.

Performing LD on high dimensional data causes huge computational burden, thus we take advantage of pre-processing performed in ONPP and use lower dimensional representation sought using PCA to find these class probabilities for each data point $\mathbf{x_i}$. Let $\mathbf{z_i}$ be a lower dimensional PCA representation of $\mathbf{x_i}$ to find probability for class c. The Eq. (3.3) becomes

$$\pi(\mathbf{x_i}) = \pi(\mathbf{z_i}; \alpha_c, \beta_c) \quad = \quad \frac{\exp(\alpha_c + {\beta_c}^T \mathbf{z_i})}{1 + \exp(\alpha_c + {\beta_c}^T \mathbf{z_i})} \tag{3.4}$$

To find probability vector $\mathbf{p}(\mathbf{x_i})$, each entry is the probability $p_c(\mathbf{x_i})$ for class c can be computed by

$$p_c(\mathbf{x_i}) \quad = \quad \frac{\pi(\mathbf{z_i}; \alpha_c, \beta_c)}{\sum_{c=1}^{C} \pi(\mathbf{z_i}; \alpha_c, \beta_c)} \tag{3.5}$$

$$\tag{3.6}$$

Note that, PCA representation $[\mathbf{z_1}, \mathbf{z_2}, ...\mathbf{z_N}]$ carries class information form corresponding data points $[\mathbf{x_1}, \mathbf{x_2}, ...\mathbf{x_N}]$.

For a pair of different data point $\mathbf{x_i}$ and $\mathbf{x_j}$, we define class-similarity by

$$\mathcal{S}(i,j) = \begin{cases} 1; & \mathbf{x_i} = \mathbf{x_j} \\ \mathbf{p}(\mathbf{x_i})^T \mathbf{p}(\mathbf{x_j}); & \mathbf{x_i} \neq \mathbf{x_j} \end{cases} \tag{3.7}$$

Using this class-similarity we define a new distance measure Δ',

$$\Delta'(\mathbf{x_i}, \mathbf{x_j}) \quad = \left\| \mathbf{x_i} - \mathbf{x_j} \right\| + \alpha \max(\Delta)(1 - \mathcal{S}(i,j)) \tag{3.8}$$

Based on this new distance, neighbors for data point $\mathbf{x_i}$ will be selected, which incorporates class information as well as similarity among neighbors. The rest of the method of finding subspace is similar to ONPP. Table 3.1 gives algorithm to find Class-Similarity based ONPP subspace.

Table 3.1: Class-Similarity based ONPP (CS-ONPP) Algorithm

	Input: Dataset $\mathbf{X} \in R^{mn \times N}$and number of reduced dimension d **Output:** Lower dimension projection $\mathbf{Y} \in R^{d \times N}$
1:	Find low dimensional representation $\mathbf{z_i}$ of data by projecting on d_{pca} dimensional space using PCA ($\mathbf{z_i} = \mathbf{V}_{pca}^{T}\mathbf{x_i}$)
2:	Use Logistic Regression on $\mathbf{z_i}$ to find class probability vector $\mathbf{p}(\mathbf{x_i})$ corresponding to data point $\mathbf{x_i}$
3:	Calculate modified distance for all data point pairs $\Delta'(\mathbf{x_i}, \mathbf{x_j})$ using Eq. (3.8)
2:	Compute NN $\mathcal{N}_{\mathbf{x_i}}$ with modified distance$\Delta'(\mathbf{x_i}, \mathbf{x_j})$
3:	Compute the weight w_{ij} for neighbors $\mathbf{x_j} \in \mathcal{N}_{\mathbf{x_i}}$ for each datapoint $\mathbf{x_i}$
4:	Compute Projection matrix $\mathbf{V} \in \mathcal{R}^{mn \times d}$whose column vectors are smallest d eigen vectors of matrix $\mathbf{M} = \mathbf{X}(\mathbf{I} - \mathbf{W})(\mathbf{I} - \mathbf{W^T})\mathbf{X^T}$
5:	Compute Embedding on lower dimension by $\mathbf{Y} = \mathbf{V^T X}$

3.3 Experiments and Results

Class-similarity based neighborhood selection approach is applied to ONPP and MONPP, Now denoted as CS-ONPP and CS-MONPP. Recognition performance of both the approaches are compared with respective conventional algorithm on some well-known face databases and handwritten numerals databases.

3.3.1 Face Recognition

Face recognition experiments using proposed methods are performed on three well-known face databases *viz* ORL face database [87] having 40 class, 10 images each with variations in facial details. UMIST face database [88] has images of 20 people, with varying samples from 19 to 48 showing different poses. CMU-PIE face database [95] consists of 38 classes with 42 images each having pose, illumination and expression variations. To maintain uniformity, all images are resized to 40×40, out of which 50% images are used for training and remaining images are used for testing. To analyze the behavior of proposed method depending on the parameters *viz* PCA dimensions d_{PCA}, the tuning parameter α and ONPP subspace dimension d, experiments are repeated with various set of (d_{PCA}, α, d), where values of d_{PCA} are taken from 2:2:10, $\alpha \in [0.25, 0.50, 0.75]$ and ONPP subspace dimensions are considered from 5:5:1600. To achieve non-biased results,

such 20 randomization for all set of (d_{PCA}, α, d) were performed on all databases. Among total $3 \times 5 \times 20 = 300$ experiments, best recognition (in%) results achieved with conventional ONPP and MONPP along with their subspace dimensions are reported in Table 3.2 with subspace dimensions required for CS based approaches to achieve the same recognition.

For all databases, it is observed that Class similarity based approaches achieve better recognition at less number of subspace dimensions compared to conventional approaches. For ORL database, CS-ONPP and CS-MONPP needs average 55 and 62 lesser subspace dimensions respectively as compared to conventional methods. For UMIST, proposed methods needs on average 100 and 85 lesser subspace dimensions. In CMU-PIE data, CS-ONPP improved dimension reduction with only a little margin, but CS-MONPP needs comparatively 700 less subspace dimensions to achieve best recognition of conventional methods.

Table 3.2: Face Image data: Best Recognition Accuracy (%) achieved using ONPP and MONPP with corresponding subspace dimensions. To achieve same recognition accuracy, subspace dimensions required in CS-ONPP and CS-MONPP are reported with corresponding tuning parameter α and PCA dimension d_{PCA}

Database	Method	Conventional Approach		Class Similarity based Approach		
		Best Recognition accuracy (%)	subspace dimensions	subspace dimensions	α	d_{PCA}
ORL	ONPP	94.00	155	95	0.25	4
	MONPP	94.50	150	85	0.25	4
UMIST	ONPP	100	145	40	0.50	4
	MONPP	100	105	20	0.50	4
CMU-PIE	ONPP	93.86	955	945	0.50	4
	MONPP	93.91	865	245	0.25	6

3.3.2 Text Recognition

Recognition experiments are also performed on handwritten numerals databases namely, MNIST [83], Gujarati [84] and Devnagari [85] having nearly 68000, 13000 and 18000 images respectively. All images were resized to 30×30. For all data bases, 100 images from each digit were randomly selected for training and remain-

ing images were used for testing. Best recognition results achieved with conventional ONPP and MONPP along with their subspace dimensions are reported in Table 3.3 with subspace dimensions required for CS based approaches to achieve the best recognition. It is observed that Class-similarity based neighborhood selection approach improves dimensionality reduction behavior by a good margin compared to class-label based neighborhood approach.

Experimentally it has been observed that in MNIST database, the best recognition can be achieved with average 45 lesser dimensions using CS-ONPP. CS-MONPP achieves best recognition with average 45 lesser dimensions. In Devnagari handwritten numerals database, to achieve best recognition performance CS-ONPP needs on average 20 less dimensions as compared to conventional method, where as CS-MONPP needs average 15 less dimensions compared to its counterpart. In Gujarati database, to reach best recognition performance of ONPP, CS-ONPP needs average 30 less subspace dimensions, where as CS-MONPP, needs average 27 less subspace dimensions compared to MONPP.

Table 3.3: Handwritten Numerals image data: Best Recognition Accuracy (%) using ONPP and MONPP with corresponding subspace dimensions. To achieve same recognition accuracy, subspace dimensions d required using CS-ONPP and CS-MONPP are reported along with tuning parameter α and PCA dimension d_{PCA}.

Database	Method	Conventional Approach		Class Similarity based Approach		
		Best Recognition accuracy (%)	subspace dimensions	subspace dimensions	α	d_{PCA}
MNIST	ONPP	86.52	90	40	0.50	4
	MONPP	87.56	70	30	0.50	4
Devnagari	ONPP	86.34	50	25	0.25	8
	MONPP	87.94	40	25	0.50	4
Gujarati	ONPP	91.76	60	40	0.50	6
	MONPP	92.08	60	30	0.50	6

It is also observed that the proposed neighborhood rule increases the performance in terms of overall recognition accuracy. Table 3.4 reports best recognition accuracy for proposed methods CS-ONPP and CS-MONPP along with parameters subspace dimensions (d), tuning parameter α and PCA dimensions d_{PCA}.

Table 3.4: Best Recognition Accuracy(%) of proposed CS-ONPP and CS-MONPP along with parameters subspace dimensions (d), tuning parameter α and PCA dimensions d_{PCA}

	CS-ONPP		CS-MONPP	
Database	Accuracy (%)	[d, α, d_{PCA}]	Accuracy (%)	[d, α, d_{PCA}]
ORL	97.50	55, 0.25, 4	98.50	30, 0.50, 4
UMIST	100	40, 0.50, 4	100	20, 0.50, 4
CMU-PIE	93.91	955, 0.75, 4	93.91	245, 0.75, 4
MNIST	88.42	30, 0.50, 4	90.21	40, 0.50, 4
Devnagari	88.89	35, 0.75, 4	90.01	35, 0.75, 2
Gujarati	92.24	55, 0.50, 6	92.42	45, 0.50, 6

3.4 Conclusion

In supervised settings conventional ONPP algorithm in the first step selects neighbors based on the knowledge of class label, which may not be the best way to select when data distribution is highly overlapping. Moreover, such class knowledge based neighbor selection rule does not exploit similarity between two data points from different classes while learning low-dimensional manifold. We propose a new neighborhood selection rule based on class similarity of data points. PCA is used to find a lower dimensional representation of training data and Logistic Discriminator is used to calculate probability for each data point to belong to a particular class, using this probability class similarity of data-point pairs is computed. A new distance measure is defined based on the class similarity between two data points. Based on this new distance measure neighbors are decided. Experiments performed on Face data and Handwritten numerals data show that Class Similarity based ONPP (CS-ONPP) outperforms conventional ONPP in recognition task and achieves superior recognition rates with compara-

tively very less number of subspace dimensions. Similar behavior is also observed in CS-MONPP. Also, CS-MONPP outperforms CS-ONPP.

CHAPTER 4

Orthogonal Neighborhood Preserving Projection with Normalization

Subspace analysis or dimensionality reduction techniques are becoming very popular for many computer vision tasks such as image recognition, image denoising etc. Some of these tasks requires image reconstruction after discarding few features in projection space. Most DR techniques deal with optimizing a cost function based on some criteria imposed on either projections of data or on the bases of the projection space. NPP and ONPP are such linear methods that preserve local linear relationship within the neighborhood, with two different constraints, normalized projection and orthogonal basis of subspace respectively. It is stated in [80] that adding orthogonality constraint to Neighborhood Preserving Projection (NPP) improves the performance empirically in terms of recognition of high-dimensional data, because orthogonality preserves distance well and balances the weight on different projection directions [106, 107]. Including orthogonality constraint in the optimization problem and relaxing normalization constraint works well in recognition problems, but normalization is also a necessary constraint for application where reconstruction with less number of dimension is desired [108, 109]. It is also established in [64] that 2D-ONPP fails at recognition task. The 2-dimensional variants of DR methods offer a great advantage in terms of less computational complexity. This motivated us to propose a variant of ONPP that gives normalized projections in the subspace having orthogonal bases - Orthogonal Neighborhood Preserving Projection with Normalization (ONPPn) for 1D and 2D data which is suitable for image recognition as well as image reconstruc-

tion application overcoming the limitations of NPP and ONPP. Following section explains proposed method in detail.

4.1 Orthogonal Neighborhood Preserving Projection with Normalization (ONPPn)

Image of size $m \times n$ is arranged into mn-dimensional vector resulting in data matrix $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, ...\mathbf{x_N}] \in \mathcal{R}^{mn \times N}$. The goal is to find a subspace with a set of orthogonal basis vectors represented by $\mathbf{V} \in \mathcal{R}^{mn \times d}$ to get normalized embeddings $\mathbf{Y} = [\mathbf{y_1}, \mathbf{y_2}, ...\mathbf{y_N}] \in \mathcal{R}^{d \times N}$ in d dimensional subspace.

The cost function for ONPPn is the same as NPP and ONPP.

$$\begin{aligned} \arg\min \mathcal{F}(\mathbf{Y}) &= \arg\min \sum_i \left\| \mathbf{y_i} - \sum_{j=1}^{k} w_{ij} \mathbf{y_j} \right\|^2 \\ \arg\min \mathcal{F}(\mathbf{V}) &= \arg\min tr[\mathbf{V^T X M X^T V}] \end{aligned} \tag{4.1}$$

with a normalization constraint (same as NPP)

$$\mathbf{v_i^T X X^T v_i} = 1, \forall i \tag{4.2}$$

with an additional orthogonality constraint (same as ONPP)

$$\mathbf{v_i}^T \mathbf{v_k} = 0, \text{for } i \neq k \tag{4.3}$$

Lagrangian expression of optimization problem for finding the first basis vector $\mathbf{v_1}$, such that it gives normalized projections, can be written using normalization constraint only,

$$\mathcal{L}(\mathbf{v_1}) = tr[\mathbf{v_1}^T \mathbf{X M X^T v_1}] - \lambda_1(\mathbf{v_1}^T \mathbf{X X^T v_1} - 1) \tag{4.4}$$

Differentiating with respect to λ_1,

$$\frac{\delta \mathcal{L}(\mathbf{v_1})}{\delta \lambda_1} = 0 \implies \mathbf{v_1 X X^T v_1} = 1 \tag{4.5}$$

Differentiating with respect to $\mathbf{v_1}$ and equating to zero will give

$$\begin{aligned} 2\mathbf{X M X^T v_1} - 2\lambda_1 \mathbf{X X^T v_1} &= 0 \\ \mathbf{X M X^T v_1} &= \lambda_1 \mathbf{X X^T v_1} \end{aligned} \tag{4.6}$$

Solution for the first basis vector turns out to be a generalized eigenvalue problem, which is the same as NPP, where $\mathbf{v_1}$ is an eigen-vector corresponding to the smallest eigenvalue λ_1.

For the rest of the basis vectors $\mathbf{v_i}$, $i = 2, 3, ..., d$, problem results into minimizing Eq. (4.1) with constraint in Eq. (4.2) with an additional constraint in Eq. (4.3). Lagrangian expression for $\mathbf{v_k}$

$$\begin{aligned} \mathcal{L}(\mathbf{v_k}) = & tr[\mathbf{v_k^T X M X^T v_k}] - \lambda_k(\mathbf{v_k^T X X^T v_k} - 1) - \\ & \mu_1 \mathbf{v_k^T v_1} - \mu_2 \mathbf{v_k^T v_2} .... - \mu_{k-1} \mathbf{v_k^T v_{k-1}} \end{aligned} \tag{4.7}$$

Equating partial derivative of $\mathcal{L}(\mathbf{v_k})$ with respect to λ_k and μ_i, to zero will give $\mathbf{v_k^T X X^T v_k} = 1$ and $\mathbf{v_k^T v_i} = 0$ (for, all $k \neq i$) respectively, and differentiating $\mathcal{L}(\mathbf{v_k})$ with respect to $\mathbf{v_k}$ and equating to zero will give

$$2\mathbf{X M X^T v_k} - 2\lambda_k \mathbf{X X^T v_k} - \mu_1 \mathbf{v_1} - \mu_2 \mathbf{v_2} ... - \mu_k \mathbf{v_k} = 0 \tag{4.8}$$

multiplying Eq. (4.8) with $\mathbf{v_k}^T$

$$\begin{aligned} 2\mathbf{v_k^T X M X^T v_k} - 2\lambda_k \mathbf{v_k^T X X^T v_k} &= 0 \\ \lambda_k &= \frac{\mathbf{v_k^T X M X^T v_k}}{\mathbf{v_k^T X X^T v_k}} \end{aligned} \tag{4.9}$$

multiplying Eq. (4.8) with $\mathbf{v_i^T}[\mathbf{X X^T}]^{-1}$, $i = 1, 2, ..., k-1$, will result in set of

$k-1$ equations.

$$\begin{aligned}
\mu_1 \mathbf{v_1^T}[\mathbf{XX^T}]^{-1}\mathbf{v_1} + \mu_2 \mathbf{v_1^T}[\mathbf{XX^T}]^{-1}\mathbf{v_2} + ... + \mu_{k-1}\mathbf{v_1^T}[\mathbf{XX^T}]^{-1}\mathbf{v_{k-1}} \\
= 2\mathbf{v_1^T}[\mathbf{XX^T}]^{-1}[\mathbf{XMX^T}]\mathbf{v_k} \\
\mu_1 \mathbf{v_2^T}[\mathbf{XX^T}]^{-1}\mathbf{v_1} + \mu_2 \mathbf{v_2^T}[\mathbf{XX^T}]^{-1}\mathbf{v_2} + ... + \mu_{k-1}\mathbf{v_2^T}[\mathbf{XX^T}]^{-1}\mathbf{v_{k-1}} \\
= 2\mathbf{v_2^T}[\mathbf{XX^T}]^{-1}[\mathbf{XMX^T}]\mathbf{v_k} \\
\vdots \\
\mu_1 \mathbf{v_{k-1}^T}[\mathbf{XX^T}]^{-1}\mathbf{v_1} + \mu_2 \mathbf{v_{k-1}^T}[\mathbf{XX^T}]^{-1}\mathbf{v_2} + ... + \mu_{k-1}\mathbf{v_{k-1}^T}[\mathbf{XX^T}]^{-1}\mathbf{v_{k-1}} \\
= 2\mathbf{v_{k-1}^T}[\mathbf{XX^T}]^{-1}[\mathbf{XMX^T}]\mathbf{v_k}
\end{aligned}$$

in general, i^{th} equation can be written as

$$\sum_{j=1}^{k-1} \mu_j \mathbf{v_i^T}[\mathbf{XX^T}]^{-1}\mathbf{v_j} = 2\mathbf{v_i^T}[\mathbf{XX^T}]^{-1}[\mathbf{XMX^T}]\mathbf{v_k} \tag{4.10}$$

writing the set of $k-1$ equations of Eq. (4.10) in a matrix form,

$$[\mathbf{U^{(k-1)}}]\text{-}^{(\mathbf{k-1})} = 2[\mathbf{V^{(k-1)}}]^{\mathbf{T}}[\mathbf{XX^T}]^{-1}[\mathbf{XMX^T}]\mathbf{v_k} \tag{4.11}$$

where, $\boldsymbol{\mu}^{(k-1)} = [\mu_1, \mu_2, ...\mu_{k-1}]^T$, $\mathbf{V^{(k-1)}} = [\mathbf{v_1}, \mathbf{v_2}, ...\mathbf{v_{k-1}}]$ and
$\mathbf{U^{(k-1)}} = [\mathbf{V^{(k-1)}}]^{\mathbf{T}}[\mathbf{XX^T}]^{-1}[\mathbf{V^{(k-1)}}]$

Lagrange's k^{th} multiplier $\boldsymbol{\mu}^{(k-1)}$ can be obtained by multiplying Eq. (4.11) with $[\mathbf{U^{(k-1)}}]^{-1}$

$$\boldsymbol{\mu}^{(k-1)} = 2[\mathbf{U^{(k-1)}}]^{-1}[\mathbf{V^{(k-1)}}]^{\mathbf{T}}[\mathbf{XX^T}]^{-1}[\mathbf{XMX^T}]\mathbf{v_k} \tag{4.12}$$

rewriting Eq. (4.8) in matrix form

$$2[\mathbf{XMX^T}]\mathbf{v_k} - 2\lambda_k[\mathbf{XX^T}]\mathbf{v_k} - [\mathbf{V^{(k-1)}}]\boldsymbol{\mu}^{(k-1)} = 0$$

multiplying with $[\mathbf{X}\mathbf{X}^\mathbf{T}]^{-\mathbf{1}}$

$$2[\mathbf{X}\mathbf{X}^\mathbf{T}]^{-\mathbf{1}}[\mathbf{X}\mathbf{M}\mathbf{X}^\mathbf{T}]\mathbf{v_k} - 2\lambda_k\mathbf{v_k} - [\mathbf{X}\mathbf{X}^\mathbf{T}]^{-\mathbf{1}}[\mathbf{V}^{(\mathbf{k-1})}]\boldsymbol{\mu}^{(k-1)} = 0$$

replacing $\boldsymbol{\mu}^{(k-1)}$with its value from Eq. (4.12)

$$\begin{aligned}
&2[\mathbf{X}\mathbf{X}^\mathbf{T}]^{-\mathbf{1}}[\mathbf{X}\mathbf{M}\mathbf{X}^\mathbf{T}]\mathbf{v_k} - 2\lambda_k\mathbf{v_k} \\
&\quad - 2[\mathbf{X}\mathbf{X}^\mathbf{T}]^{-\mathbf{1}}[\mathbf{V}^{(\mathbf{k-1})}][\mathbf{U}^{(\mathbf{k-1})}]^{-\mathbf{1}}[\mathbf{V}^{(\mathbf{k-1})}]^\mathbf{T}[\mathbf{X}\mathbf{X}^\mathbf{T}]^{-\mathbf{1}}[\mathbf{X}\mathbf{M}\mathbf{X}^\mathbf{T}]\mathbf{v_k} = 0 \\
&\{\mathbf{I} - \mathbf{A}\}[\mathbf{X}\mathbf{X}^\mathbf{T}]^{-\mathbf{1}}[\mathbf{X}\mathbf{M}\mathbf{X}^\mathbf{T}]\mathbf{v_k} = \lambda\mathbf{v_k} \qquad (4.13) \\
&\text{where, } \mathbf{A} = [\mathbf{X}\mathbf{X}^\mathbf{T}]^{-\mathbf{1}}[\mathbf{V}^{(\mathbf{k-1})}][\mathbf{U}^{(\mathbf{k-1})}]^{-\mathbf{1}}[\mathbf{V}^{(\mathbf{k-1})}]^\mathbf{T}
\end{aligned}$$

Eq. (4.13) can be used to iteratively find orthogonal basis $\mathbf{V}$ such that the projection of data $\mathbf{Y}(= \mathbf{V}^\mathbf{T}\mathbf{X})$ will result in normalized projection. The algorithm is summarized in Table 4.1. It is important to mention that the derivation is based on [110] that solves eigenvalue problem for each basis that handles orthogonality as well as normalization constraint simultaneously.

An alternate solution, as suggested in [111] and [106] can also be derived where projection matrix is recursively updated in ortho-complemented space. Moreover, in the solution proposed in [111] and [110], a basis vector is randomly initialized and then iteratively updated until convergence, so for d dimensional subspace the computation of projection matrix $\mathbf{V}$ involves d recursive procedures for a certain number of iterations until convergence is achieved for each basis. The proposed approach on the other hand computes basis vector as a closed form solution for each basis vector.

4.2 2-dimensional ONPPn (2D-ONPPn)

The main advantage offered by 2-dimensional variants of dimensionality reduction methods is computational efficiency, 1-dimensional techniques involve eigenvalue problem of order mn, which requires huge computational capacity and are time consuming, whereas 2-dimensional approaches involve eigenvalue problem of order m. Also the data matrix in 1D is created by rearranging 2D image into

Table 4.1: The algorithm to find ONPPn embedding

	INPUT: $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, ...\mathbf{x_N}]$
	OUTPUT: Projection matrix $\mathbf{V}$
	$\mathbf{Y} = [\mathbf{y_1}, \mathbf{y_2}, ...\mathbf{y_N}]$
1:	Find nearest neighbors $\mathcal{N}_{\mathbf{x_i}}$ for each data point $\mathbf{x_i}$
2:	Compute weight matrix $\mathbf{W}$
3:	Compute matrix $\mathbf{M} = (\mathbf{I} - \mathbf{W})^{\mathbf{T}}(\mathbf{I} - \mathbf{W})$
4:	Find $\mathbf{v_1}$ by solving $\mathbf{XMX^T v_1} = \lambda_1 \mathbf{XX^T v_1}$
5:	**for** $k = 2$ to d
	$\mathbf{V^{(k-1)}} = [\mathbf{v_1}, \mathbf{v_2}, ...\mathbf{v_{k-1}}]$
	$\mathbf{U^{(k-1)}} = [\mathbf{V^{(k-1)}}]^{\mathbf{T}}[\mathbf{XX^T}]^{-1}[\mathbf{V^{(k-1)}}]$
	$\mathbf{A} = [\mathbf{XX^T}]^{-1}[\mathbf{V^{(k-1)}}][\mathbf{U^{(k-1)}}]^{-1}[\mathbf{V^{(k-1)}}]^{\mathbf{T}}$
	solve for $\mathbf{v_k}$ by solving $\{\mathbf{I} - \mathbf{A}\}[\mathbf{XX^T}]^{-1}[\mathbf{XMX^T}]\mathbf{v_k} = \lambda_k \mathbf{v_k}$
6:	Project Data $\mathbf{X}$ on projection matrix $\mathbf{V}$ to achieve d-dimensional embeddings $\mathbf{Y}$

1D vector, thus losing inter-pixel similarity in at-least one direction of the image, while in 2-dimensional methods keeps inter-pixel dependency in image as it is [5]. Moreover, 2D approaches overcome the hurdle of under-sampled size problem which indicates that the number of samples is less than dimensions of data. In 1D methods, dimensional of data is mn which is less than number of samples N available for finding subspace, thus requires PCA to obtain initial subspace, whereas data in 2D approaches is arranged in such a way that it takes care of small-sampled size problem and do not require any pre-processing.

In 2D approaches, instead of rearranging $m \times n$ image into mn-dimensional vector, images are rearranged side by side as a matrix, resulting in data matrix $\mathbf{X} = [\mathbf{X_1}, \mathbf{X_2}, ...\mathbf{X_N}] \in \mathcal{R}^{mn \times N}$, where $\mathbf{X_i}$ is i^{th} image. The goal is to find a subspace with a set of orthogonal basis vectors $\mathbf{V} \in \mathcal{R}^{m \times d}$ to get normalized embedding $\mathbf{Y} = [\mathbf{Y_1}, \mathbf{Y_2}, ...\mathbf{Y_N}] \in \mathcal{R}^{d \times nN}$.

The cost function for 2D-ONPP will be the same as Eq. (4.1) with only a different arrangement for 2D data,

$$\begin{aligned} \arg\min \mathcal{F}(\mathbf{V}) &= \arg\min tr[\mathbf{V^T X}[(\mathbf{I} - \mathbf{W})^{\mathbf{T}}(\mathbf{I} - \mathbf{W})] \otimes \mathbf{I_n}]\mathbf{X^T V}] \\ &= \arg\min tr[\mathbf{V^T X}(\mathbf{M} \otimes \mathbf{I_n})\mathbf{X^T V}] \end{aligned} \tag{4.14}$$

First basis vector can be computed using generalized eigenvalue problem

$$\mathbf{X}(\mathbf{M} \otimes \mathbf{I_n})\mathbf{X^T}\mathbf{v_1} = \lambda_1 \mathbf{X}\mathbf{X^T}\mathbf{v_1} \tag{4.15}$$

The rest of the basis vectors can be found by iteratively solving ordinary eigenvalue problem using the equation

$$\{\mathbf{I} - \mathbf{A}\}[\mathbf{X}\mathbf{X^T}]^{-1}[\mathbf{X}(\mathbf{M} \otimes \mathbf{I_n})\mathbf{X^T}]\mathbf{v_k} = \lambda_k \mathbf{v_k} \tag{4.16}$$
$$\text{where, } \mathbf{A} = [\mathbf{X}\mathbf{X^T}]^{-1}[\mathbf{V^{(k-1)}}][\mathbf{U^{(k-1)}}]^{-1}[\mathbf{V^{(k-1)}}]^{\mathbf{T}}$$

Table 4.2: Comparison between the optimization problems of NPP, ONPP and ONPPn and its 2-dimensional variants. Here, $\mathbf{X} \in \mathcal{R}^{mn \times N}$ is data matrix for one-dimensional methods and $\mathbf{X_2} \in \mathcal{R}^{m \times nN}$ is data matrix for two-dimensional methods, where N is number of images and $m \times n$ is image size.

Method	Cost Function (min)	Constraints	Eigenvalue Problem
NPP	$Tr[\mathbf{V^T X M X^T V}]$	$\mathbf{V^T X X^T V} = \mathbf{I_d}$	generalized
ONPP	$Tr[\mathbf{V^T X M X^T V}]$	$\mathbf{V^T V} = \mathbf{I_d}$	ordinary
ONPPn	$Tr[\mathbf{V^T X M X^T V}]$	$\mathbf{V^T V} = \mathbf{I_d}$ and $\mathbf{v_i^T X X^T v_i} = 1, \forall i$	iterative-solution (both)
2D-NPP	$Tr[\mathbf{V^T X_2}(\mathbf{M} \otimes \mathbf{I_n})\mathbf{X_2}^\mathbf{T}\mathbf{V}]$	$\mathbf{V^T X_2 X_2}^\mathbf{T}\mathbf{V} = \mathbf{I_d}$	generalized
2D-ONPP	$Tr[\mathbf{V^T X_2}(\mathbf{M} \otimes \mathbf{I_n})\mathbf{X_2}^\mathbf{T}\mathbf{V}]$	$\mathbf{V^T V} = \mathbf{I_d}$	ordinary
2D-ONPPn	$Tr[\mathbf{V^T X_2}(\mathbf{M} \otimes \mathbf{I_n})\mathbf{X_2}^\mathbf{T}\mathbf{V}]$	$\mathbf{V^T V} = \mathbf{I_d}$ and $\mathbf{v_i^T X_2 X_2}^\mathbf{T}\mathbf{v_i} = 1, \forall i$	iterative-solution (both)

4.3 Experiments and Results

The performance of proposed algorithm ONPPn and its 2-dimensional variant are tested under two different applications - recognition and reconstruction. This section reports results of experiments performed using conventional method of weighing neighbors and the modified weights proposed in chapter 2. This article also compares these algorithm with NPP with modified weighing scheme now onward named as Modified NPP (MNPP), Modified ONPP (MONPP) and Modified ONPPn (MONPPn). This study has included extensive experiments on various well-known face database and handwritten numerals databases. The ex-

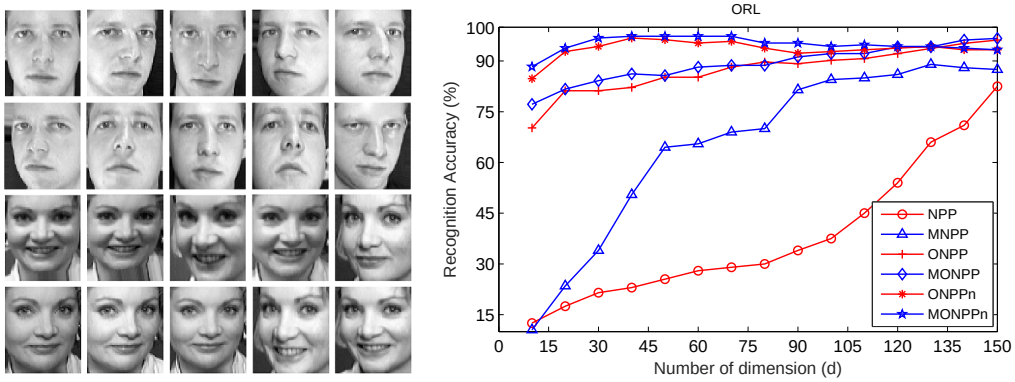
periments are performed in supervised settings and NN is used as a classifier for its simplicity.

4.3.1 Face Recognition

Recognition experiment is performed for all six methods discussed here on four widely used face databases. To maintain uniformity among databases images were resized to 32×32. To avoid any bias due to random selection of training set, 20 iterations were performed, average results and best results for all approaches are reported in Table 4.3. Nearest Neighbor (NN) is used as a classifier for its simplicity.

ORL: The ORL database [87] contains gray scale face images of 40 individuals, 10 images each with variations in facial details (with or without glasses), facial expressions (open/closed eyes smiling/not smiling) and poses. Fig. 4.1 (a) shows one sample subject from ORL database. 200 images from 40 individuals (i.e. 5 from each) are selected randomly for training and the remaining 200 images are used as test set to evaluate the proposed technique in comparison with existing methods. Fig. 4.1 (b) shows recognition rates using varying number of dimensions (d) for all six methods. NPP and MNPP start with low recognition rates with a small number of dimensions and recognition accuracy increases with increasing number of dimensions. ONPP, MONPP, ONPPn and MONPPn have respectable recognition rates, but the proposed methods ONPPn and MONPPn outperform the other four methods with a huge margin, and methods with modified weights perform comparatively better than the conventional methods having linear weights.

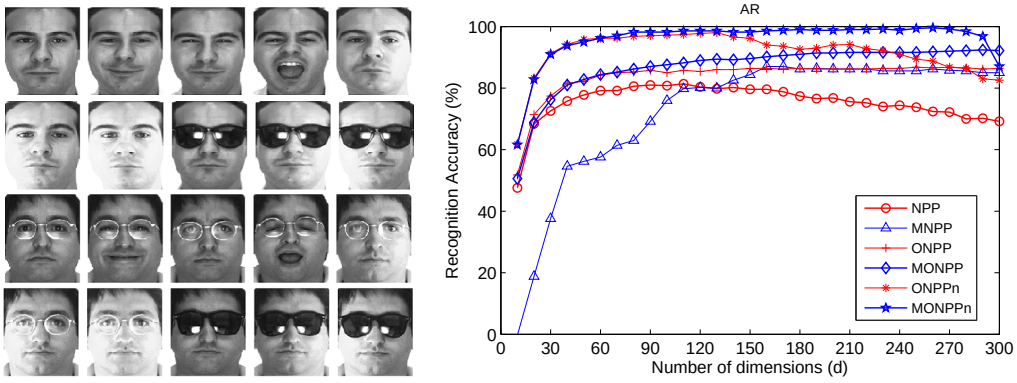
AR: AR face database [86] contains nearly 4000 color face images corresponding to 126 people including 70 men and 56 women. Images feature frontal view faces with variations in facial expressions, illumination conditions, and occlusions such as glasses and scarves. The pictures were taken in two sessions, separated by two weeks time. The same pictures were taken in both sessions. We have chosen 20



(a) (b)

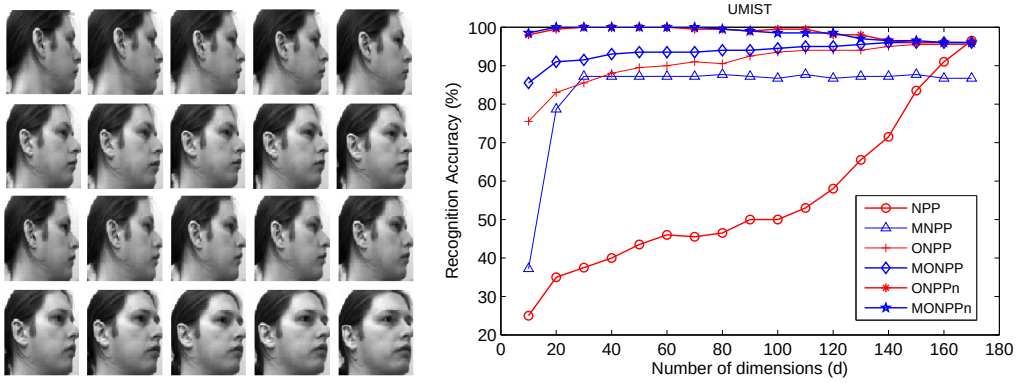
Figure 4.1: ORL face data: (a) Sample images of one sample having variations in pose and facial expressions. (b) Recognition accuracy (%) with varying number of dimensions (d)

images from each class for experiments (not considering 6 images occluded with scarves) and resized them to 32×32, as shown in Fig. 4.2 (a), out of which 50% images were used as a training set and remaining images were used for testing. The recognition rates with varying number of dimensions (d) is shown in Fig. 4.2 (b). Performance of ONPPn and MONPPn surpasses other methods. Average and Best recognition rates are reported in Table 4.3. Unlike ORL data, for AR database, MNPP lags the performance of NPP, but MONPP and MONPPn performs as good as ONPP and ONPPn, respectively.



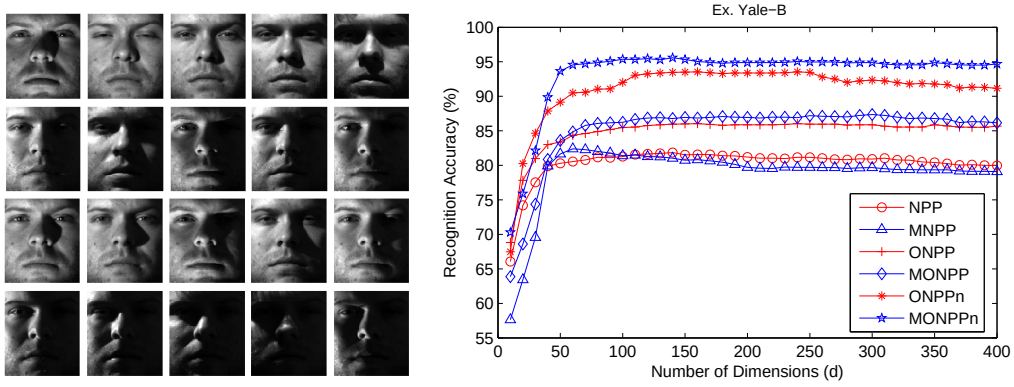
(a) (b)

Figure 4.2: AR face database: (a) Sample images of two samples with variations from left to right: natural expression, smile, anger, scream, left light on, right light on, all lights on, wearing sunglasses, wearing sunglasses and left light on, wearing sunglasses and right light on. (b) Recognition Accuracy (%) with varying number of dimensions (d)



(a) (b)

Figure 4.3: UMIST face database: (a) Sample images of a person with varying poses (b) Recognition Accuracy (%) with varying number of dimensions (d)



(a) (b)

Figure 4.4: Extended Yale-B database: (a) Sample images of frontal faces from YaleB database. YaleB database has wide variation in illumination directions. (b) Recognition Accuracy (%) with varying number of dimensions (d)

UMIST: UMIST database [88] have images of 20 people, with varying samples from 19 to 48, having different poses. For experiments, first 20 images from each class are taken as shown in Fig. 4.3 (a), out of which 50% are used for training and the rest for the testing. Fig. 4.3 (b) shows the recognition rates for all six methods with varying dimensions (d). Recognition rate of NPP is poor compared to the other five methods for UMIST database, but recognition rates using ONPPn and MONPPn outperforms MNPP, ONPP and MONPP with a good margin. Average and best recognition rates for UMIST database are reported in Table 4.3.

The extended Yale-B: The extended Yale-B database [94] have face images of 28 subjects with 9 poses and 64 varying illumination conditions. For experiments 2432 frontal face images are chosen and resized to 32×32. Randomly 50% images from each subject were taken as a training set, remaining images were used as a test set. Fig. 4.4 shows randomly chosen sample images of a person from Extended Yale-B database. Recognition accuracy achieved with varying number of dimensions (d) is shown in Fig. 4.4. Table 4.3 reports average and best recognition achieved with all methods. Bold letters in the Table indicate best performance among six methods compared. MONPPn outperforms other methods with considerable margin.

Table 4.3: Comparison of performance of NPP, MNPP, ONPP, MONPP, ONPPn and MONPPn on popular face databases in terms of recognition accuracy (in %) with corresponding subspace dimensions (d)

Data	Method	NPP	MNPP	ONPP	MONPP	ONPPn	MONPPn
AR	Average	85.53	92.60	92.77	94.47	98.07	**98.67**
	Best	87.2	89.33	94.00	96.41	98.60	**99.20**
	(at dim d)	(110)	(160)	(120)	(130)	(120)	(110)
ORL	Average	86.50	87.00	94.20	94.37	96.93	**98.91**
	Best	89.00	89.50	95.20	96.70	98.20	**99.00**
	(at dim d)	(155)	(145)	(155)	(155)	(40)	(40)
UMIST	Average	97.33	86.53	96.50	97.16	99.00	**99.00**
	Best	98.00	87.50	98.00	98.50	99.50	**100.00**
	(at dim d)	(170)	(140)	(165)	(150)	(45)	(50)
Ex.YaleB	Average	80.42	81.58	84.80	87.01	92.41	**94.20**
	Best	81.85	82.38	86.04	87.20	93.54	**95.57**
	(at dim d)	(140)	(60)	(160)	(170)	(120)	(130)

Recognition in 2D-variants : Fig. 4.5 shows the recognition accuracy achieved with 2D variants of NPP, as proved in [64] 2D orthogonal variants fails at recognition tasks, compared to its 1D variant and normalized variants. Proposed technique tries to overcome the limitation of 2D orthogonal variant. As shown in Fig. 4.5 the recognition accuracy achieved with 2D-ONPPn and 2D-MONPPn is improved greatly as compared to 2D-ONPP and 2D-MONPP, but the performance is still inferior to 2D-NPP and 2D-MNPP. The similar behavior is observed on other face and handwritten numerals databases. Hence, only results on AR database is

reported here.

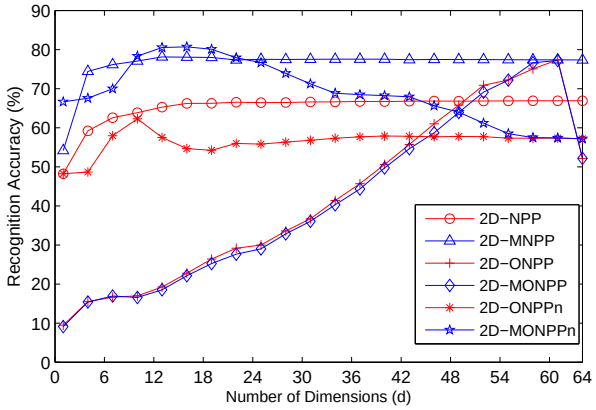


Figure 4.5: Recognition accuracy (%) achieved using 2D variants on AR face database

4.3.2 Text Recognition

To compare performance of ONPPn and MONPPn with other four methods, handwritten numeral databases in English and three different Indian languages *wiz* Gujarati, Bangla and Devnagari are used. To maintain uniformity across all databases, images in each database is resized to 20×20. Average and best recognition rates are reported in Table 4.4.

MNIST Digit Database : The MNIST digit database [83] contains nearly 68,000 images of digits of size 28×28. For this experiment, images are resized to 20×20 and nearly 100 images out of each class were selected randomly as training set and remaining images are used for testing. Recognition accuracy with varying number of dimensions (d) is shown in Fig. 4.6 (a). Due to the presence of huge variation in the data, it is expected that methods with modified weight will perform better than methods with linear weights. For handwritten numerals data NPP and MNPP fails at recognition with very poor performance. ONPP and MONPP performs better than NPP and MNPP but proposed methods outperform all the four existing methods with large margin.

Gujarati Numerals Database : The Gujarati Numerals database [84] has nearly 1300 images for each class, out of which 100 were randomly selected for training

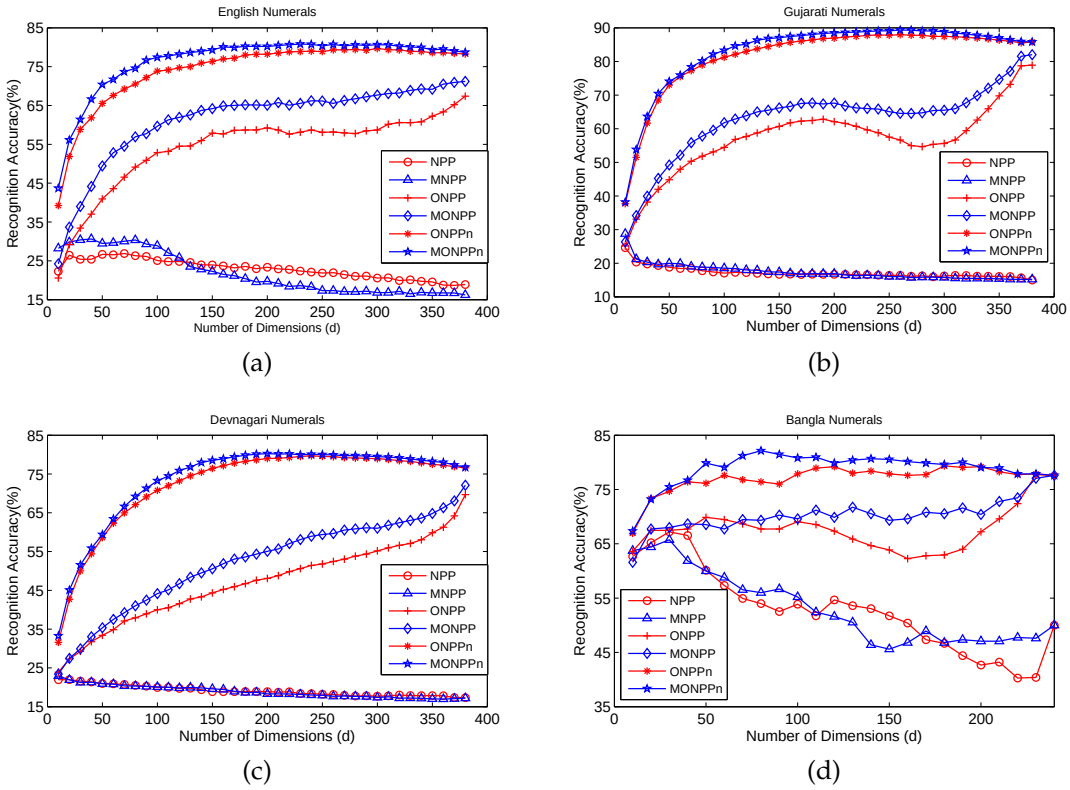


Figure 4.6: Recognition accuracy with respect to reduced number of dimensions(d) for handwritten numerals databases (a) English MNIST (b) Gujarati (c) Devnagari (d) Bangla

and the remaining were used for testing. Recognition rate of all methods with varying number of dimensions d are presented in Fig. 4.6 (b), average and best results are reported in Table 4.4. As seen in MNIST data, for Gujarati numerals also NPP and MNPP fail in recognition, whereas recognition rate using ONPP and MONPP increases with more number of dimensions, but it is still poor compared to the proposed ONPPn and MONPPn.

Devnagari Numerals Database : Another database for Indian script Devnagari was used for recognition experiment. Handwritten Devnagari database [85] have approximately 1800 images for each class. Fig. 4.6 (c) shows average recognition accuracy with varying number of dimensions (d), best and average results are recorded in Table 4.4. The performance of all methods are similar to that on the Gujarati and MNIST data, proposed methods surpasses the performance of other existing methods used here.

Bangla Numerals Database : Unlike other three numerals databases used here, Bangla [85] have small number of samples accounting to 50 for each class. Out of 50 images of each class 25 images were randomly chosen for training, remaining images were used for testing. Fig. 4.6 (d) shows recognition accuracy for all methods with varying number of dimensions (d), as it can be noticed that performance of NPP and MNPP is slightly better in case of Bangla numerals compared to other data, but it is still poor compared to other methods. Proposed methods give higher recognition accuracy compared to other four methods. Table 4.4 reports average and best recognition accuracy along with dimensions for Bangla numerals.

Table 4.4: Comparison of performance of NPP, MNPP, ONPP, MONPP, ONPPn and MONPPn on handwritten numerals databases in terms of recognition accuracy (in %) with corresponding subspace dimensions (d)

Database	Method	NPP	MNPP	ONPP	MONPP	ONPPn	MONPPn
English	Average	27.76	31.85	67.41	71.31	79.45	**80.73**
	Best	28.50	33.38	70.24	74.83	83.94	**84.99**
	(at dim d)	(20)	(25)	(380)	(380)	(270)	(215)
Gujarati	Average	27.89	28.78	78.72	81.55	88.37	**89.91**
	Best	29.19	29.01	82.74	84.48	88.95	**90.92**
	(at dim d)	(5)	(5)	(370)	(370)	(235)	(230)
Devnagari	Average	22.17	22.99	69.67	72.12	79.66	**80.22**
	Best	23.19	25.29	71.79	74.05	80.07	**80.91**
	(at dim d)	(15)	(10)	(380)	(380)	(245)	(200)
Bangla	Average	67.20	65.80	77.80	77.20	79.40	**82.20**
	Best	69.40	67.60	79.20	80.40	81.20	**85.60**
	(at dim d)	(30)	(25)	(230)	(230)	(180)	(80)

4.3.3 Image Reconstruction

The neighborhood preserving property of NPP can be quantified as $E(\mathbf{v_k}) = \frac{\mathbf{v_k^T X M X^T v_k}}{\mathbf{v_k^T X X^T v_k}}$ which is essentially k^{th} eigenvalue for given data. As shown in Fig. 4.7, eigenvalue of ONPPn is consistently lower than that of NPP and ONPP, thus it is expected that ONPPn will preserve neighborhood relationship better than NPP and ONPP. Moreover, proposed method gives an orthogonal subspace which is beneficial to reconstruct data point after discarding some of the dimensions in the

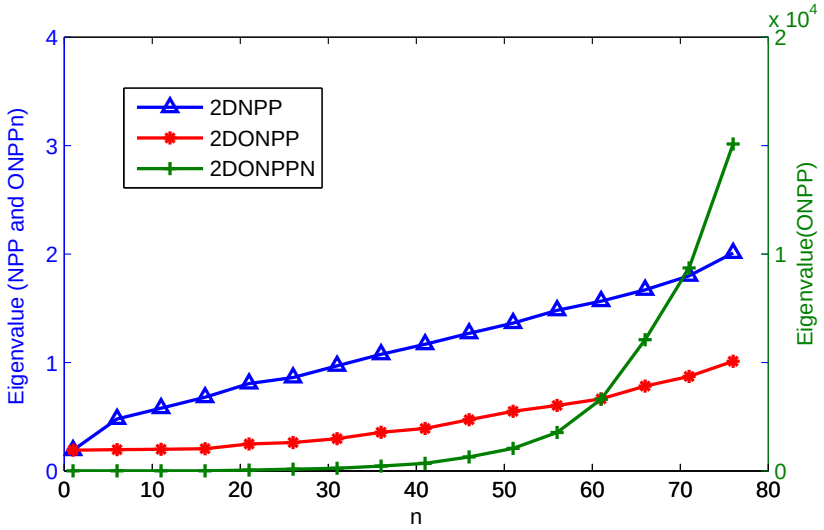


Figure 4.7: Eigenvalues of NPP, ONPP and ONPPn. Eigenvalues of ONPPn are consistently lower than that of NPP, which indicates that ONPPn has more neighborhood preserving power than its non-orthogonal variant NPP. Eigenvalues of ONPP are larger by nearly inthe order 10^4 than that of NPP and ONPPn.

projection space, because in case of orthogonal basis $\mathbf{V}$, $\mathbf{V}^{\dagger} = (\mathbf{V}^{\mathbf{T}}\mathbf{V})^{-1}\mathbf{V}^{\mathbf{T}} = \mathbf{V}^{\mathbf{T}}$. As basis vectors of 2D-NPP are non-orthogonal, reconstruction after discarding some of the dimensions in projection space involves calculation of Moore-Penrose (pseudo)-inverse, that makes it computationally complex to reconstruct the data. 2D-ONPP and 2D-ONPPn both gives an orthogonal projection space, so can be easily used for reconstruction of image data. The quality of reconstruction using d dimensions can be measured using the reconstruction error $Er_i = \left\|\mathbf{x_i} - \mathbf{v_d}(\mathbf{v_d}^{\mathbf{T}}\mathbf{x_i})\right\|^2$, for orthogonal basis and $Er_i = \left\|\mathbf{x_i} - \mathbf{v_d^{\dagger}}^{\mathbf{T}}(\mathbf{v_d}^{\mathbf{T}}\mathbf{x_i})\right\|^2$, for non-orthogonal basis. Here, $\mathbf{V_d}$ contains first d eigen-vectors of respective projection matrix. Fig. 4.8 shows reconstruction error using $d = \{10, 20, ..., 80(all)\}$ dimensions for 2D-NPP (2D-MNPP), 2D-ONPP (2D-MONPP) and 2D-ONPPn (2D-MONPPn) methods. Fig. 4.9 shows that 2D-NPP (2D-MNPP) can reconstruct data, but involves huge calculations, whereas 2D-ONPP (2D-MONPP) can reconstruct an image but with very high error rates at low dimensions, satisfactory results can not be achieved even at 90% of dimensions are retained in projection space. In comparison to 2D-ONPP (2D-MONPP), 2D-ONPPn (2D-MONPPn) excels at reconstruction, as can be seen from Fig. 4.8, the structure of face is retained only with nearly 60% of dimensions, the reconstruction error consistently reduces and fall to zero when no dimensions are

discarded in projection space (refer to Table 4.5).

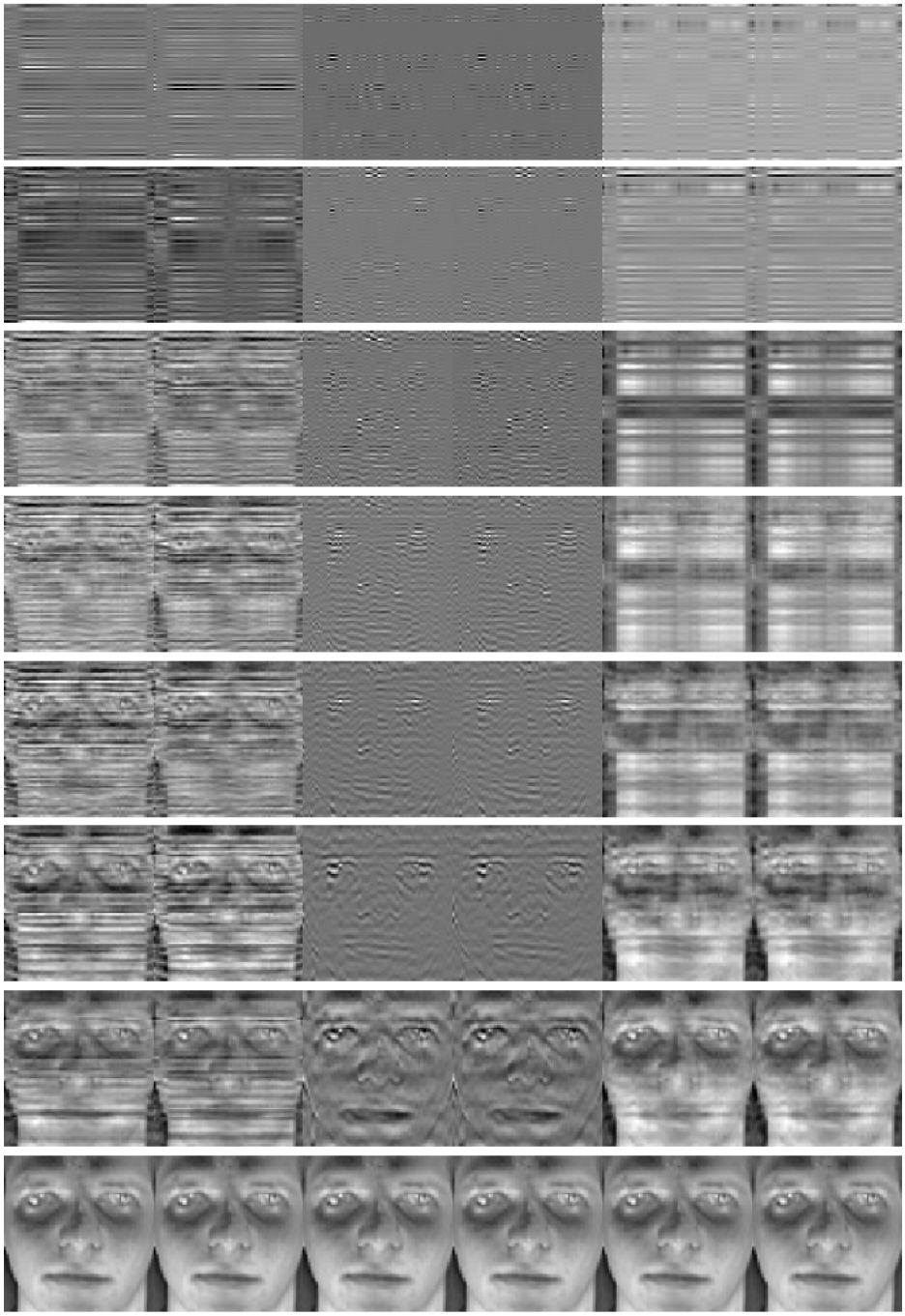


Figure 4.8: Reconstruction of a face image from ORL database using (from top to bottom) 10,20,30,40,50,60,70 and 80 dimensions from projection space using (a) 2D-NPP (b) 2D-MNPP (c) 2D-ONPP (d) 2D-MONPP (e) 2D-ONPPn and (f) 2D-MONPPn

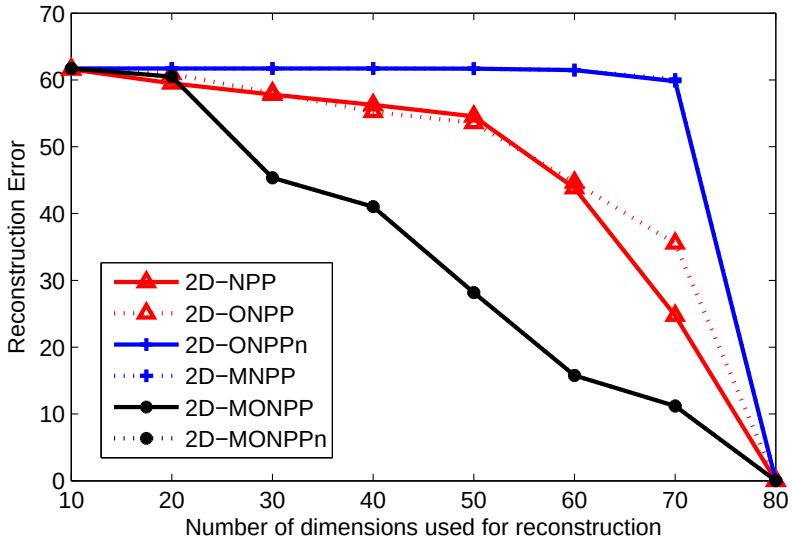


Figure 4.9: Reconstruction error calculated for face image shown in Fig. 4.8 with varying number of dimensions retained in the projection space using 2D-NPP, 2D-MNPP, 2D-ONPP, 2D-MONPP, 2D-ONPPn and 2D-MONPPn

Table 4.5: Reconstruction error of 2D variants 2D-NPP, 2D-MNPP, 2D-ONPP, 2D-MONPP, 2D-ONPPn and 2D-MONPPn for face image shown in 4.8

dim(d)	10	20	30	40	50	60	70	80
2D-NPP	: 61.58	59.49	57.80	56.27	54.55	43.81	24.74	5.07E-12
2D-MNPP	: 61.64	60.89	58.08	55.24	53.58	44.66	35.55	4.27E-12
2D-ONPP	: 61.72	61.72	61.72	61.71	61.68	61.47	59.81	1.05E-13
2D-MONPP	: 61.72	61.72	61.72	61.71	61.68	61.48	60.07	9.31E-14
2D-ONPPn	: 61.72	60.49	45.34	41.02	28.18	15.77	11.20	1.34E-18
2D-MONPPn	: 61.72	60.49	45.34	41.03	28.18	15.77	11.20	1.14E-18

4.4 Discussion

- **1D approaches Vs. 2D approaches**: NPP gives decent recognition rates but ONPP is better at recognition task, whereas 2D-ONPP fails at recognition task [64], which led us to proposing a new method which forces projection to be normal in an orthogonal subspace. The proposed method drastically improves recognition rate for 1D approaches. Adding normalization constraint increases the recognition rate of 2D-ONPP also, which is comparable to that of 2D-NPP.

- **Normalization and Orthogonality**: Reconstructing data with non-orthogonal subspace involves huge calculations. Orthogonal subspace is necessary for a meaningful reconstruction of data, but as observed in reconstruction experiments documented in Section 4.3.3 only orthogonal subspace is not enough to give a good reconstruction with reduced dimensions, but normalized projection plays an important role in reconstruction, adding normalization constraint on projection of data improves reconstruction significantly at lesser number of dimensions than that of 2D-ONPP.

- **computational and time complexity**: NPP (MNPP) and ONPP (MONPP) involve non-iterative closed form solutions in forms of generalized eigenvalue problem and ordinary eigenvalue problem, respectively. On the other hand, proposed methods ONPPn (MONPPn) and 2D-ONPPn (2D-MONPPn) involves solving eigenvalue problems iterative manner for a matrix of increasing size (for $d = 1$ to $d = mn$ for 1D variants and $d = 1$ to $d = m$ for 2D variants), thus have higher computational complexity. Table 4.6 lists computational complexity for all six methods. Table 4.7 reports time taken to find bases matrix when 1000 images of size is 32×32 are considered as training set. Configuration of machine used is Intel Xeon® E5-2620 @ 2.40GB, 24 core, 64-bit with 2GB RAM allocation.

Table 4.6: Computational complexity for calculating basis matrix for all methods for an image size of $m \times n$

Method	NPP/ MNPP	ONPP/ MONPP	ONPPn/ MONPPn
1D	$\mathcal{O}(m^3n^3)$	$\mathcal{O}(m^3n^3)$	$\mathcal{O}(m^4n^4)$
2D	$\mathcal{O}(m^3)$	$\mathcal{O}(m^3)$	$\mathcal{O}(m^4)$

Table 4.7: Time taken (in ms) for calculating basis matrix for all methods

Method	NPP	MNPP	ONPP	MONPP	ONPPn	MONPPn
1D	2356	612	2148	598	4208	3519
2D	199	43	193	42	330	223

4.5 Conclusion

Linear dimensionality reduction techniques such as PCA, LPP, NPP and ONPP try to solve an optimization problem with some constraints either on projected data or on projection space. The present work proposes 1D and 2D variants of NPP that give normalized projection in an orthogonal projection space simultaneously - ONPPn and 2D-ONPPn. Recognition experiments were performed using both the variants on well-known face databases and hand-written numerals databases, for 1D versions of NPP, ONPP and ONPPn. Experiments were also performed using a modified weighing scheme namely MNPP, MONPP and MONPPn. ONPPn (MONPPn) outperforms NPP (MNPP) and ONPP (MONPP) in recognition task. It is an established fact that 2D variants of dimensionality reduction methods are computationally helpful, but they do not give competitive recognition results, experiments stated in this article shows that the recognition accuracy of 2D-ONPP can be increased by adding a normalization constraint with an added computational cost. On the other hand, methods that seek only normalized projection are not able to reconstruct the data effectively, an orthogonal subspace is advantageous for reconstruction. 2D-NPP involves huge calculations because of its non-orthogonality, 2D-ONPP can reconstruct the images but does not provide dimensionality reduction, where as proposed 2D-ONPPn is able to reconstruct an image with less number of dimensions and gives better result with lower reconstruction error as compared to 2D-ONPP.

CHAPTER 5

$l1-$norm based Orthogonal Neighborhood Preserving Projections (L1-ONPP)

All Dimensionality Reduction (DR) techniques try to optimize an error functions based on some criteria imposed either on original data points in higher dimensional space and/or on its embedding in lower dimensional space. Most of these error functions are formulated using $l2$-norm, which is not robust to the outliers. In [112], it is proved that LLE is susceptible to the presence of outliers. Being a linear extension of LLE, ONPP inherits this sensitivity towards outliers present in the data. However, recently, due to the capability of handling outliers, $l1$-norm optimization is drawing the attention of researchers [113]. The work documented here is the first attempt towards the same goal where Orthogonal Neighborhood Preserving Projection (ONPP) technique is performed using optimization in terms of $l1$-norm to handle data having outliers.

In recent times, many DR techniques involve $l1$-norm optimization [114, 115, 116, 117]. This chapter proposes a method that uses L1-PCA algorithm given in [114] to achieve $l1$-norm based ONPP (now onward denoted as L1-ONPP). This work firstly documents the experiments performed on synthetic data using L2-ONPP, showing susceptibility of it towards outliers. Secondly, the relationship between ONPP and PCA is established and proved theoretically, the experiments performed on synthetic as well as real data support the claim that ONPP bases can be obtained using PCA. L1-PCA is used to obtain L1-ONPP and performance of L2-ONPP and L1-ONPP in the presence of outliers is compared. Experimental outcomes imply that L1-ONPP outperforms L2-ONPP while dealing with the data

having outliers.

5.1 $l1$-norm for Dimensionality Reduction

As discussed, all conventional DR techniques employ optimization of a cost function expressed using $l2$-norm. Conventional ONPP proposed in [80] is also based on $l2$-norm optimization. Despite the fact that it has been employed successfully in many problems like face recognition, it is prone to the presence of outliers because the effect of the outliers with a large norm is magnified by the use of the $l2$-norm. In order to mitigate this problem and achieve robustness against outliers, research has been performed on dimensionality reduction techniques based on $l1$-norm. Many works have been done in PCA and LDA based on the use of $l1$-norm [114, 115, 116, 118, 119]. Not many efforts have been put into the use of $l1$-norm based methods in recently proposed DR techniques such as LPP and ONPP.

In [119], instead of assuming that each component of error between the original data point and its projection follows a Gaussian distribution, it is assumed to follow a Laplacian distribution and maximum likelihood estimation was used to formulate $l1$-norm PCA (L1- PCA) basis for the given data. In [119], a heuristic estimation approach for general $l1$-norm problem was applied to solve L1-PCA optimization. Whereas in [120], convex programming methods and the weighted median method were proposed for $l1$-norm PCA. Despite Being robust, L1-PCA has several disadvantages, it is computationally expensive because it is based on linear or quadratic programming. [117] discusses 2D variants of l1-norm PCA. In [118], proposed R1-PCA, bands together the merits of L2-PCA and those of L1-PCA. R1-PCA is rotational invariant like L2-PCA and it also overcomes the effect of outliers as L1-PCA does. However, these methods are highly dependent on the dimension d of a subspace to be found. For example, the projection vector obtained when $d = 1$ may not be in a subspace obtained when $d = 2$. Moreover, as it is an iterative algorithm so for a large dimensional input space, it takes a lot of time to achieve convergence. Let us now discuss the work done on $l1$-norm based

PCA in [114].

5.1.1 L2-PCA and L1-PCA

Let each data point $\mathbf{x_i}$ be a column of $\mathbf{X}$ such that $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}] \in \mathcal{R}^{mn \times N}$ be the given data matrix, where mn denotes dimensions of the original input space and N denotes number of data samples. Without the loss of generality, data is assumed to be centered at origin i.e. $\bar{\mathbf{x}} = \mathbf{0}$. L2-PCA tries to search a $d(<mn)$ dimensional linear subspace such that the bases vectors capture the direction of maximum variances by minimizing the cost function in terms of $l2$-norm:

$$\begin{aligned} \arg\max \mathcal{E}(\mathbf{y}) &= \arg\max_{\mathbf{y}} \sum_{i=1}^{N} \parallel \mathbf{y_i} - \bar{\mathbf{y}} \parallel_2 \\ & \quad where,\ \mathbf{y_i} = \mathbf{V^T x_i} \end{aligned} \tag{5.1}$$

Thus, for projection matrix $\mathbf{V}$, cost function can be written as

$$\begin{aligned} \arg\max \mathcal{E}(\mathbf{V}) &= \arg\max_{\mathbf{V}} \sum_{i=1}^{N} \parallel \mathbf{V^T x_i} - \mathbf{V^T} \bar{\mathbf{x}} \parallel_2 \\ &= \arg\max_{\mathbf{V}} \sum_{i=1}^{N} \parallel \mathbf{V^T x_i} \parallel_2 \\ \arg\max \mathcal{E}(\mathbf{V}) &= \arg\max_{\mathbf{V}} \parallel \mathbf{V^T X} \parallel_2 \\ & \quad s.\,t.\ \mathbf{V^T V} = \mathbf{I_d} \end{aligned} \tag{5.2}$$

where, $\mathbf{V} \in \mathcal{R}^{\mathbf{mn} \times \mathbf{d}}$ is the projection matrix and its d columns are the bases of the d dimensional linear subspace. Closed form solution in terms of eigenvalue problem $\mathbf{Cv} = \lambda \mathbf{v}$ where $\mathbf{C}$ is co-variance matrix of data.

Whereas, in PCA-L1 proposed in [114], instead of finding bases in the original data space that capture the direction of maximum variances which is based on the $l2$-norm, a method that maximizes the dispersion in terms of $l2$-norm in the feature space is presented to achieve robust and rotation invariant PCA. The approach presented in [114] for $l1$-norm optimization is iterative and also proven

to find a locally maximal solution.

Maximizing the dispersion in the feature space using $l1$-norm can be formulated as

$$\arg\max \mathcal{E}(\mathbf{V}) = \arg\max_{\mathbf{V}} \parallel \mathbf{V^T X} \parallel_1 \tag{5.3}$$

Since the closed form solution of problems involving $l1$-norm is not possible, the basis are sought iteratively as follows:
For $d = 1$

$$\mathbf{v_1} = \arg\max_{\mathbf{v}} \parallel \mathbf{v^T X} \parallel_1 = \arg\max_{\mathbf{v}} \sum_{i=1}^{N} |\mathbf{v^T x_i}| \quad s.\,t. \parallel \mathbf{v} \parallel_\mathbf{2} = 1 \tag{5.4}$$

For $d > 1$:
Once the basis in the direction of i^{th} maximum variance $\mathbf{v_j}$ ($\mathbf{v_1}$ for first basis) is sought by solving Eq. (5.4), the data is projected on this newly found basis vector. For rest of the basis vectors $\mathbf{v_j}$ $(2 \leq j \leq d)$ the same maximization problem given in Eq. (5.4) is solved on projected data ($\mathbf{X_j} = \mathbf{X_{j-1}} - \mathbf{v_{j-1}}(\mathbf{v_{j-1}^T X_{j-1}})$) iteratively, which essentially means in every iteration, direction of maximum variance in feature space is sought, until desirable $d(d < mn)$ dimensional space is achieved.

Algorithm to compute bases of PCA-L1[114]:

For $d = 1$:

1. Initialization:
   Pick any $\mathbf{v(0)}$
   Set $\mathbf{v(0)} \leftarrow \mathbf{v(0)} / \parallel \mathbf{v(0)} \parallel_\mathbf{2}$
   Set $t = 0$.

2. Polarity Check:
   $\forall i \in 1, ..., N$,

if $\mathbf{v^T(t)x_i} < 0$, $\mathbf{p_i(t)} = -1$,

otherwise $\mathbf{p_i(t)} = 1$

3. Flipping and maximization:

   Set $t \leftarrow t + 1$

   Set $\mathbf{v(t)} = \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{p_i(t)x_i}$

   Set $\mathbf{v(t)} \leftarrow \mathbf{v(t)} / \parallel \mathbf{v(t)} \parallel_{\mathbf{2}}$

4. Convergence Check:

   a. if $\mathbf{v(t)} \neq \mathbf{v(t-1)}$, go to Step 2.

   b. Else if there exists i such that $\mathbf{v^T(t)x_i} = \mathbf{0}$,

   set $\mathbf{v(t)} \leftarrow (\mathbf{v(t)} + \Delta\mathbf{v}) / \parallel \mathbf{v(t)} + \Delta\mathbf{v} \parallel_2$ and go to step 2. (Here, $\Delta\mathbf{v}$ is a small nonzero random vector.)

   c. Otherwise, set $\mathbf{v}^* = \mathbf{v(t)}$ and stop.

For $d > 1$:

For $j = 2$ to d,

1. Projecting Data:

   $\mathbf{X_j} = \mathbf{X_{j-1}} - \mathbf{v_{j-1}}(\mathbf{v_{j-1}^T X_{j-1}})$

2. Finding L1-PCA basis:

   in order to find $\mathbf{v_j}$, apply L1-PCA procedure to $\mathbf{X_j}$

end

5.2 L1-ONPP using PCA-L1

As stated in Section 5.1, ONPP [80] is a linear extension of LLE and thus inherits the sensitivity of LLE towards outliers. The degradation in manifold learning when the data have outliers inspired the use of $l1$-norm minimization in ONPP to tackle the outliers. In order to use PCA-L1 to achieve L1-ONPP, a relationship between PCA and ONPP in established in this Section. Firstly ONPP and a modified variant of ONPP namely MONPP is explained in detail in Section 5.2.1, followed

by a theoretical explanation of a relation between PCA and ONPP. Section 5.2.3 explains how PCA-L1 can be used to compute L1-ONPP bases.

5.2.1 L2-ONPP and L2-MONPP

ONPP and MONPP both use $l2$-norm optimization to find projection matrix, thus inherit the susceptibility of LLE to the presence of the outliers. Optimization problem of finding ONPP embedding can be posed as a minimization problem with cost function

$$\arg\min \mathcal{E}(\mathbf{Y}) \quad = \quad \arg\min_{Y} \sum_{i=1}^{N} \parallel \mathbf{y_i} - \sum_{j=1}^{N} w_{ij}\mathbf{y_j} \parallel_2$$

The projection matrix $\mathbf{V}$ can be achieved by solving

$$\begin{aligned} \arg\min \mathcal{E}(\mathbf{V}) \quad &= \quad \arg\min_{V} \sum_{i=1}^{N} \parallel \mathbf{V^T x_i} - \sum_{j=1}^{N} w_{ij}\mathbf{V^T x_j} \parallel_2 \\ & \qquad s.\,t.\ \mathbf{V}^T\mathbf{V} = \mathbf{I_d} \end{aligned} \tag{5.5}$$

This optimization problem results in an eigenvalue problem with the closed form solution. The eigen-vectors corresponding to the smallest d eigen-values of matrix $\mathbf{M} = \mathbf{X}(\mathbf{I} - \mathbf{W})(\mathbf{I} - \mathbf{W}^T)\mathbf{X}^T$ constitutes the basis of the low dimensional ONPP space. ONPP explicitly maps $\mathbf{X}$ to $\mathbf{Y}$, which is of the form $\mathbf{Y} = \mathbf{V}^T\mathbf{X}$, where, each column of $\mathbf{V}$ is an eigen-vector of $\mathbf{M}$.

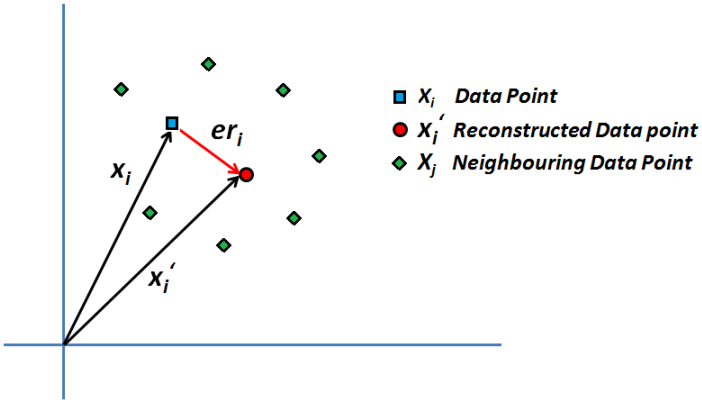


Figure 5.1: Illustration of data point $\mathbf{x_i}$ (represented by a blue square), its reconstruction $\mathbf{x_i^{'}}$ using neighbors (represented by a red circle) $\mathcal{N}_{\mathbf{x_i}}$ and error vector $\mathbf{er_i}$ (represented by a green diamond). i^{th} reconstruction error vector is denoted by $\mathbf{er_i}$

5.2.2 ONPP as a PCA on Reconstruction Error

Rewriting the Eq. (5.5) and Eq. (5.5) in a matrix form, to establish the relationship between ONPP and PCA:

$$\begin{aligned} \arg\min \mathcal{E}(\mathbf{Y}) &= \arg\min_{\mathbf{Y}} \parallel \mathbf{Y} - \mathbf{YW} \parallel_2 \qquad (5.6) \\ \arg\min \mathcal{E}(\mathbf{V}) &= \arg\min_{\mathbf{V}} \parallel \mathbf{V^T X} - \mathbf{V^T XW} \parallel_2 \\ &= \arg\min_{\mathbf{V}} \parallel \mathbf{V^T}(\mathbf{X} - \mathbf{XW}) \parallel_2 \\ \arg\min \mathcal{E}(\mathbf{V}) &= \arg\min_{\mathbf{V}} \parallel \mathbf{V^T Er} \parallel_2 \qquad (5.7) \\ & \quad s.t.\ \mathbf{V}^T\mathbf{V} = \mathbf{I} \end{aligned}$$

Now comparing the optimization problems of PCA (Eq. (5.2)) and the optimization problem of ONPP (Eq. (5.6)), both results into an eigenvalue problems and have closed form solutions in term of eigen-vectors. Eq. (5.2) is maximization problem thus the bases vectors of PCA are eigen-vectors corresponding to largest d eigenvalues, where as Eq. (5.6) is minimization problem thus the desired ONPP bases are eigen-vectors corresponding to smallest d eigenvalues.

In other words, ONPP can be stated as a PCA of reconstruction errors, conventional ONPP algorithm is essentially finding the basis vectors $\mathbf{V}$ such that

it actually captures the directions of minimum variances of reconstruction error. Thus, finding the strongest ONPP basis is same as finding the weakest basis of PCA when performed on the reconstruction errors **Er**. Each column of **Er** is a reconstruction error for i^{th} data point and calculated using its neighbors and corresponding weights w_{ij}

$$\mathbf{er_i} = \mathbf{x_i} - \sum_{j=1}^{N} w_{ij}\mathbf{x_j} \tag{5.8}$$

This relationship of PCA and ONPP bases is verified in experiments performed on the synthetic data and it is observed that the ONPP bases obtained using the conventional L2-ONPP algorithm and ONPP bases obtained using L2-PCA on reconstruction error are same. Details of this experiment are documented in Section 5.3.

5.2.3 L1-ONPP using PCA on Reconstruction Error

Once the relationship between L2-PCA bases and L2-ONPP bases is in place, it is evident that PCA algorithm can also be used to find ONPP bases. This led to the use of existing $l1$-norm based PCA algorithms to solve L1-ONPP optimization problem. Rewriting L2-ONPP optimization problem in Eq. (5.5) using $l1$-norm minimization, we have

$$\begin{aligned} \arg\min \mathcal{F}(\mathbf{Y}) &= \arg\min_{\mathbf{Y}} \sum_{i=1}^{N} \parallel \mathbf{y_i} - \sum_{j=1}^{N} w_{ij}\mathbf{y_j} \parallel_1 \\ & \quad s.\,t.,\ \mathbf{V}^T\mathbf{V} = I \end{aligned} \tag{5.9}$$

In matrix form,

$$
\begin{aligned}
\arg\min \mathcal{F}(\mathbf{Y}) &= \arg\min_{\mathbf{Y}} \parallel \mathbf{Y} - \mathbf{YW} \parallel_1 \\
\arg\min \mathcal{F}(\mathbf{V}) &= \arg\min_{\mathbf{V}} \parallel \mathbf{V^T X} - \mathbf{V^T XW} \parallel_1 \\
&= \arg\min_{\mathbf{V}} \parallel \mathbf{V^T}(\mathbf{X} - \mathbf{XW}) \parallel_1 \\
&= \arg\min_{\mathbf{V}} \parallel \mathbf{V^T Er} \parallel_1 \qquad (5.10)
\end{aligned}
$$

The problem stated in Eq. (5.10) is similar to problem stated in PCA-L1 (Eq. (5.4)), thus the solution of Eq. (5.4)) can be used to solve Eq.(5.10). L1-ONPP bases can be found using any $l1$-norm based PCA algorithm when performed on reconstruction error matrix **Er**. As discussed in Section 5.1 many $l1$-norm based PCA methods have been developed which find bases vectors through linear or quadratic programming. These methods are computationally expensive. The PCA-L1 algorithm [114] used here is a robust as well as fast $l1$-norm based method. PCA-L1 converts the $l1$-norm variance into a direct sum of signed training points into projection space. The bases vectors are updated by the sum of signed training points. As a result, the convergence procedure is fast. Refer to [114] for the proof.

L2-ONPP involves the closed form solution which involves eigenvalue problem of matrix size $mn \times mn$. L1-ONPP is computationally costly because it involves an iterative procedure, each basis vector $\mathbf{v_k}$ starts from a random mn dimensional vector and polarity check, flipping and maximization is performed iteratively until $\mathbf{v_k}$ converges.

Comparing Eq. (5.10) of L1-ONPP with Eq. (5.4) of L1-PCA, we can intuitively state that the component in the direction of minimum variance gives the strongest ONPP basis. Considering reconstruction error between a data point $\mathbf{x_i}$ and its approximation $\mathbf{x_i^{'}}$ as a vector $\mathbf{er_i}$, which is also a point in mn-dimensional space (as shown in Fig. 5.1) PCA-L1 can be performed on the reconstruction errors to search for the d-dimensional space such that the bases vectors are in the direction of minimum variances of these reconstruction errors. Such bases can be computed using proposed algorithm given in Table 5.1.

Table 5.1: L1-ONPP Algorithm

Input: Dataset $\mathbf{X} \in R^{mn \times N}$and number of reduced dimension d

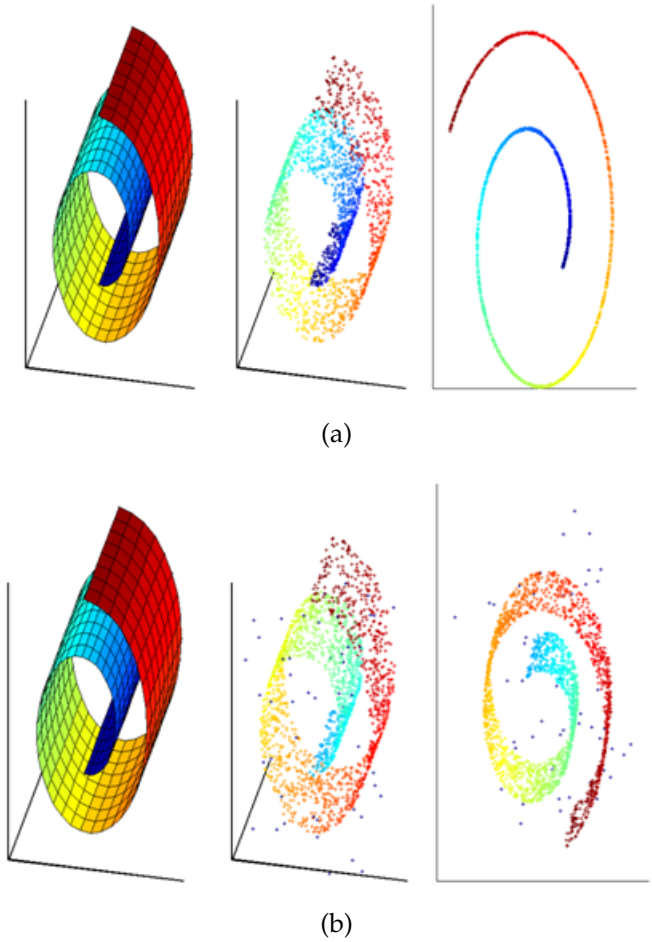
Output: Lower dimension projection $\mathbf{Y} \in R^{d \times N}$

1: Compute NN with class label information (in supervised mode) or using k-NN algorithm (in unsupervised mode).
2: Compute the weight w_{ij} for each neighbor of $\mathbf{x_i}$ data point $\mathbf{x_j} \in \mathcal{N}_{\mathbf{x_i}}$
3: Compute reconstruction error matrix $\mathbf{Er}$ using equation (5.8)
4: Set $\mathbf{v_0} = \mathbf{0}$, $\mathbf{Er^0} = \mathbf{Er}$
5: **for** $(j = 1, j \leq d, ++j)$ **do**
6: Set $\mathbf{Er^j} = \mathbf{Er^{j-1}} - \mathbf{v_{j-1}}(\mathbf{v_{j-1}^T}\mathbf{Er^{j-1}})$
7: Initialize $\mathbf{v(0)}$
8: Set $t = 0$
9: **for**$(i = 1; i \leq N; ++i)$ **do**
10: **if** $\mathbf{v(t)^T er_i^j} < 0$ **then**
11: $\mathbf{p_i(t)} = -1$
12: **else**
13: $\mathbf{p_i(t)} = 1$
14: **end if**
15: **end for**
16: Set $t = t + 1$
17: Set $\mathbf{v(t)} = \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{p_i(t) er_i^j}$
18: Set $\mathbf{v(t)} \leftarrow \mathbf{v(t)} / \parallel \mathbf{v(t)} \parallel_{\mathbf{2}}$
19: **if** $\mathbf{v(t)} \neq \mathbf{v(t-1)}$ **then**
20: Go to Step 9.
21: **else if** There exists i such that $\mathbf{v^T(t) er_i^j} = \mathbf{0}$ **then**
22: Set $\mathbf{v(t)} \leftarrow (\mathbf{v(t)} + \Delta\mathbf{v}) / \parallel \mathbf{v(t)} + \Delta\mathbf{v} \parallel_2$ and go to step 9.
(Here, $\Delta\mathbf{v}$ is a small nonzero random vector.)
23: **else**
24: Set $\mathbf{v_j} = \mathbf{v(t)}$
25: **end if**
26: **end for**
27: Project $\mathbf{X}$ on L1-ONPP space using $\mathbf{V}$ to get embeddings $\mathbf{Y} = \mathbf{V^T X}$.

5.3 Experiments

To validate the theoretical conclusion on the relationship between ONPP and PCA, experiments were performed on the synthetic as well as real data as documented in this Section.

5.3.1 A small problem with Swiss role data



(a)

(b)

Figure 5.2: L2-ONPP performed on Swiss role data (a) Continuous manifold (left), sampled 3D data (middle) and its 2D representation using strongest 2 basis of ONPP (right) (b) Continuous manifold (left), sampled 3D data corrupted with additional outliers from uniform distribution (middle) and its 2D representation using strongest 2 basis of ONPP (right)

In literature, the definition of an outlier is given as a data point that seems to be taken from an entirely different distribution. To observe the effect of outliers on L2-ONPP algorithm, an experiment was performed on *Swissrole* data. Over 2000

3-dimensional data points were randomly sampled from a continuous *Swissrole* manifold. 2-dimensional embedding of clean data were found using L2-ONPP as shown in Fig. 5.2(a-*right*). Now 50 data points (nearly 2.5% of clean data) from a normal distribution are added to these 2000 clean data point as outliers, 2-dimensional embedding of this data are also found using L2-ONPP as shown in Fig. 5.2(b-*right*).

Comparing embedding from clean data (Fig. 5.2(a)) and embedding from data having outliers (Fig. 5.2(b)), it can be observed that global structure, as well as local geometry is well preserved in case of clean data. Whereas in case of noisy data (Fig. 5.2(b)), 2-dimensional representation is distorted. The reason is all neighbors of the clean data point may not lie on a locally linear patch of a manifold in the presence of outliers, which leads to the biased reconstruction. On the other hand, the neighborhood patch of the outlier will be comparatively very large and thus does not capture local geometry very well, as the effect of large distance is exaggerated by the use of $l2$-norm. It has been known that $l2$-norm based techniques are not robust, in the sense that the presence of outliers can arbitrarily skew the solution from the desired solution.

Another experiment was performed to analyze the effectiveness of proposed algorithm in the presence of varying numbers of outliers. As shown in Fig. 5.3(b), 1000 3D clean data points were sampled from continuous S-shaped manifold shown in Fig. 5.3(a). Fig. 5.3(c) and Fig. 5.3(h) show the 2D representation of clean data using L2-ONPP and L1-ONPP respectively. The clean data was corrupted with 100, 200, 300 and 400 outliers sampled from a uniform distribution. Fig. 5.3(d) to (g) shows the 2D representation of noisy data using L2-ONPP and Fig. 5.3(i) to (l) shows the 2D representation of noisy data using L1-ONPP. As it can be seen from this experiment L1-ONPP very well handles outliers by preserving intrinsic neighborhood relations as well as global geometry of data. On the contrary, increasing density of outliers distorts the learned manifold using L2-ONPP in increasing manner, the presence of outliers even affects the orientation of data as can be seen from Fig. 5.3 (d) and (e).

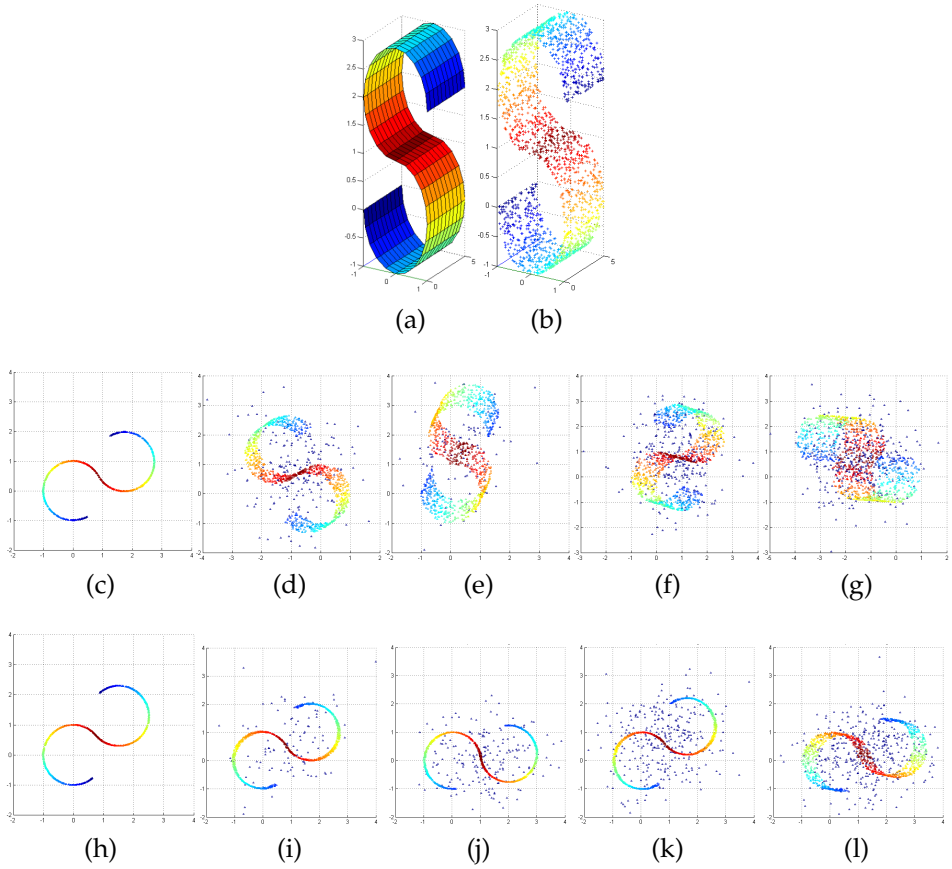


Figure 5.3: Manifold learning on S-Curve data (a) Continuous manifold (b) Sampled clean 3D data points. Its 2D representation using strongest 2 basis of L2-ONPP starting with (c) to (g) with clean data, 100, 200, 300 and 400 outliers respectively. Its 2D representation using strongest 2 basis of L1-ONPP starting with (h) to (l) with clean data, 100, 200, 300 and 400 outliers respectively.

5.3.2 Comparing Bases of L2-PCA, L2-ONPP and L1-ONPP

To validate the relationship between PCA and ONPP as described in Section 5.2.1 the experiment was performed on synthetically generated data. 2D data was randomly generated to form 7 clusters with 100 data point each resulting in 700 data points. The clusters are closely placed and slightly overlapping, 2 out of 7 were slightly separated as shown in Fig. 5.4(a). L2-ONPP bases were found using the conventional algorithm and another set of bases vectors were computed by performing L2-PCA on reconstruction error. The bases found using both methods are same.

L2-norm ONPP basis [Fig. 5.4(a)]
1^{st} basis : $[0.6361, 0.7716]^T$ 2^{nd} basis: $[-0.7716, 0.6361]^T$

PCA basis on Reconstruction errors [Figure 5.4(b)]
1^{st} basis : $[0.6360, 0.7717]^T$ 2^{nd} basis: $[-0.7717, 0.6360]^T$

Figure 5.4(c) shows that L2-ONPP bases are essentially pointing the direction in which the variance of reconstruction error is minimum. Note that the reconstruction error of all data point is centered at origin (same as the assumption in PCA that the data points are mean centered) For this data, L1-ONPP bases were computed using PCA-L1 algorithm. As it can be seen from Figure 5.4(d) the projection basis are tilted towards the outlier data.

L1-norm ONPP basis [Figure 5.4(b)]
1^{st} basis : $[0.4741, 0.8805]^T$ 2^{nd} basis: $[-0.8805, 0.4741]^T$

In this experiment, the residual error was observed for both, L2-ONPP and L1-ONPP. The residual error is a measure of how well the information is preserved while projecting data on lower dimensional space using few strongest bases while discarding other dimensions. In this case, the data was projected using only 1 dimension using the strongest basis vector. The average residual error was calculated using

$$\mathbf{e_{avg}} = \frac{\mathbf{1}}{\mathbf{N}} \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{x_i} - \mathbf{v_1}(\mathbf{v_1^T x_i}) \tag{5.11}$$

The average residual errors of L2-ONPP and L1-ONPP are 2.3221 and 0.7894, respectively. Thus. it can be concluded that L1-ONPP is less susceptible to outliers compared to L2-ONPP. The same behavior related to residual error is observed with real data also as stated in the following experiment.

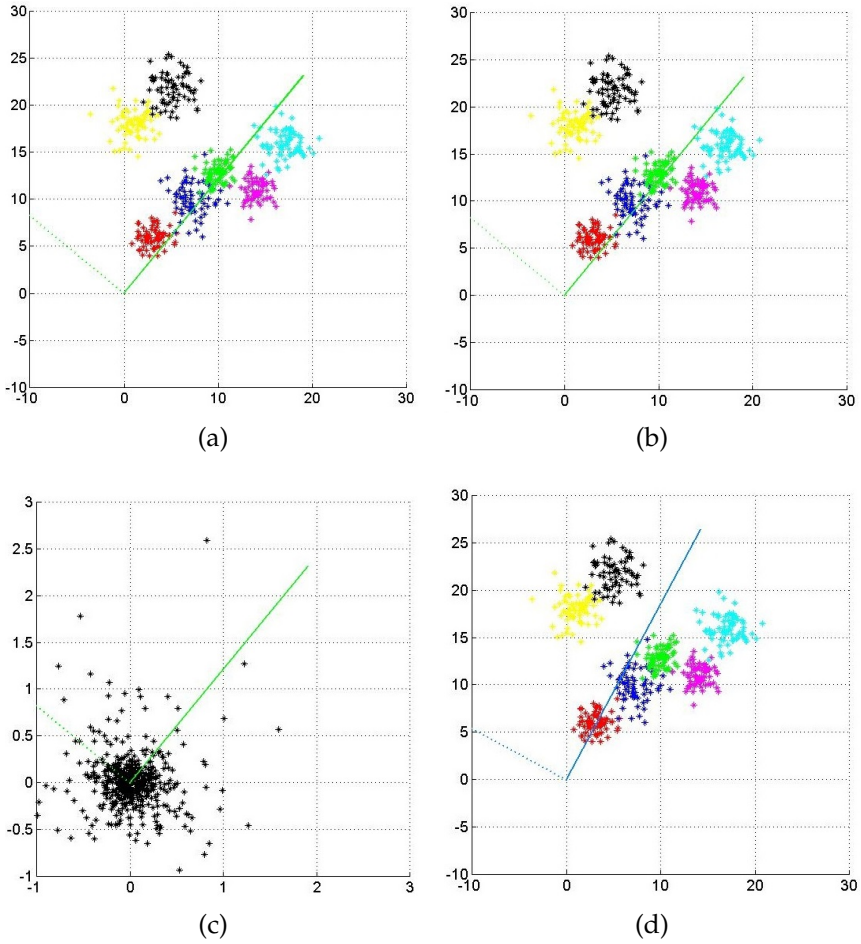


Figure 5.4: A toy example with 700 data samples from 7 clusters. Solid line represents first projection basis and dotted line represents second projection basis (a) Projection basis using conventional L2-ONPP (b) Projection basis using L2-PCA on reconstruction basis (c) Projection basis overlapped on reconstruction errors (d) Projection basis using proposed L1-ONPP

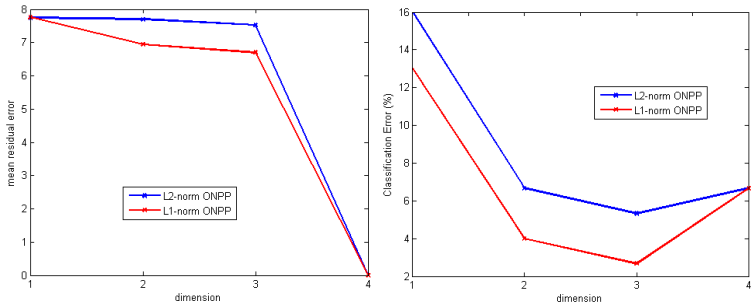
5.3.3 Experiment with IRIS dataset

To further compare the behavior of L2-ONPP and L1-ONPP regarding residual error and performance in classification task, IRIS data from UCI Machine Learning Repository [121] is used. The data-set contains 150 instances of 4-dimensional data belonging to three different classes. The residual error obtained while reconstructing the data using a varying number of dimensions is shown in Fig. 5.5. Table 5.2 lists the residual errors using the different number of dimensions, as it can be seen, the residual error is less in L1-ONPP as compared to L2-ONPP which significantly improves classification error at lower dimensions. When in

case of all 4 dimensions are used in projection space, the projection of data spans entire original space, thus the residual error drops to almost zero for both methods, L2-ONPP and L1-ONPP. Similar behaviors expected for classification also at 4 dimensions, as can be seen from Table 5.2 the classification error at 4 dimensions yields greater than the lower dimension representation because it includes the redundant details present in higher dimensions. The same behavior at higher dimensions can be observed in all DR techniques. Here, Nearest Neighbor (NN) is used as a classifier.

Table 5.2: Comparison of performance in terms of residual error and classification error (in %) of L2-ONPP and L1-ONPP with varying number of dimensions on IRIS data

	Residual Error		Classification Error (%)	
Dim	L2-ONPP	L1-ONPP	L2-ONPP	L1-ONPP
1	7.8614	7.8546	16.00	13.00
2	7.7450	7.6730	6.67	4.00
3	7.0055	5.6545	5.33	2.67
4	1.44e-15	1.45e-15	6.67	6.67



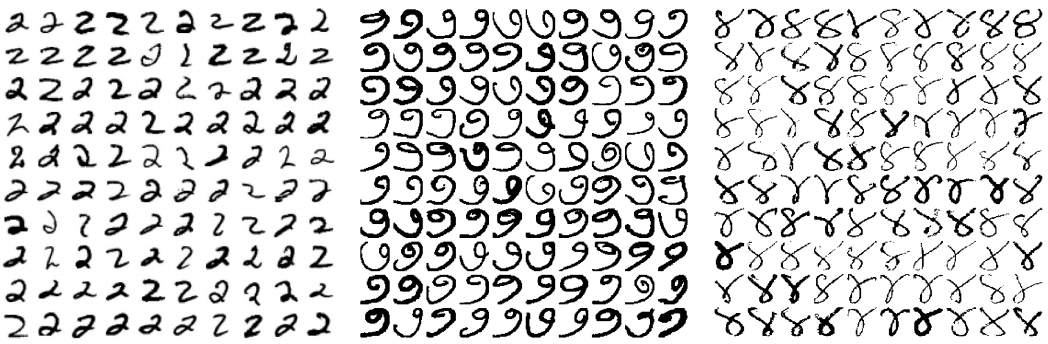
(a) (b)

Figure 5.5: Performance comparison of L2-ONPP and L1-ONPP with respect to varying number of dimensions used to reconstruct the IRIS data in terms of (a) Residual Error (b) Classification Error(%)

5.3.4 Experiment with Handwritten Numerals

Handwritten text or numerals have huge variations in terms of shape, stroke width, orientations and pattern, thus makes an ideal data to test the capacity of L1-ONPP to handle outliers. To compare the performance of L1-ONPP and L2-ONPP with real-world data having outliers, three handwritten numeral databases,

one English and two different Indian scripts Gujarati and Devanagari are used. The images in each database are resized to 30×30 to maintain uniformity across all three databases.



(a) (b) (c)

Figure 5.6: (a) Examples of 2s in the MNIST database first 100 examples. (b)Examples of 7s in the Gujarati database first 100 examples. (c) Examples of 4s in the Devnagari Numerals database first 100 examples. Notice the very diverse shape, stroke width, orientation and pattern of different digits.

English Numerals

The MNIST database [83] is a large database of handwritten English digits which contains nearly 68,000 images of each digit. Some of the examples of digit $'2'$ are shown in Figure 5.6(a). 1000 samples were selected randomly such that each digit is equally present in the training, while remaining samples were used for testing. Average recognition accuracy of 20 randomizations are reported here. Performance of L2-ONPP, L2-MONPP are compared with L1-ONPP and L1-MONPP with varying number of dimensions as shown in Figure 5.7. As it can be seen L1-ONPP and L1-MONPP outperforms its L2-norm counterparts with a large difference. Best average recognition accuracy of L1-ONPP and L1-MONPP are almost same, nearly 87.52% achieved at 210 dimensions. As it can be seen from Figure 5.7, the performance of L2-ONPP and L2-MONPP is poor compared to its L1-norm counterparts.

Gujarati Numerals

The Gujarati Handwritten numeral dataset [84] have nearly 1300 samples for each digit. Some of the examples of digit $'7'$ are given in Figure 5.6 (b) to show the large variations in the dataset. Randomly 1000 samples were selected such that

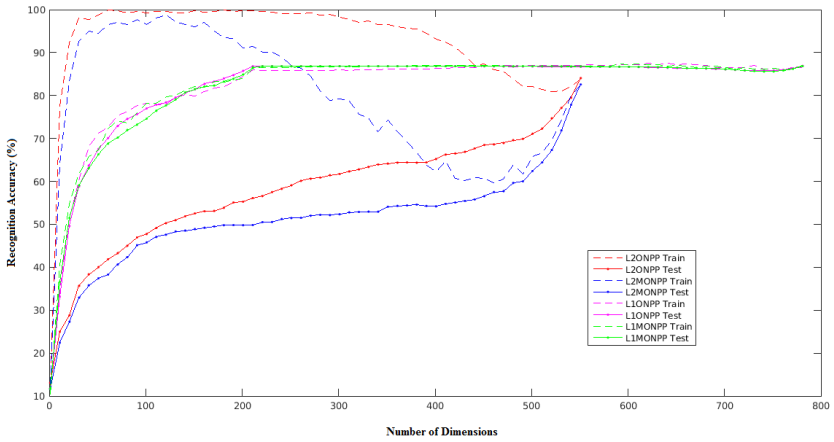


Figure 5.7: Performance comparision in terms of recognition accuracy for L2-ONPP, L1-ONPP, L2-MONPP and L1-MONPP for MNIST Handwritten numerals database

each digit is well represented in training data and the remaining samples were used as testing data. Recognition accuracy of L2-ONPP, L2-MONPP, L1-ONPP, and L1-MONPP with varying number of dimensions are compared in Fig. 5.8. Average recognition accuracy achieved by L1-ONPP and L1-MONPP is 75.20% and 67.32% nearly at 20 dimensions, whereas that of L2-ONPP and L2-MONPP is 67.16% and 61.28% nearly at 290 dimensions. As it can be seen the performance of L1-ONPP and L1-MONPP stabilizes nearly at 30 dimensions, but the performance of L2-ONPP and L2-MONPP deteriorates after 300 dimensions due to the presence of redundant information present at higher dimensions, but the use all dimensions again leads to almost similar recognition as that of L1-norm counterparts as observed with IRIS data.

Devnagari Numerals

The Devnagari Hand-written Numerals Database [85] have approximately 1800 sample of each digit. Randomly selected samples of digit '4' are shown in Fig. 5.6(c). Randomly 900 samples are used for training data and the remaining are used for testing. As it can be seen from Figure 5.9 L1-ONPP and L1-MONPP achieves nearly 50% and 60% recognition accuracy at 30 dimensions, whereas the performance of L2-ONPP and L2-MONPP is consistently poor compared to L1-norm counter-part.

With a larger number of training data, L1-ONPP proves to be better at recog-

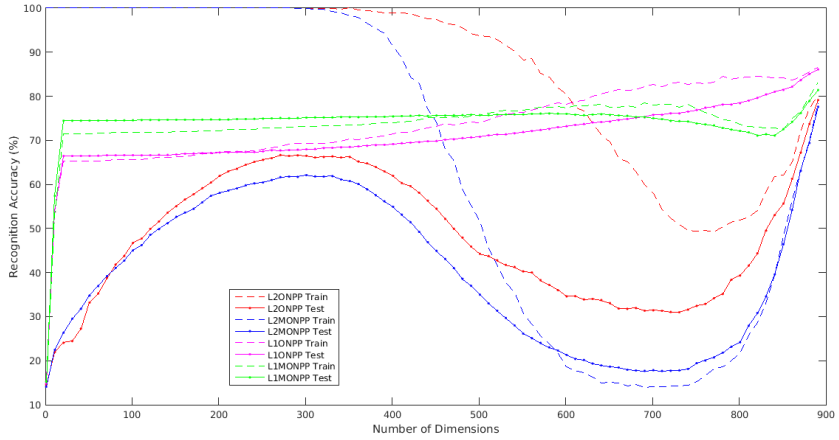


Figure 5.8: Performance comparison in terms of recognition accuracy for L2-ONPP, L1-ONPP, L2-MONPP and L1-MONPP for Gujarati Handwritten numerals

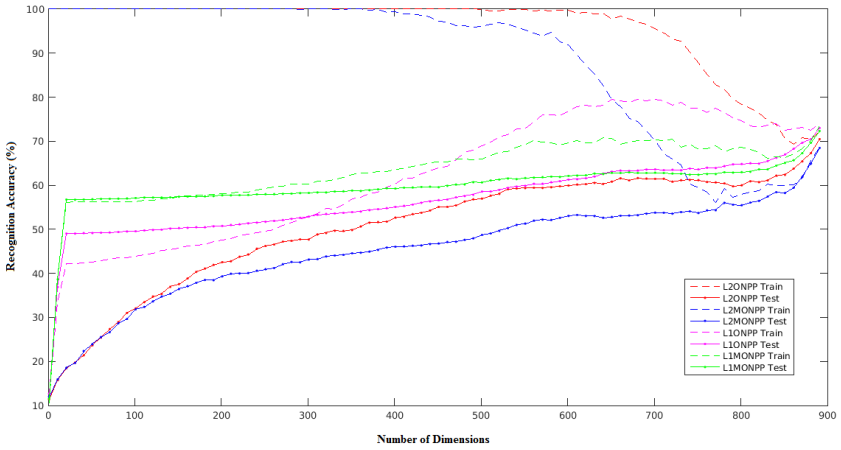


Figure 5.9: Performance comparison in terms of recognition accuracy for L2-ONPP, L1-ONPP, L2-MONPP and L1-MONPP for Devnagari Handwritten numerals database

nizing digits compared to L2-ONPP. L2-MONPP and L1-MONPP also perform at par with L2-ONPP and L1-ONPP. Here, Nearest Neighbor (NN) is used for classification, the use of sophisticated classifier like SVM etc. can lead to improved recognition accuracy. The proposed L2-ONPP algorithm converges in about 16 iterations for a single basis vector when nearly 1000 sample of images having size 30×30 are used for training and the procedure of learning L1-ONPP bases took average 3 min. On the other hand, having a closed form solution, L2-ONPP takes on an average 7 sec to learn the bases. The configuration of the machine used is as follows: Intel Xeon® E5-2620 @ 2.40GB, 24 core, 64-bit with 2GB RAM allocation.

5.4 Conclusion

Linear dimensionality reduction techniques such as PCA, LDA, LPP, and ONPP solve an optimization problem based on some criteria. Usually, the optimization problem is defined using $l2$-norm. However, use of $l2$-norm makes these techniques susceptible to outliers present in the data. The present work is first attempt to compute bases vectors for ONPP using $l1$-norm. In particular, a relation is established to show that ONPP bases can be obtained by performing PCA on reconstruction error. This phenomenon is established both theoretically and experimentally. An existing algorithm of finding PCA bases using $l1$-norm optimization is applied to compute the L1-ONPP bases. It has also been proved experimentally that the residual error calculated after discarding few dimension in projection space and reconstructing data with less number of dimension is comparatively low in case of L1-ONPP than that of L2-ONPP. Experiments are performed for synthetic as well as real data, and the same conclusion as mentioned above is observed. Performance of L1-ONPP is compared with L2-ONPP on numeral recognition task, it is observed that with a larger number of training data, L1-ONPP outperforms L2-ONPP with huge margin, but being an iterative method, L1-ONPP is computationally expensive compared to L2-ONPP.

CHAPTER 6

Conclusions and Future Work

Advancement in image capturing sensors and storage devices led us to the a huge amount of high-definition images and pose a challenge in the form of image recognition in high-dimensional space. It is proved that natural images, even though represented as a very high-dimensional data point, lies on or near a linear or non-linear manifold. Such subspace or manifolds based methods for image recognition have been a major area of research and already proven to be more efficient. Work presented in this thesis, addresses few concerns with one such technique ONPP and provides solution for the issues addressed. Basically, this thesis suggests an improvement of each stage of ONPP to handle various problems faced by conventional technique to make it more robust for image recognition and image reconstruction task.

6.1 Overall Conclusion

Orthogonal Neighborhood Preserving Projection (ONPP) is assumed to handle the intrinsic non-linearity of the data manifold. The second step of ONPP deals with a linear model building within local neighborhoods. In this thesis, it is shown that the linearity assumption may not be valid for a moderately large neighborhood. This linear model is thus replaced by a notion of non-linearity in the proposed method MONPP, where a piece-wise non-linear model (z-shaped) is used instead. The suitability of the proposal is tested on non-linear synthetic data as well as a few benchmark face databases. Significant and consistent improvement in data compactness is observed for synthetic data like *swissrole* and *Scurve* where

manifold is surely nonlinear. This signifies the suitability of the present proposal to handle non-linear manifold of the data. Lower-dimensional visualization using MONPP on handwritten numerals data is also seen to be more compact than that of ONPP. On the other hand, noticeable improvement is obtained for the face recognition problem performed on benchmark databases like ORL, AR and UMIST. The modification suggested over existing ONPP though very simple but overall improvement in face recognition results is very encouraging.

Handling illumination variations present in face images captured under uncontrolled environment while face recognition task is paramount, thus a robust pre-processing technique is highly sought. This thesis contributes to adopt Locally Tuned Inverse Sine Nonlinear (LTISN) transformation for gray-scale face images to nullify the illumination variations present in the face database to improve recognition rate. The result of recognition along with LTISN as pre-processing is compared with that of another pre-processing technique called Difference of Gaussian (DoG). The classifier used in all technique is nearest neighbor applied on the coefficient obtained using ONPP and MONPP. Image recognition experiments were performed on two well-known face database having illumination variations, namely The Extended Yale-B and CMU-PIE. It is observed that LTISN based enhancement followed by MONPP outperforms ONPP with or without both DoG and LTISN enhancement techniques.

For image recognition tasks, dimensionality reduction techniques are usually implemented in supervised settings. Conventional supervised ONPP algorithm, in the first step selects neighbors based on the knowledge of class label, which may not be the best way to select when data distribution is highly overlapping. In this thesis, we propose a new neighborhood selection rule based on class similarity of all data points. The proposed Class-Similarity based ONPP (CS-ONPP) takes advantage of pre-processing PCA, the lower dimensional projections on PCA space is used to find a class probability vector for each data-point to belonging into a particular class and class similarity of two neighboring data-points using Logistic Discriminator (LD). Class Similarity is computed based on this probability vector and a new distance measure is defined based on the class simi-

larity between two data points. Neighbors are selected using this new distance measure. Experiments performed on well-known Face data like ORL, UMIST and CMU-PIE, it is observed that CS-ONPP achieves same performance as ONPP with comparatively a very less number of subspace dimensions. The same performance is observed on handwritten numerals databases like MNIST, Devnagari and Gujarati.

Linear DR techniques such as PCA, LPP, NPP and ONPP try to solve an optimization problem with some constraints either on projected data or on projection space. This thesis proposes 1D and 2D variant of ONPP that gives normalized projection in an orthogonal projection space - ONPPn and 2D-ONPPn. Recognition experiments were performed using both the variants on well-known face databases and hand-written numerals databases, for 1D versions of NPP, ONPP and ONPPn. Experiments were also performed using a modified weighing scheme namely MNPP, MONPP and MONPPn. ONPPn (MONPPn) outperforms NPP (MNPP) and ONPP (MONPP) in image recognition task performed on face databases like AR, ORL, UMIST, The Extended Yale-B and hand-written numerals databases like MNIST, Gujarati, Devnagari and Bangla. It is an established fact that 2D variants of dimensionality reduction methods are computationally helpful because they involve eigenvalue problem of order m^3 instead of $(mn)^3$, but they do not give competitive recognition results. Experiments stated in this thesis show that the recognition accuracy of 2D-ONPP can be increased by adding a normalization constraint with an added computational complexity. On the other hand, methods that seek only normalized projection are not able to reconstruct the data, an orthogonal subspace is desired for reconstruction. Experiment shows that 2D-NPP is computationally costly at reconstruction because of its non-orthogonality, 2D-ONPP can reconstruct the images but does not provide dimensionality reduction, 2D-ONPPn is able to reconstruct an image with retaining less number of subspace dimensions and gives better result with lower reconstruction error as compared to 2D-ONPP.

Most DR techniques such as PCA, LDA, LPP and ONPP solve an optimization problem which is usually defined using $l2$-norm. However, use of $l2$-norm makes

these techniques susceptible to outliers present in the data. The presented work is first attempt to compute bases vectors for ONPP using $l1$-norm. In particular, a relation is established to show that ONPP bases can be obtained by performing PCA on reconstruction error. These phenomenon is established both theoretically and experimentally. An existing algorithm of finding PCA bases using $l1$-norm optimization is applied to compute the L1-ONPP bases. It has also been proved experimentally that the residual error calculated after discarding few dimension in projection space and reconstructing data with less number of subspace dimension is comparatively low in case of L1-ONPP than that of L2-ONPP. Experiments are performed for synthetic as well as real data, and the same conclusion as mentioned above is observed. Performance of L1-ONPP is compared with L2-ONPP on numeral recognition task, it is observed that with larger number of training data, L1-ONPP outperforms L2-ONPP with huge margin, but being an iterative method, L1-ONPP is computationally expensive compared to L2-ONPP but easy to implement.

Over all conclusion is that the thesis addresses some of the issues in conventional ONPP and implemented those successfully for image recognition and image reconstruction. All the new proposal are tested on benchmark databases of face recognition and handwritten numerals recognition. In all cases, the new proposals outperforms the conventional method in terms of recognition accuracy with reduced subspace dimensions.

6.2 Contributions

Summary of contributions of the thesis is listed below:

- A variant of ONPP namely Modified ONPP (MONPP) that handles local non-linearity that comes with a varying number of neighbors. MONPP employs a modified technique for calculating reconstruction weights for neighbors of a data point.

- Locally adaptable pre-processing technique to handle illumination variations in gray-scale face images

- Class similarity based Orthogonal Neighborhood Preserving Projections: A new approach to find neighbors instead of relying on only Euclidean distance or Class label knowledge proposed method uses both the information to define a new distance measure.

- Orthogonal Neighborhood Preserving Projection with Normalization that brings normalization and orthogonality constraint together in ONPP and its 2D variant. Normalized projections increased the image recognition accuracy, and adding normalization constraint to the orthogonality constraint enabled 2D-ONPP to reconstruct an image using less number of subspace dimensions.

- A L1-norm based ONPP technique to handle data having wide variations. To achieve L1-ONPP bases, a relationship between PCA and ONPP is established and PCA-L1 is used to compute L1-ONPP subspace that is invariant to the presence of outliers.

6.3 Future Work

The scope of the thesis lies in the proposing a robust neighborhood preserving dimensionality reduction technique that can be used in image recognition application. Though the variants proposed during the course of this thesis is used in face recognition and handwritten numerals recognition task, their application is not limited to the image recognition. As shown in [122, 123], applicability of neighborhood preserving DR is not limited on image data only. The usefulness of proposed methods can be explored for various other applications involving high-dimensional data too.

The thesis defined neighborhood based on the pairwise distances between two images, or class label of the images or some combination of the both. Euclidean distance is proved to be a good measure for image data. While dealing with other type of high-dimensional data for example text document or Hyperspectral Images (HSI), it will be a good choice to explore the distance measure suitable for the mentioned problem and data at hand.

The data in its raw form, does not necessorily lie on a linear manifold, thus non-linear DR methods try to learn local neighborhoods and patches them up in low-dimensional manifold to get to the global geometry of data, but may fail if the underlying structure contains complex non-linearities. One way to handle such data is to project them on even high-dimensional non-linear feature space and followed by application of DR technique. This technique is known as a Kernels trick. One possible direction of work can include defining new kernels to better suit the given data and application.

In this thesis, images are considered as a gray-scale images and treated as either mn-dimensional vector or $m \times n$ matrix, but for color images tensor representation can be exploited using tensor variants of dimensionality reduction methods that can handle 3 or higher dimensional data. One possible direction of work can be exploiting L1-ONPP for face image reconstruction where a part of face is occluded. As occluded faces can be considered the outliers in the face image database. It will be an interesting work to see if a 2D variant of ONPP with $l1$-norm can handle problem of reconstructing such outlying face images.

Deep learning has emerged as a very interesting of research with widely applicable area nowadays, due to the availability of huge unlabeled data, patterns can be learned from unlabeled data from a related task domain and the learning can be transferred to target domain where the labeled data is available. In recent time, deep neural networks are used to learn underlying low dimensional representations of data [79]. It will be an interesting investigation to see if we can take advantage of unlabeled data to learn underlying low-dimensional manifold using deep neural networks. Auto-encoders and Generalized auto encoders can be used to with a cost function that immitate the optimization problem of well-known dimensionality reduction methods and instead of solving a generalised eigenvalue problem the bases can be trained using DNN.

References

[1] David L Donoho. Aide-memoire. high-dimensional data analysis: The curses and blessings of dimensionality. 2000.

[2] Christopher M Bishop et al. Pattern recognition and machine learning (information science and statistics). 2006.

[3] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Academic press, 2013.

[4] Rafael C Gonzalez. Digital image processing, prentice hall, 2016.

[5] G Shikkenawis and S K Mitra. 2D Orthogonal Locality Preserving Projection for Image Denoising. *IEEE Transactions on Image Processing*, 25(1):262–273, 2016.

[6] X He and P Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems*, pages 153–160, 2004.

[7] X He, D Cai, S Yan, and H J Zhang. Neighborhood preserving embedding. In *Tenth IEEE International Conference on Computer Vision, ICCV 2005.*, volume 2, pages 1208–1213. IEEE, 2005.

[8] D Sacha, L Zhang, M Sedlmair, J A Lee, J Peltonen, D Weiskopf, S C North, and D A Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE transactions on visual. and comp. graphics*, 23(1):241–250, 2017.

[9] Y. H. Tsai and M. H. Yang. Locality preserving hashing. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 2988–2992, Oct 2014.

[10] Kang Zhao, Hongtao Lu, and Jincheng Mei. Locality preserving hashing. In *AAAI*, pages 2874–2881, 2014.

[11] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[12] L. O. Jimenez and D. A. Landgrebe. Supervised classification in high-dimensional space: geometrical, statistical, and asymptotical properties of multivariate data. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 28(1):39–54, 1998.

[13] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: a comparative review. 10(1-41):66–71, 2009.

[14] Ian T Jolliffe. Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer, 1986.

[15] Aapo Hyvarinen. Survey on Independent Component Analysis. *Neural computing surveys*, 2(4):94–128, 1999.

[16] I K Fodor. *A Survey of Dimension Reduction Techniques*. May 2002.

[17] Haozhe Xie, Jie Li, and Hanqing Xue. A survey of dimensionality reduction techniques based on random projection. *CoRR*, abs/1706.04371, 2017.

[18] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[19] S T Roweis and L K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[20] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[21] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[22] Y. Koren and L. Carmel. Robust linear dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):459–470, July 2004.

[23] Yue Zhao and Jinwen Ma. Local fisher discriminant analysis with locally linear embedding affinity matrix. In *Advances in Neural Networks – ISNN 2013*, pages 471–478. Springer Berlin Heidelberg, 2013.

[24] L. Wang and Q. Mao. Probabilistic dimensionality reduction via structure learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2017.

[25] Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 857–864, 2003.

[26] Miguel A Carreira-Perpinán. The elastic embedding algorithm for dimensionality reduction. In *ICML*, volume 10, pages 167–174, 2010.

[27] Lawrence Cayton and Sanjoy Dasgupta. Robust euclidean embedding. In *Proceedings of the 23rd international conference on machine learning*, pages 169–176. ACM, 2006.

[28] X. Jiang. Linear subspace learning-based dimensionality reduction. *IEEE Signal Processing Magazine*, 28(2):16–26, 2011.

[29] M Turk and A Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[30] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*, volume 46. John Wiley & Sons, 2004.

[31] J Lu, K.N. Plataniotis, and A.N. Venetsanopoulos. Face recognition using LDA-based algorithms. *IEEE Transactions on Neural Networks*, 14(1):195–200, 2003.

[32] Hui Kong, Lei Wang, Eam Khwang Teoh, Xuchun Li, Jian-Gang Wang, and Ronda Venkateswarlu. Generalized 2D Principal Component Analysis for

face image representation and recognition. *Neural Networks*, 18(5):585–594, 2005.

[33] Jian Yang, David Zhang, Alejandro F. Frangi, and Jing-yu Yang. Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(1):131–137, 2004.

[34] Daoqiang Zhang and Zhi-Hua Zhou. (2D) 2PCA: Two-directional two-dimensional PCA for efficient face representation and recognition. *Neurocomputing*, 69(1):224–231, 2005.

[35] B Schölkopf, A Smola, and K R Müller. Kernel Principal Component Analysis. In *in Proceedings of Seventh International Conference on Artificial Neural Networks, ICANN'97*, pages 583–588. Springer, 1997.

[36] Changshui Zhang, Feiping Nie, and Shiming Xiang. A general kernelization framework for learning algorithms based on kernel PCA. *Neurocomput.*, 73(4-6):959–967, 2010.

[37] Vijayakumar Kadappa and Atul Negi. Global modular principal component analysis. *Signal Processing*, 105:381–388, 2014.

[38] Guangming Lu, David Zhang, and Kuanquan Wang. Palmprint recognition using eigenpalms features. *Pattern Recognition Letters*, 24(9-10):1463–1467, 2003.

[39] Clement Fredembach, Michael Schroder, and Sabine Susstrunk. Eigenregions for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1645–1649, 2004.

[40] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[41] Ming Li and Baozong Yuan. 2D-LDA: A statistical linear discriminant analysis for image matrix. *Pattern Recognition Letters*, 26(5):527–532, 2005.

[42] Wei-Shi Zheng, Jian-Huang Lai, and Stan Z Li. 1D-LDA vs. 2D-LDA: When is vector-based linear discriminant analysis better than matrix-based? *Pattern Recognition*, 41(7):2156–2172, 2008.

[43] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE transactions on pattern analysis and machine intelligence*, 29(1):40–51, 2007.

[44] Vin D Silva and Joshua B Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in neural information processing systems*, pages 721–728, 2003.

[45] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pages 585–591, 2002.

[46] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In *Advances in neural information processing systems*, pages 507–514, 2006.

[47] Lawrence K Saul and Sam T Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of machine learning research*, 4(Jun):119–155, 2003.

[48] Jing Chen and Zhengming Ma. Locally linear embedding: a review. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(07):985–1008, 2011.

[49] C. Yao, Y. F. Liu, B. Jiang, J. Han, and J. Han. LLE Score: A new filter-based unsupervised feature selection method based on nonlinear manifold embedding and its application to image recognition. *IEEE Transactions on Image Processing*, 26(11):5257–5269, Nov 2017.

[50] Olga Kouropteva, Oleg Okun, and Matti Pietikäinen. Incremental locally linear embedding algorithm. In *Image Analysis*, pages 521–530. Springer Berlin Heidelberg, 2005.

[51] Shi-qing Zhang. Enhanced supervised locally linear embedding. *Pattern Recognition Letters*, 30(13):1208–1218, 2009.

[52] Jing Chen and Yang Liu. Locally linear embedding: a survey. *Artificial Intelligence Review*, 36(1):29–48, 2011.

[53] Xiaofei He, Shuicheng Yan, Yuxiao Hu, and Hong-Jiang Zhang. Learning a locality preserving subspace for visual recognition. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 385–392. IEEE, 2003.

[54] Guoxian Yu, Hong Peng, Jia Wei, and Qianli Ma. Enhanced locality preserving projections using robust path based similarity. *Neurocomputing*, 74(4):598–605, 2011.

[55] Fadi Dornaika and Ammar Assoum. Enhanced and parameterless locality preserving projections for face recognition. *Neurocomputing*, 99:448–457, 2013.

[56] Xiaofei He, Shuicheng Yan, Yuxiao Hu, P. Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

[57] D Cai, X He, J Han, and H J Zhang. Orthogonal laplacianfaces for face recognition. *IEEE Transactions on Image Processing*, 15(11):3608–3614, 2006.

[58] Yong Xu, Aini Zhong, Jian Yang, and David Zhang. LPP solution schemes for use with face recognition. *Pattern Recognition*, 43(12):4165–4176, 2010.

[59] Jiwen Lu and Yap-Peng Tan. Regularized locality preserving projections and its extensions for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(3):958–963, 2010.

[60] Zhonglong Zheng, Fan Yang, Wenan Tan, Jiong Jia, and Jie Yang. Gabor feature-based face recognition using supervised locality preserving projection. *Signal Processing*, 87(10):2473–2483, 2007.

[61] Sibao Chen, Haifeng Zhao, Min Kong, and Bin Luo. 2D-LPP: A two-dimensional extension of locality preserving projections. *Neurocomput.*, 70(4-6):912–921, 2007.

[62] M Ravishankar, DR Rameshbabu, et al. Ten-LoPP: tensor locality preserving projections approach for moving object detection and tracking. In *The 9th International Conference on Computing and InformationTechnology (IC2IT2013)*, pages 291–300. Springer, 2013.

[63] Liwei Wang, Yan Zhang, and Jufu Feng. On the Euclidean distance of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1334–1339, Aug 2005.

[64] Haijun Zhang, QM Jonathan Wu, Tommy WS Chow, and Mingbo Zhao. A two-dimensional neighborhood preserving projection for appearance-based face recognition. *Pattern Recognition*, 45(5):1866–1876, 2012.

[65] Pang Ying Han, Andrew Teoh Beng Jin, and Fazly Salleh Abas. Neighbourhood preserving discriminant embedding in face recognition. *Journal of Visual Communication and Image Representation*, 20(8):532–542, 2009.

[66] Sebastian Mika, Bernhard Schölkopf, Alex J Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel PCA and de-noising in feature spaces. In *Advances in neural information processing systems*, pages 536–542, 1999.

[67] Jian Yang, Alejandro F Frangi, Jing-yu Yang, David Zhang, and Zhong Jin. KPCA plus LDA: a complete kernel fisher discriminant framework for feature extraction and recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 27(2):230–244, 2005.

[68] G Feng, D Hu, D Zhang, and Z Zhou. An alternative formulation of kernel LPP with application to image recognition. *Neurocomputing*, 69(13):1733–1738, 2006.

[69] Jian Cheng, Qingshan Liu, Hanqing Lu, and Yen-Wei Chen. Supervised kernel locality preserving projections for face recognition. *Neurocomputing*, 67:443–449, 2005.

[70] Gaston Baudat and Fatiha Anouar. Generalized discriminant analysis using a kernel approach. *Neural computation*, 12(10):2385–2404, 2000.

[71] Zhen Lei, Matti Pietikäinen, and Stan Z Li. Learning discriminant face descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):289–302, 2014.

[72] Jie Gui, Zhenan Sun, Wei Jia, Rongxiang Hu, Yingke Lei, and Shuiwang Ji. Discriminant sparse neighborhood preserving embedding for face recognition. *Pattern Recognition*, 45(8):2884–2893, 2012.

[73] Jie Gui, Wei Jia, Ling Zhu, Shu-Ling Wang, and De-Shuang Huang. Locality preserving discriminant projections for face and palmprint recognition. *Neurocomputing*, 73(13-15):2696–2707, 2010.

[74] T Zhang, K Huang, X Li, J Yang, and D Tao. Discriminative orthogonal neighborhood-preserving projections for classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 40(1):253–263, 2010.

[75] Jie Gui, Shu-Lin Wang, and Ying-Ke Lei. Multi-step dimensionality reduction and semi-supervised graph-based tumor classification using gene expression data. *Artificial intelligence in medicine*, 50(3):181–191, 2010.

[76] Feiping Nie, Dong Xu, Ivor Wai-Hung Tsang, and Changshui Zhang. Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *IEEE Transactions on Image Processing*, 19(7):1921–1932, 2010.

[77] Feiping Nie, Shiming Xiang, Yangqiu Song, and Changshui Zhang. Extracting the optimal dimensionality for local tensor discriminant analysis. *Pattern Recognition*, 42(1):105–114, 2009.

[78] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487, 2016.

[79] Yasi Wang, Hongxun Yao, and Sicheng Zhao. Auto-encoder based dimensionality reduction. *Neurocomputing*, 184:232–242, 2016.

[80] E Kokiopoulou and Y Saad. Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2143–2156, 2007.

[81] Z. Lai, W. K. Wong, Y. Xu, J. Yang, and D. Zhang. Approximate orthogonal sparse embedding for dimensionality reduction. *IEEE Transactions on Neural Networks and Learning Systems*, 27(4):723–735, April 2016.

[82] Laurens Van Der Maaten. Accelerating t-SNE using tree-based algorithms. *Journal of machine learning research*, 15(1):3221–3245, 2014.

[83] Yann LeCun and Corinna Cortes. TheMNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1998.

[84] R. Nagar and S. K. Mitra. Feature extraction based on stroke orientation estimation technique for handwritten numeral. In 8^{th} *International Conference on Advances in Pattern Recognition (ICAPR)*, pages 1–6, 2015.

[85] Ujjwal Bhattacharya and BB Chaudhuri. Databases for research on recognition of handwritten characters of indian scripts. In *Proceedings of* 8^{th} *International Conference on Document Analysis and Recognition.*, pages 789–793. IEEE, 2005.

[86] A M Martinez. The AR face database. *CVC Technical Report*, 24, 1998.

[87] Ferdinando S and Harter A. Parameterisation of a Stochastic Model for Human Face Identification . In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*. AT&T Laboratories Cambridge, 1994.

[88] D B Graham and N M Allinson. Characterising virtual eigensignatures for general purpose face recognition. In *Face Recognition*, pages 446–456. Springer, 1998.

[89] Y Adini, Y Moses, and S Ullman. Face recognition: The problem of compensating for changes in illumination direction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):721–732, 1997.

[90] W Chen, M J Er, and S Wu. Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 36(2):458–466, 2006.

[91] S M Pizer, E P Amburn, J D Austin, R Cromartie, A Geselowitz, T Greer, B Romeny, J B Zimmerman, and K Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987.

[92] S Shan, W Gao, B Cao, and D Zhao. Illumination normalization for robust face recognition against varying lighting conditions. In *Analysis and Modeling of Faces and Gestures. IEEE International Workshop on*, pages 157–164. IEEE, 2003.

[93] M Savvides and BVK V Kumar. Illumination normalization using logarithm transforms for face authentication. In *Audio-and Video-Based Biometric Person Authentication*, pages 549–556. Springer, 2003.

[94] A S Georghiades, P N Belhumeur, and D J Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[95] T Sim, S Baker, and M Bsat. The CMU pose, illumination, and expression database. In *Automatic Face and Gesture Recognition. Proceedings. Fifth IEEE International Conference on*, pages 46–51. IEEE, 2002.

[96] E Krieger, VK Asari, and S Arigela. Color image enhancement of low-resolution images captured in extreme lighting conditions. In *SPIE Sensing Technology+ Applications*, pages 91200Q–91200Q. International Society for Optics and Photonics, 2014.

[97] S Arigela and VK Asari. Self-tunable transformation function for enhancement of high contrast color images. *Journal of Electronic Imaging*, 22(2):023010–023010, 2013.

[98] X Tan and B Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *Image Processing, IEEE Transactions on*, 19(6):1635–1650, 2010.

[99] Wan-Jui Lee, R. P. W. Duin, A. Ibba, and M. Loog. An experimental study on combining euclidean distances. In *2010 2nd International Workshop on Cognitive Information Processing*, pages 304–309, June 2010.

[100] Dick de Ridder, Olga Kouropteva, Oleg Okun, Matti Pietikäinen, and Robert P. W. Duin. Supervised locally linear embedding. In *Proceedings of the 2003 Joint International Conference on Artificial Neural Networks and Neural Information Processing*, pages 333–341. Springer-Verlag, 2003.

[101] Kanghua Hui and Chunheng Wang. Clustering-based locally linear embedding. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[102] Feiping Nie, Xiaoqian Wang, and Heng Huang. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 977–986. ACM, 2014.

[103] Guihua Wen, Lijun Jiang, Jun Wen, and Nigel Shadbolt. Performing locally linear embedding with adaptable neighborhood size on manifold. *PRICAI 2006: Trends in Artificial Intelligence*, pages 985–989, 2006.

[104] S. Feinberg. The analysis of crossclassified categorical data, 2nd ed., mit press, cambridge, 1985.

[105] Lingxiao Zhao and Zhenyue Zhang. Supervised locally linear embedding with probability-based distance for classification. *Computers & Mathematics with Applications*, 57(6):919–926, 2009.

[106] Hanyang Liu, Junwei Han, and Feiping Nie. Semi-supervised orthogonal graph embedding with recursive projections. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2308–2314. AAAI Press, 2017.

[107] Feiping Nie, Shiming Xiang, Yangqiu Song, and Changshui Zhang. Orthogonal locality minimizing globality maximizing projections for feature extraction. *Optical Engineering*, 48(1):017202, 2009.

[108] Tolga Tasdizen. Principal neighborhood dictionaries for nonlocal means image denoising. *IEEE Transactions on Image Processing*, 18(12):2649–2660, 2009.

[109] Gitam Shikkenawis. *Similarity Preserving Dimensionality Reduction for Image Data*. PhD thesis, Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar-India, 2016.

[110] J Duchene and S Leclercq. An optimal transformation for discriminant and principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):978–983, 1988.

[111] Hua Wang, Feiping Nie, and Heng Huang. Robust distance metric learning via simultaneous l1-norm minimization and maximization. In *International Conference on Machine Learning*, pages 1836–1844, 2014.

[112] Hong Chang and Dit-Yan Yeung. Robust locally linear embedding. *Pattern recognition*, 39(6):1053–1065, 2006.

[113] Qifa Ke and Takeo Kanade. Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 739–746. IEEE, 2005.

[114] Nojun Kwak. Principal component analysis based on l1-norm maximization. *IEEE transactions on pattern analysis and machine intelligence*, 30(9):1672–1680, 2008.

[115] Fujin Zhong and Jiashu Zhang. Linear discriminant analysis based on l1-norm maximization. *IEEE Transactions on Image Processing*, 22(8):3018–3027, 2013.

[116] H. Wang, X. Lu, Z. Hu, and W. Zheng. Fisher discriminant analysis with l1-norm. *IEEE Transactions on Cybernetics*, 44(6):828–842, 2014.

[117] X. Li, Y. Pang, and Y. Yuan. L1-norm-based 2dpca. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(4):1170–1175, Aug 2010.

[118] Chris Ding, Ding Zhou, Xiaofeng He, and Hongyuan Zha. R1-pca: rotational invariant l 1-norm principal component analysis for robust subspace factorization. *Proceedings of the 23rd international conference on Machine learning*, pages 281–288, 2006.

[119] A Baccini, Ph Besse, and A De Falguerolles. A l1-norm pca and a heuristic approach. 1996.

[120] Qifa Ke and Takeo Kanade. Robust subspace computation using l1 norm. 2003.

[121] R. A. Fisher. UCI repository of machine learning databases – iris data set. ftp://ftp.ics.uci.edu/pub/machine-learning-databases/iris, 1999.

[122] Xiaofeng Yuan, Zhiqiang Ge, Lingjian Ye, and Zhihuan Song. Supervised neighborhood preserving embedding for feature extraction and its application for soft sensor modeling. *Journal of Chemometrics*, 30(8):430–441, 2016.

[123] Cong Li, Weimin Bao, Luping Xu, Hua Zhang, and Yan Bo. Multiple kernel learning via orthogonal neighborhood preserving projection and maximum margin criterion method for synthetic aperture radar target recognition. *Optical Engineering*, 57(5):053105, 2018.

CHAPTER A
publications

- **Journal:**

  1. Purvi A. Koringa, and Suman K. Mitra, "ONPPn: Orthogonal Neighborhood Preserving Projection with Normalization and its Applications" *Image and Vision Computing, Elsevier*, volume 76(C) pp. 64-75, August 2018.
  2. Purvi A. Koringa, and Suman K. Mitra, "L1-norm Orthogonal Neighborhood Preserving Projection and its Applications" available online, July 2018 *Pattern Analysis and Applications, Springer*

- **Conferences:**

  1. Purvi A. Koringa, Gitam Shikkenawis, Suman K. Mitra, S. K. Parulekar, "Modified Orthogonal Neighborhood Preserving Projection for Face Recognition," in *Proceedings of the Pattern Recognition and Machine Intelligence: PReMI 2015*. LNCS 9124, pp. 225-235, Warsaw, Poland, 2015. Springer.
  2. Purvi A. Koringa, Suman K. Mitra, and Vijayan K. Asari. "Handling Illumination Variation: A Challenge for Face Recognition," in *Proceedings of International Conference on Computer Vision and Image Processing: CVIP 2016, Volume 2* pp. 273-283. Roorkee, India, 2016. Springer.
  3. Purvi A. Koringa and Suman K. Mitra. "Orthogonal Neighborhood Preserving Projection using L1-norm Minimization." in *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods* pp. 165-172, ICPRAM 2017, Porto, Portugal, February, 2017.

SCITEPRESS.

4. Purvi A. Koringa and Suman K. Mitra. ″Class Similarity based Orthogonal Neighborhood Preserving Projection for Image Recognition" *manuscript submitted to conference*