Multiscale Principal Component Analysis

Thesis submitted for the degree of

Doctor of Philosophy

at the university of Leicester

By

Akinduko Ayodeji Akinwumi

Department of Mathematics

University of Leicester

August 2015

Abstract

The problem of approximating multidimensional data with objects of lower dimension is a classical problem in complexity reduction. It is important that data approximation capture the structure(s) and dynamics of the data, however distortion to data by many methods during approximation implies that some geometric structure(s) of the data may not be preserved during data approximation. For methods that model the manifold of the data, the quality of approximation depends crucially on the initialization of the method. The first part of this thesis investigates the effect of initialization on manifold modelling methods. Using Self Organising Maps (SOM) as a case study, we compared the quality of learning of manifold methods for two popular initialization methods; random initialization and principal component initialization. To further understand the dynamics of manifold learning, datasets were further classified into linear, quasilinear and nonlinear.

The second part of this thesis focuses on revealing geometric structure(s) in high dimension data using an extension of Principal Component Analysis (PCA). Feature extraction using (PCA) favours direction with large variance which could obfuscate other interesting geometric structure(s) that could be present in the data. To reveal these intrinsic structures, we analysed the local PCA structures of the dataset. An equivalent definition of PCA is that it seeks subspaces that maximize the sum of pairwise distances of data projection; extending this definition we define localization in term of scale as maximizing the sum of weighted squared pairwise distances between data projections for various distributions of weights (scales). Since for complex data various regions of the dataspace could have different PCA structures, we also define localization with regards to dataspace. The resulting local PCA structures were represented by the projection matrix corresponding to the subspaces and analysed to reveal some structures in the data at various localizations.

Acknowledgment

I will like to appreciate my supervisor and scientific father Professor Alexander Gorban, for his invaluable support and encouragement throughout my study. He is indeed a rare gem and perhaps the best scientist I have ever met. I hope that some day I will make you really proud.

I will also like to thank Professor O.K. Koriko for being an inspiration to me as an undergraduate student in mathematics, and also Dr O.A Fasoranbaku for encouraging me to further my studies. My special thanks to Dr Evgeny Mirkes for providing the software and support for the early part of my thesis and to the Department of Mathematics, University of Leicester for providing funds for my study.

Also my appreciation goes to my mother (Victoria Akinduko), late father (Ven. Moses A. Akinduko) and my siblings (Bayo, Akin, Yemi, and Doyin) for their numerous supports. Special thanks to my parents in law (Oluwasanya). To my most wonderful wife (Remi) and adorable son (Ayomide), words are insufficient to express my sincere appreciation. Finally my special thanks to my Lord and Saviour Jesus Christ for everything.

Table of Contents

Chapter 1

Introduction

1.1 A brief History

In an effort to study, understand and improve the various systems we interact with as human, we often generate huge data which we need to extract knowledge from. These data are usually multivariate (multidimensional) distribution of vectors which represent certain observed attributes of the system we seek to understand. We now live in a data driven world and with increasing advances in technology comes increasing complexity in the nature of data collected.

Multi-dimensional data are usually difficult to visualize, analyse and model. Mathematical models which depend on high-dimensional data usually suffer from what Bellman (1961) termed as the curse of dimensionality [10]. Therefore there is a need for approximating high-dimensional vector distributions by lower dimensional objects in such a way that relevant structures and dynamics are preserved. The choice of relevant structures and dynamics which are to be preserved by the data approximation is subjective and this subjectivity has led to the development of various methods for approximating high dimensional data.

In 1901, using a geometric approach, Karl Pearson proposed approximating high dimensional data with lines and planes which 'best fit' the data and thus invented the Principal Component Analysis (PCA) [85]. Another approach to data approximation is to represent complex multidimensional data by a smaller set of finite points, leading to methods like *k-means* which approximate data with several 'mean' points. In the last few decades there have been considerable developments in these two major directions.

Though it could be said that Karl Pearson invented PCA in his paper [85], Harold Hotelling (1933) also independently derived the PCA in his paper [54]. It should be mentioned however that the fundamental ideas of PCA have been developed by mathematicians much earlier. While Hotelling started from the idea of factor analysis and choose factors (which he called components) that successively maximize their

contribution to the variance of the original data, his derivation was PCA and not factor analysis as generally agreed these days. Due to the lack of computational resources, PCA was not of much use in its early years after development. However, the advent of computers and advancement in technology has made such computational resources readily available, making the application of PCA to increase exponentially over the last two decades.

Further development of PCA was done by Girshick (1936) in his papers [37], [38]; He gave an alternative derivation of PCA and discussed the asymptotic sampling distribution of the coefficients and variances of sample principal components. Further theoretical development on asymptotic sampling distribution of the coefficients and variances of the sample principal components was done by Anderson (1963) in his paper [5], building on the earlier work by Girshick. Also of importance is the paper by Rao (1964), which discussed several ideas concerning applications, interpretations and extensions of PCA [90].

PCA has been re-invented in other fields with different names. Some examples of such are Karhunen-Loève decomposition in signal processing [61], [75], Hotelling transform based on [54], Proper Orthogonal Decomposition in the field of mechanics [77], Spectral decomposition in noise and vibration and others.

1.2 Definition and Derivation of PCA

In this section we consider four classical approaches to PCA which are equivalent as given by [40] and we also give the necessary mathematical background that will be needed for this thesis.

1.2.1 Definitions of PCA

Let L_k be a linear manifold of dimension k *given in the* parametric form as $L_k = \{\mathbf{v}_0 + a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \ldots + a_k\mathbf{v}_k\}$, where $a_i \in \mathbb{R}$, $\mathbf{v}_0 \in \mathbb{R}^m$ and $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ is a set of orthonormal vectors in $\mathbb{R}^m$.

Let $\mathbf{x}_i \in \mathbb{R}^m$ be data elements, we will let $\mathbf{x}_{i\alpha}$ represent the value of the αth variable for the ith observation, where $i = 1,2,\ldots,n$ and $\alpha = 1,2,\ldots,m$. For this thesis, the coordinates

will be represented by Greek indices while the observations will be represented by Latin indices. We let X denote an $n \times m$ matrix whose (i,α)th element is $\mathbf{x}_{i\alpha}$. That is

$$X = \begin{pmatrix} \mathbf{x}_{11} & \mathbf{x}_{12} & \cdots & \mathbf{x}_{1m} \\ \mathbf{x}_{21} & \mathbf{x}_{22} & \cdots & \mathbf{x}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{n1} & \mathbf{x}_{n2} & \cdots & \mathbf{x}_{nm} \end{pmatrix}.$$

For all computations, we assume that the data is centered; this can be achieved by simple translation of the data.

For any pair of vectors $\mathbf{x}$ and $\mathbf{y}$, we define the distance function $dist(\mathbf{x},\mathbf{y})$ such that the following axioms are satisfied:

i. $dist(\mathbf{x},\mathbf{y}) \geq 0$ and $dist(\mathbf{x},\mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$ (nonnegative)

ii. $dist(\mathbf{x},\mathbf{y}) = dist(\mathbf{y},\mathbf{x})$ (symmetry property)

iii. $\text{dist}(\mathbf{x},\mathbf{z}) \leq \text{dist}(\mathbf{x},\mathbf{y}) + \text{dist}(\mathbf{y},\mathbf{z})$ (triangle inequality)

The orthogonal projection denoted by $P_Y(\mathbf{x})$ is defined for an object $\mathbf{x}$ and a set of vectors Y as a vector in Y which minimizes $dist(\mathbf{x},\mathbf{y})$, $\mathbf{y} \in Y$. That is $P_Y(\mathbf{x}) = \arg\min_{\mathbf{y}\in Y} dist(\mathbf{x},\mathbf{y})$.

When $dist(\mathbf{x},\mathbf{y})$ is the Euclidean distance, it can be shown that the the orthogonal porojection of data $\mathbf{x}_i$, $i = 1,2,\ldots,n$ to the plane L_k denoted by $P_L(\mathbf{x}) = \sum_{\alpha=1}^{k} \mathbf{v}_\alpha \langle \mathbf{v}_\alpha, \mathbf{x} \rangle$. Where the inner product between any pair of vectors $\mathbf{a}$ and $\mathbf{b} \in \mathbb{R}^m$ is given as $\langle \mathbf{a},\mathbf{b} \rangle = \sum_{i=1}^{m} a_i b_i$.

In his paper "On Lines and Planes of Closest Fit to Systems of Points in Space", Karl Pearson (1901) introduced the first approach to PCA . He proposed approximating multidimensional data by line or plane of 'best fit'. He argues that a good fit will be line or plane that minimizes the sum of the squares distance of the dataset to its orthogonal projection onto the line or plane. This led to definition 1.

Definition 1 (Data approximation by lines and planes).

Given a dataset *X,* PCA computes the sequences of linear manifolds $L_k, (k = 1,2,\ldots,m-1)$ embedded in $\mathbb{R}^m$ such that the sum of squared distances from data points in *X*, to their orthogonal projections on L_k is minimal over all linear manifolds of dimension k.

In other words PCA solves the problem given below as

$$MSD(X, L_k) \to \min \quad (k = 1,2,\ldots,m-1),$$

where $MSD(X,Y)$ denotes the mean squared distance between the dataset X and the set Y and defined as $MSD(X,Y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} dist^2(\mathbf{x}_i, P_Y(\mathbf{x}_i))}, \quad \mathbf{x}_i \in X$.

Using the Euclidean distance, this can be stated as minimize

$$D_X = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i - P_L(\mathbf{x}_i)\|_2^2 , \tag{1.1}$$

where $P_L(\mathbf{x}) = \sum_{\alpha=1}^{k} \mathbf{v}_\alpha \langle \mathbf{v}_\alpha, \mathbf{x} \rangle$, $k < m$ and $\langle \mathbf{v}_\alpha, \mathbf{v}_\beta \rangle = \delta_{\alpha\beta}$ (δ is Kronecker delta).

$$D_X = \frac{1}{n}\sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{\alpha=1}^{k} \langle \mathbf{x}_i, \mathbf{v}_\alpha \rangle \mathbf{v}_\alpha \right\|_2^2 \tag{1.2}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \left(\langle \mathbf{x}_i, \mathbf{x}_i \rangle - \sum_{\alpha=1}^{k} \langle \mathbf{x}_i, \mathbf{v}_\alpha \rangle^2 \right). \tag{1.3}$$

The cross terms are zero since $\langle \mathbf{v}_\alpha, \mathbf{v}_\beta \rangle = 0$ for $\alpha \neq \beta$. Since $\sum_{i=1}^{n} \langle \mathbf{x}_i, \mathbf{x}_i \rangle$ is the sum of the length of the data elements which is positive, minimizing (1.3) reduces to maximizing

$$\frac{1}{n}\sum_{i=1}^{n} \left(\sum_{\alpha=1}^{k} \langle \mathbf{x}_i, \mathbf{v}_\alpha \rangle^2 \right) \tag{1.4}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{\alpha=1}^{k} \mathbf{v}_\alpha{}^T \mathbf{x}_i \mathbf{x}_i{}^T \mathbf{v}_\alpha \tag{1.5}$$

$$= \frac{1}{n}\sum_{\alpha=1}^{k} \mathbf{v}_\alpha{}^T \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i{}^T \mathbf{v}_\alpha \tag{1.6}$$

$$= \sum_{\alpha=1}^{k} \mathbf{v}_\alpha{}^T \operatorname{cov}(X) \mathbf{v}_\alpha . \tag{1.7}$$

Where $\operatorname{cov}(X) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i{}^T$ is the empirical covariance matrix of the data matrix X which is a symmetric positive semi definite matrix. Therefore finding the lines and planes of best fit (in the sense of equation 1.1) to a system of data points reduces to finding the vector that maximizes the quadratic form (1.7). This is maximized by choosing the

vectors $\mathbf{v}_\alpha, \alpha = 1,\ldots,k$ to be the *k* eigenvectors corresponding to the largest *k* eigenvalues of the covariance matrix of X (see Theorem 1.1).

Theorem 1.1: *Let A be an $m \times m$ symmetric matrix with real entries and let the sorted eigenvalues be given by $\lambda_1, \geq \lambda_2 \geq \ldots. \geq \lambda_m$ with $\mathbf{e}_1,\ldots,\mathbf{e}_m$ being the corresponding eigenvectors. Then $\mathbf{e}_1,\ldots,\mathbf{e}_k$, $k = 1,2,\ldots,m,$ is a maximizer of the constrained maximization problem*

$$\max_{\mathbf{u}_1,\ldots,\mathbf{u}_k} \sum_{\alpha=1}^{k} \mathbf{u}_\alpha^T A \mathbf{u}_\alpha$$
$$\text{Subject to: } \left\langle \mathbf{u}_\alpha, \mathbf{u}_\beta \right\rangle = \delta_{\alpha\beta}, \ \alpha,\beta = 1,2,\ldots k. \tag{1.8a}$$

If we arrange the vectors $\mathbf{u}_\alpha$, $\alpha = 1,\ldots,k$, as the columns of a matrix $B = [\mathbf{u}_1\ \mathbf{u}_2\ \ldots \mathbf{u}_k]$, the optimization problem (1.8a) can be reformulated as

$$\max_B \left(trace\left(B^T A B \right) \right)$$
$$\text{Subject to: } B^T B = I. \tag{1.8b}$$

where *I* is the $k \times k$ identity matrix.
The proof of (1.8b) is available in [58] and also presented below.

***Proof**:*

The eigenvectors $\mathbf{e}_1,\ldots,\mathbf{e}_m$ of A form a basis for the *m*-dimensional space, therefore we can write $\mathbf{u}_\alpha$ as a linear combination of $\mathbf{e}_1,\ldots,\mathbf{e}_m$. That is $\mathbf{u}_\alpha = \sum_{i=1}^{m} \mu_{i\alpha} \mathbf{e}_i$, $\alpha = 1,\ldots,k$, where $\mu_{i\alpha}$ are appropriately defined coefficients. Let E be the matrix whose *k*-th column is the *k*-th eigenvector of matrix A such that the *k*-th eigenvector corresponds to the *k*-th largest eigenvalue. Then we can write $B = EC$, where C is the $m \times k$ matrix with $(i,\alpha) - th$ element $C_{i\alpha}$.

Since A is a symmetric matrix with real entries, the matrix of its eigenvectors E diagonalizes it. That is $E^T A E = \Lambda$, where Λ is a diagonal matrix. Therefore we have that $B^T A B = C^T E^T A E C = C^T \Lambda C$. Let C_i be the *i*th row vector of the matrix C, then the objective function in (1.8b) can be written as

$$trace\left(B^T AB\right) = trace\left(C^T \Lambda C\right)$$

$$= \sum_{i=1}^{m}\sum_{\alpha=1}^{k} \lambda_i C_{i\alpha}^2 \,. \tag{1.81}$$

Now since B is orthogonal and the columns of E are orthonormal, we have that $C = E^T B$ and $C^T C = B^T EE^T B = B^T B = I$. Therefore the columns of C are also orthonormal and

$$\sum_{i=1}^{m}\sum_{\alpha=1}^{k} C_{i\alpha}^2 = k. \tag{1.82}$$

Let D be an $m \times m$ orthogonal matrix such that C is the first *k* columns of D. Now the rows of D are orthonormal and satisfy $D_i D_i^T = 1,\ i = 1,...,m$, where D_i is the *i*th row vector of matrix D. Since the rows of matrix C consist of the first *k* elements of the rows of D, it follows that $C_i C_i^T \le 1,\ i = 1,...,k$, that is

$$\sum_{\alpha=1}^{k} C_{i\alpha}^2 \le 1. \tag{1.83}$$

Now $\sum_{\alpha=1}^{k} C_{i\alpha}^2$ is the coefficient of λ_i in equation (1.81). From equation (1.82) we have that the sum of these coefficients is *k,* and from equation (1.83) none of the coefficient can exceed 1. Since we have that $\lambda_1 \ge \lambda_2 \ge .... \ge \lambda_k$, then $\sum_{i=1}^{m}\left(\sum_{\alpha=1}^{k} C_{i\alpha}^2\right)\lambda_i$ will be maximized if we can find a set of $C_{i\alpha}$ for which

$$\sum_{\alpha=1}^{k} C_{i\alpha}^2 = \begin{cases} 1, & i = 1,...,k, \\ 0, & i = k+1,...,m. \end{cases} \tag{1.84}$$

If we choose $C_{i\alpha} = \delta_{i\alpha},\ i = 1,...,k$ where $\delta_{i\alpha}$ is Kronecker delta, the condition given in (1.84) will be satisfied and B will be the first *k* columns of E.

Therefore $trace\left(B^T AB\right)$ achieves it maximum value if B is chosen as the first *k* column of the matrix E whose columns are the eigenvectors corresponding to the *k* largest eigenvalue of matrix A. In other words $\mathbf{e}_1,...,\mathbf{e}_k$, is a maximizer of the constrained maximization problem (1.8a).

Now we present the second approach to PCA. In approximating multidimensional data with objects of lower dimension, it is often desirable to retain as much variation of the data as possible. Line and plane such that the variance of data

projection is large compared to the variation in the data is usually considered to be informative direction.

Definition 2 (Variance maximization).

For a dataset X and for a given vector $\mathbf{v}_\alpha$, Let us construct a one-dimensional distribution $\mathbf{B}_\alpha = \{\beta : \beta = \langle \mathbf{x}, \mathbf{v}_\alpha \rangle, \mathbf{x} \in X\}$. If we define the empirical variance of X along $\mathbf{v}_\alpha$ as $Var(\mathbf{B}_\alpha)$, where $Var()$ is the standard empirical variance, then PCA seeks to find such L_k that the sum of empirical variances of X along $\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k\}$ would be maximal over all linear manifolds of dimension k embedded in R^m.

The problem PCA solves is given as maximize

$$\sum_{\alpha=1,...k} Var(\mathbf{B}_\alpha) \tag{1.9}$$

$$= \frac{1}{n} \sum_{\alpha=1}^{k} \sum_{i=1}^{n} \langle \mathbf{x}_i, \mathbf{v}_\alpha \rangle^2 \tag{1.10}$$

We observe that equation 1.10 is the same as equation 1.4, therefore this reduces to equation 1.7 as given below:

$$\sum_{\alpha=1}^{k} \mathbf{v}_\alpha{}^T \operatorname{cov}(X) \mathbf{v}_\alpha \to max$$

under the condition that $\langle \mathbf{v}_\alpha, \mathbf{v}_\beta \rangle = \delta_{\alpha\beta}$. From theorem 1.1, this is maximized by choosing the vectors $\mathbf{v}_\alpha, \alpha = 1,...,k$ to be the k eigenvectors corresponding to the largest k eigenvalues of the covariance matrix of X. In addition to this, the variance $\mathbf{B}_\alpha$, $\alpha = 1,...,k$ is given by the eigenvalue of the covariance matrix with respect to eigenvector $\mathbf{v}_\alpha$.

When analyzing a high dimensional data, usually some or even a large number of the variables under study are interrelated, leading to redundancy in the data. Sometimes it becomes desirable to re-express the data such that the transformed data eliminates redundancy. The solution to this problem using linear transformation is the third classical approach that leads to PCA.

***Definition 3* (*correlation cancellation*):**

Given a dataset *X,* PCA seeks such an orthonormal basis $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ in which the covariance matrix for the projection of X is diagonal.

Evidently, in this basis the distributions $\langle \mathbf{v}_\alpha, \mathbf{x} \rangle$ and $\langle \mathbf{v}_\beta, \mathbf{x} \rangle$, for $\alpha \neq \beta$, have zero correlation where the correlation here is the Pearson product–moment correlation coefficient defined as $\frac{E\left[\langle \mathbf{v}_\alpha, \mathbf{x} \rangle \langle \mathbf{v}_\beta, \mathbf{x} \rangle\right]}{\sigma_{\langle \mathbf{v}_\alpha, \mathbf{x} \rangle}, \sigma_{\langle \mathbf{v}_\beta, \mathbf{x} \rangle}} = 0$, E is the expected value operator and σ_a is the standard deviation of variable *a.*

Given a $n \times m$ data matrix X, let the matrix P be a $m \times m$ transformation matrix. We seek a new matrix $Y = XP$ such that covariance matrix of Y is diagonal,

$$Cov(Y) = \frac{1}{n} Y^T Y = \frac{1}{n} P^T X^T X P = P^T Cov(X) P \tag{1.11}$$

We know that the covariance matrix is a symmetric positive definite matrix which is orthogonally diagonalizable and it is diagonalized by the matrix of its orthonormal eigenvectors E (i.e E is the matrix whose columns are the eigenvectors of covariance matrix). Therefore

$$Cov(Y) = P^T Cov(X) P = P^T (E^T D E) P \ , \tag{1.12}$$

where D is a diagonal matrix consisting of the eigenvalues corresponding to the eigenvectors. Since the objective is to diagonalize $Cov(Y) = P^T (E^T D E) P$, then choosing $P = E^T$ gives

$$P^T (E^T D E) P = E^T (E D E^T) E = D. \tag{1.13}$$

This satisfies the objective.

This result is the same as previous definitions. This implies that the dataset is de-correlated by projecting it to the direction of the eigenvectors of its covariance matrix, which is equivalent to finding the principal component of the data matrix X. We note the following:

1) the notion of correlation is basis-dependent (i.e. data can be correlated in one basis and uncorrelated in another),
2) PCA de-correlate the dataset irrespective of the underlying distribution.

3) The diagonalization $\text{cov}(X) = (E^T DE)$ is not unique, as one can have different choices of orthonormal bases for those eigenspaces with dimension (geometric multiplicity) greater than one.

From linear algebra we know that orthogonal projections onto lower-dimensional space lead to contraction of all point-to-point distances (except for some that do not change). The fourth approach is the problem of approximating our data by projection onto lower dimensional subspace with the objective of maximizing the mean point-to-point squared distances of data projection.

***Definition 4* (*mean point-to-point squared distance maximisation*)**

Therefore, PCA seeks to find such sequence $L_k, (k = 1,2,\ldots,m-1)$ such that the mean point-to-point squared distances between the orthogonal projections of data points on L_k is maximal over all linear manifolds of dimension k embedded in $\mathbb{R}^m$. That is we seek to maximize

$$\frac{1}{n}\sum_{i,j=1}^{n} dist^2(P_L\mathbf{x}_i, P_L\mathbf{x}_j).$$

The distance function $dist(\mathbf{x},\mathbf{y})$ is taking to be the Euclidean distance, therefore the problem can be stated as

$$D_X = \sum_{i<j} \| P_L(\mathbf{x}_i - \mathbf{x}_j) \|_2^2 \longrightarrow \max, \tag{1.14}$$

Where (1.14) can be re-expressed as

$$D_X = \sum_{i<j}\left[\sum_{\alpha=1}^{k}\left\langle \mathbf{v}_\alpha, (\mathbf{x}_i - \mathbf{x}_j)\right\rangle^2\right] \tag{1.15}$$

$$= \sum_{\alpha=1}^{k}\left[\sum_{i<j}\left\langle \mathbf{v}_\alpha, (\mathbf{x}_i - \mathbf{x}_j)\right\rangle^2\right]. \tag{1.16}$$

The expression in the bracket given as

$$\sum_{i<j}\left\langle \mathbf{v}_\alpha, (\mathbf{x}_i - \mathbf{x}_j)\right\rangle^2 = \sum_{i<j}\left\langle \mathbf{v}_\alpha, (\mathbf{x}_i - \mathbf{x}_j)\right\rangle\left\langle (\mathbf{x}_i - \mathbf{x}_j), \mathbf{v}_\alpha)\right\rangle$$

$$= \mathbf{v}_\alpha^T \tilde{S}\mathbf{v}_\alpha. \tag{1.17}$$

where

$$\tilde{S} = \sum_{i<j} \left[(\mathbf{x}_i - \mathbf{x}_j) \otimes (\mathbf{x}_i - \mathbf{x}_j) \right],$$

and each element of $\tilde{S}$ is given as

$$\tilde{S}_{\alpha\beta} = \sum_{i<j} \left[(\mathbf{x}_{i\alpha} - \mathbf{x}_{j\alpha})(\mathbf{x}_{i\beta} - \mathbf{x}_{j\beta}) \right]. \tag{1.18}$$

The matrix $\tilde{S}_{ij}$ is symmetric positive semi-definite, because for every y, $y \otimes y$ is positive semi-definite.

Therefore the problem given by 1.14 can be stated as

$$\max_{v_1,\ldots,v_k} \sum_{\alpha=1}^{k} \mathbf{v}_\alpha^T \tilde{S} \mathbf{v}_\alpha \,. \tag{1.19}$$

$$\text{Subject to } (\mathbf{v}_\alpha, \mathbf{v}_\beta) = \delta_{\alpha\beta} \quad \alpha,\beta = 1,2,..,k \,.$$

where the constraint is from the orthonormality condition on P_L. Now it is left to show that the solution to the problem 1.19 is actually the principal components. To show this, we introduce lemma 1.1

Lemma 1.1: *The matrix* $\tilde{S}$ *and* $\mathrm{cov}(X)$ *are identical up to a positive multiplicative factor,* $\tilde{S} = n^2 \mathrm{cov}(X)$

Proof:

Let us examine the matrix

$$\begin{aligned}
\tilde{S} &= \sum_{i<j} \left[(\mathbf{x}_i - \mathbf{x}_j) \otimes (\mathbf{x}_i - \mathbf{x}_j) \right] = \frac{1}{2} \sum_{i,j=1}^{n} \left[(\mathbf{x}_i - \mathbf{x}_j) \otimes (\mathbf{x}_i - \mathbf{x}_j) \right] \\
&= \frac{1}{2} \left[n \sum_{i=1}^{n} (\mathbf{x}_i \otimes \mathbf{x}_i) + n \sum_{j=1}^{n} (\mathbf{x}_j \otimes \mathbf{x}_j) - \sum_{i,j=1}^{n} (\mathbf{x}_i \otimes \mathbf{x}_j) - (\mathbf{x}_j \otimes \mathbf{x}_i) \right] \\
&= \frac{1}{2} \left[2n \sum_{i=1}^{n} (\mathbf{x}_i \otimes \mathbf{x}_i) - n^2 (\boldsymbol{\mu} \otimes \boldsymbol{\mu}) - n^2 (\boldsymbol{\mu} \otimes \boldsymbol{\mu}) \right].
\end{aligned} \tag{1.20}$$

where $\sum_{i,j=1}^{n} (\mathbf{x}_i \otimes \mathbf{x}_j) = \sum_{i=1}^{n} \mathbf{x}_i \otimes \sum_{j=1}^{n} \mathbf{x}_j$ and $\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i$,

Therefore $\tilde{S} = n \sum_{i=1}^{n} (\mathbf{x}_i \otimes \mathbf{x}_i)$. (1.21)

The remaining terms are zero because the data has been centered. We can write equation 1.21 as

$$\widetilde{S} = n^2 \operatorname{cov}(X). \tag{1.22}$$

Based on lemma 1.1, the optimization problem 1.19 is the same as 1.7 with the $\operatorname{cov}(X)$ replaced by $\widetilde{S} = n^2 \operatorname{cov}(X)$ which is maximized by choosing the vectors $\mathbf{v}_\alpha, \alpha = 1,\ldots,k$ to be the k eigenvectors corresponding to the largest k eigenvalues of the matrix covariance matrix of $\widetilde{S} = n^2.\operatorname{cov}(X)$ (see theorem 1.1). Since the multiplication of a matrix by a positive constant does not change the eigenvectors or their order, the eigenvectors of $\operatorname{cov}(X)$ is also the eigenvectors of $\widetilde{S}$. Therefore the solution to 1.19 is the principal component.

Let us define the distance distortion of the data arising from data approximation as $\sum_{i,j=1}^{n}[dist^2(\mathbf{x}_i,\mathbf{x}_j) - dist^2(P_L\mathbf{x}_i, P_L\mathbf{x}_j)]$. Definition 4 can also be formulated in terms of the distance distortion, that is, PCA seeks the sequence L_k such that the distance distortion in the data is minimized.

$$\sum_{i,j=1}^{n}[dist^2(\mathbf{x}_i,\mathbf{x}_j) - dist^2(P_L\mathbf{x}_i, P_L\mathbf{x}_j)] \longrightarrow min. \tag{1.23}$$

1.3 Some Assumptions and Limitations of PCA

We briefly discuss some underlying assumptions and some limitations of PCA. This will help us to understand the performance of PCA in application to data and how PCA can be extended or adapted for various situations. See [89].

1. Linearity – PCA find the basis that "best" re-expresses the data. It also restricts the set of basis that is considered since the basis must be orthonormal. The principal components are linear combination of the original variables. The development of nonlinear techniques has been based on the motivation that PCA may be inadequate for data in which nonlinearity is involved. For example, in engineering where most problems are nonlinear [111].
2. Directions with large variance are informative - from definition 2, the basis of the principal components is found using sample covariance matrix. However sample covariance matrix suffers the drawback of not being robust to outliers [56], [103]. The presence of outliers in data influences the result of the analysis as the

principal components align with the direction of large variance which in this case will be the direction of the outliers. Since the directions with large variance are assumed to be informative, for data with outliers, the result can be misleading.

3. Principal Components are orthogonal- From definition 3, the objective of PCA is to de-correlate the data (i.e. to remove second-order dependencies). This was achieved by constraining the principal component to be orthogonal. Even though the assumption of orthogonal principal components makes finding principal component easy by using techniques from linear algebra however for data with higher order dependencies, PCA may not reveal all structures in the data. Second order dependencies are only sufficient to reveal dependencies in data for which the first and second order are sufficient statistic [89]. An example of such is data with non-orthogonal axis.
4. Non parametric- PCA is a non-parametric analysis and hence does not incorporate any priori knowledge available. It is independent of any hypothesis about data distribution.

1.4 Properties of PCA

In this section we consider some important optimal algebraic properties of PCA which have statistical implications.

Property 1: For any integer $1 \le k \le m$, consider the orthonormal linear transformation

$$\mathbf{y} = B^T \mathbf{x} \tag{1.24}$$

Where $\mathbf{y}$ is a *k*-element vector and B is a *m*×*k* matrix, and let S_y be the covariance matrix for $\mathbf{y}$, $S_y = B^T SB$, where S is the sample covariance matrix of $\mathbf{x}$. Then the trace of S_y denoted $trace(S_y)$ is maximized by taking $B = E_k$ where E_k consists of the first *k* columns of matrix *E* and the matrix *E* is the matrix of eigenvectors (i.e. the columns of *E* are the eigenvectors of S which we otherwise call the loading vectors).
The statistical implication of this property is that of all *k*-dimensional subspace projection of $\mathbf{x}$, the variance of the projection of $\mathbf{x}$ is maximized by the subspace spanned by the loading vector. The proof of this property is available in [58]. This is the same as definition 2.

Property 2: Consider the orthonormal linear transformation as defined in *property 1*

$$\mathbf{y} = B^T \mathbf{x} \tag{1.25}$$

Then $trace(S_{\mathrm{y}})$ is minimized by taking $B = E_k^*$ where E_k^* consist of the last *k* columns of the matrix of eigenvectors *E*. Similar to *property 1*, the statistical implication of this property is that of all *k*-dimensional subspace projection of X , the variance of the projection of X is minimized by the subspace spanned by the eigenvectors which correspond to the *k*-smallest eigenvalues of S_{y}. Similar proof to *property 1* can be adapted to proof this. The axes along which data projections have minimal variance can be useful in detecting near linear relationship in the data which can be helpful in identifying outliers in the data. This is further discussed in section 1.5.

Property 3: **(**The spectral decomposition of *S*)

The sample covariance matrix *S* can be decomposed as follows:

$$S = \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^T + \ldots + \lambda_m \mathbf{e}_m \mathbf{e}_m^T, \tag{1.26}$$

where $\mathbf{e}_\alpha$ is the eigenvector of *S.*

The statistical implication of this property is that in addition to being able to decompose the combined variances of all the elements of X into decreasing contributions of the principal components, we can also decompose the covariance matrix into the contribution from each principal component even though this is not strictly decreasing.

Property 4: PCA maximizes Mutual Information on Gaussian Data.

Let $\mathbf{x} \sim N(\mathbf{0}, S)$ and let $\mathbf{y} = B^T \mathbf{x}$ be as previously defined, since $\mathbf{y}'s$ are linear combinations of the columns of X then they are normally distributed with zero mean and covariance $S_{\mathrm{y}} = B^T S B$. Since B is deterministic, the conditional entropy $H(\mathbf{y} / \mathbf{x})$ vanishes.

Therefore mutual information

$$I(\mathbf{x}, \mathbf{y}) = H(\mathbf{y}) - H(\mathbf{y} / \mathbf{x}) = H(\mathbf{y}), \tag{1.27}$$

can be approximated with differential entropy,

$$H(\mathbf{y}) = -\int p(\mathbf{y}) \log_2 \, p(\mathbf{y}) d\mathbf{y} = \frac{1}{2} \log_2 (e(2\pi)^k) + \frac{1}{2} \log_2 (\det(S_y)). \tag{1.28}$$

To maximize equation 1.28, it is sufficient to maximize $\det(S_y)$ over all choices of B. This attain its maximal value when $B = E_k$. For further reading see [16], [8] and [22].

1.5 Some Applications of Principal Component Analysis

PCA is a non-parametric eigenvector-based multivariate analysis and it is regarded as one of the most important results from linear algebra. Often used in the data pre-processing step of data analysis or modelling, PCA has been successfully applied to data from numerous fields of human endeavours. PCA is usually performed on dataset in order to achieve various objectives, and in this section, we will look at a broad classification of these objectives.

Feature Extraction:

Let us consider the problem of finding a one dimension representation of a set of datapoints $\mathbf{x}_i \in \mathbb{R}^m, i = 1, \ldots, n$ in which the datapoints is actually distributed along a line l spanned by the unit vector $\mathbf{u}$ and embedded in $\mathbb{R}^m$. Therefore $\mathbf{x}_i = \mu + a_i \mathbf{u}$ for some $a_i \in \mathbb{R}$, where μ is the sample mean, and $\mathbf{u} \in \mathbb{R}^m$. The variance of the data projection along any vector $\mathbf{v} \in \mathbb{R}^m$ of unit length is given as

$$S \equiv \frac{1}{n-1} \sum_{i=1}^{n} \langle \mathbf{x}_i - \mu, \mathbf{v} \rangle^2 = \frac{1}{n-1} \sum_{i=1}^{n} a_i^2 \langle \mathbf{u}, \mathbf{v} \rangle^2 . \tag{1.29}$$

We have that $\langle \mathbf{u}, \mathbf{v} \rangle^2 = cos^2\theta$, where θ is the angle between $\mathbf{u}$ and $\mathbf{v}$ and equation 1.29 is maximized when $\mathbf{v} = \pm\mathbf{u}$, provided that the data has finite variance. In this example, PCA has been used as a tool for feature extraction. We also note that variance along orthogonal direction will be zero. Feature extraction is often used to pre-process a dataset in order to enhance the performance of other analysis and statistical methods.

An example of the use of PCA for feature extraction is in neuroscience where a variant of PCA is used to identify features of a stimulus that increase the probability of a neuron to generate potential actions. This technique is called spike-triggered covariance analysis.

Dimension Reduction

One of the main objectives of developing PCA is as a dimension reduction technique. That is given $\mathbf{x}_i \in \mathbb{R}^m, i = 1, \ldots, n$. We want to find sequences of L_k, usually with $(k << m)$ that best approximate the data in the sense that it retains as much variability in the data as possible while eliminating redundancy in the data. For multivariate data with large number of variables, high correlation or collinearity usually exists (i.e. one variable can be linearly predicted from the others with a good degree of freedom). This phenomenom implies that there is redundancy among some of the variables which can lead to poor performance of some traditional statistical methods [109]. An example of such is the problem of multicollinearity in linear regression [108] which will be discussed later in section 1.9.

Also, in modelling a dataset, usually data samples sparsely populate the dataspace in very high dimension; the sampling density is proportional to $n^{1/m}$ where n is the sample size and m is the dimension of the dataspace. This is a consequence of the curse of dimesionality [10]. It is sometimes desirable to reduce the dimension of the data in order to deal with this problem and also to avoid a situation where the model overfits.

Therefore, PCA is used for dimension reduction to enhance the performance of traditional statistical methods such as regression analysis, discriminant analysis, cluster analysis and canonical correlation analysis. In addition to this, due to the huge data generated by some systems, data storage has become a big challenge and dimension reduction techniques are employed to reduce the size of data to a manageable size for storage while preserving as much information of the data as possible.

De-noising Data

We define noise as unexplained variation or randomness in a dataset. The presence of noise in data further obfuscates the underlying structure(s) of the data. One application of PCA is to remove these distracting variances. Since PCA rotates the coordinate axes

of the data such that much of the variance is concentrated in the first few coordinate axes, the remaining axes are usually dominated by noise. Discarding the axes with very low variance helps to eliminate or reduce the effect of noise. This is the same as using PCA for dimension reduction purposes in which we select $k < m$ such that principal components with large variances are retained while the remaining $m-k$ principal components with very low variances are discarded.

Even though there is no absolute scale for noise, according to [89] a common measure is the signal-to-noise ratio (SNR) given as

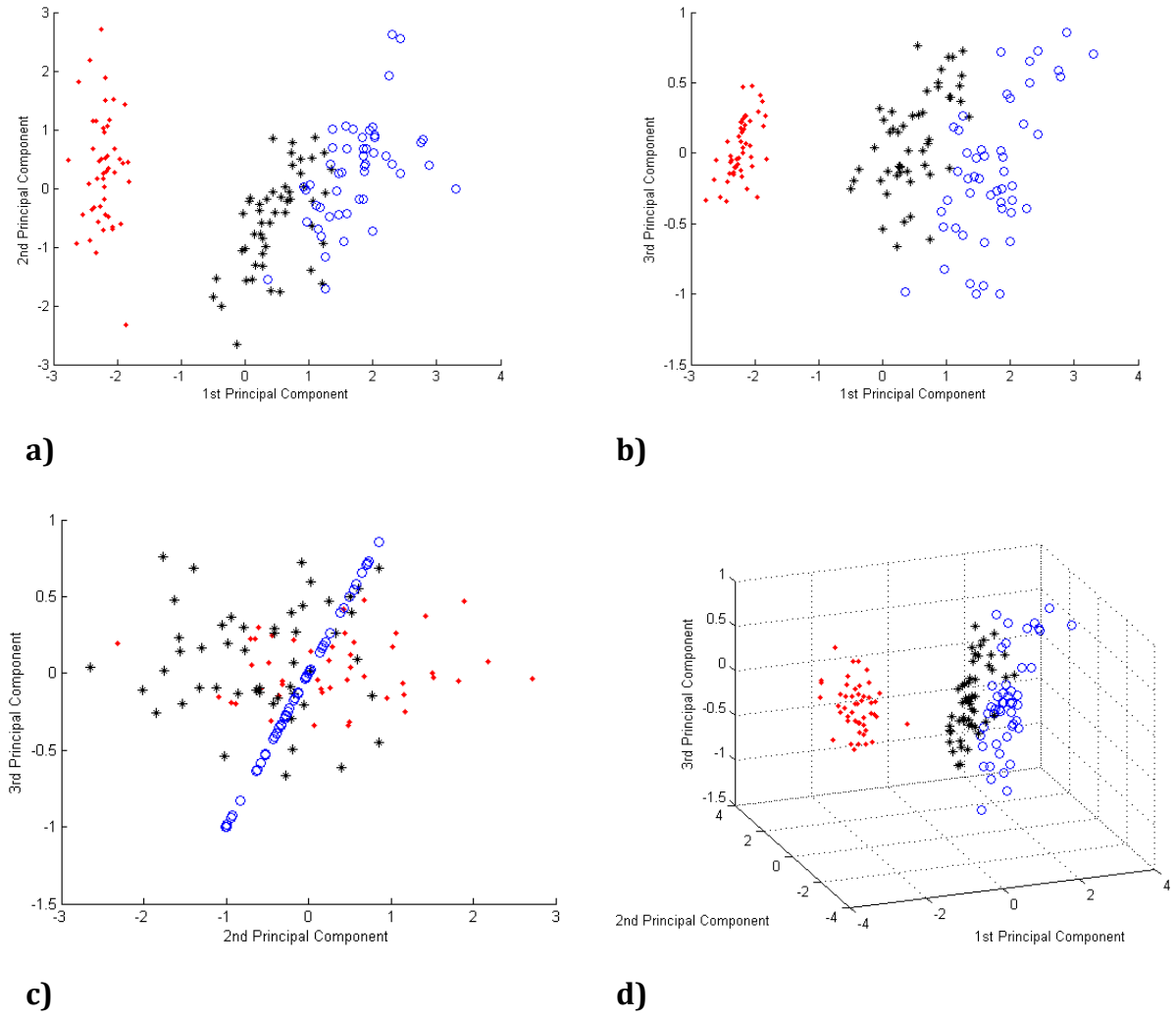
$$SNR = \frac{\sigma^2_{signal}}{\sigma^2_{noise}}.$$

Where σ_a^2 denotes the variance of is variance of a. We should however mention that axes with small variance (which could be classified as noise) sometimes could be useful for some analysis and in addition, the last few principal components can help detect near constant linear relationship in the data. This can be useful in identifying outliers that are not apparent with respect to the original variable. A strong correlation between the variables implies that there are linear functions of the variables with smaller variance than the original variable; such linear functions with small variance help detect outliers which may not be easily detected in the original variable. The situation where the variance along an axis is 0 signifies a constant relationship among the variables which should be removed. Identifying such constant relationship in the data may be difficult, however this is revealed by PCA as axis with zero variance. For further reading on this see section 10.1 of [58].

Data Visualization in Low Dimension

One important step in exploratory data analysis is data visualization, because it helps to interact better with the data. Data visualization becomes difficult when the dimension of the data is more than three, however such data can be approximated with objects of dimension two or three which capture the important structure(s) in the data (for PCA, this is retaining as much variance in the data as possible) and we can visualize this lower dimension approximations of the data.

For example we consider visualizing the Iris flower dataset using PCA. This is a multivariate dataset with and 4 variables and 150 samples (consisting of 50 samples of each of the three species of the Iris flower) and was collected by Sir Ronald Fisher (1936) as an example for discriminant analysis [4]. To visualize this data we project the data onto the loading vectors of the principal components and visualize various 2 and 3 dimensional approximations of the data (See figure 1).



a) **b)**

c) **d)**

Figure 1: Scatter plot of Iris dataset projected onto various principal axes. Datapoints belonging to the same class are shown using the same colour.

Other uses of PCA include assigning weight to the variables in a dataset. Sometimes in data analysis there is a need to attach importance to certain variables, this is usually achieved by multiplying the variables with numbers (weights) which help to emphasize its importance in the analysis. The choice of weights is based on the problem being

solved, however for some analysis; these weights can be chosen as the loading vector of the first principal component. This is because the first principal component is the direction of maximum spread of the data and its loading vector is the coefficient of the linear combination of the original variables that achieve this. One application of this is in equity market where traders sometimes use PCA to weight a portfolio [110]

1.6 Principal Component Analysis and Singular Value Decomposition

Sylvester (1889) in his paper [95] showed that any real matrix M with rank k can be factorized into the form

$$M = UDV^T \tag{1.30}$$

Called Singular Value Decomposition (SVD) of the Matrix M, where U is an $n \times n$ orthogonal matrix, V is an $m \times m$ orthogonal matrix (i.e. $U^T U = I_n, V^T V = I_m$) and D is an $n \times m$ diagonal matrix with non-zero diagonal entries d_i, ($i = 1,2,\ldots,k$) called singular values. The non-zero columns U_α of U are called the right singulars vector while the non-zero columns V_α of V are called the left singular vectors. SVD is a general method for understanding change of basis and it provides an algebraic solution to PCA.

Given that X is a data matrix which is already centered (this can be accomplish by a simple translation of the data); we can define a new matrix $Z = \frac{1}{\sqrt{n-1}} X$ and let $S = Z^T Z$.

We note that $S = \frac{1}{n-1} X^T X$ is the sample covariance matrix of X. The principal components of X are computed using eigenvectors decomposition of S,

$$S = V \Lambda V^T,$$

where the columns V_α of V are eigenvectors of S arranged in order of the magnitude of their corresponding eigenvalue.

It turns out that SVD represent a solid mathematical foundation for PCA (Strang 1993). Consider the SVD of X,

$$Z = UDV^T \tag{1.31}$$

The covariance matrix of X is

$$S = \frac{1}{n-1} X^T X = \frac{1}{n-1} VD^T U^T UDV^T \tag{1.32}$$

$$S = \frac{1}{n-1} VD^2 V^T \tag{1.33}$$

We remark that S and D^2 are similar matrices and therefore V is the matrix whose columns are the eigenvectors of S and the singular values d_i of D are the root of λ_i the eigenvalues of S. Also from equation 1.31,

$$U = ZVD^{-1} \tag{1.34}$$

Where we note that ZV are the principal components of X and since D is diagonal then $U = ZVD^{-1}$ is the scaled version of the principal components where each principal component has been scaled by the singular value d_i.

One interesting result of SVD is given by the Eckart-Young theorem which gives a solution to the problem of approximating a matrix *A* of rank *m* with another matrix *B* of rank *k*, $k < m$.

Theorem 1.2 Eckart-Young theorem

The approximating matrix *B* of rank *k*, $k < m$ to a matrix *A* of rank *m* which has an error matrix with the lowest Frobenius norm is formed by taking the matrix *B* to be the SVD of *A* under the constraint that the lowest $m-k-1$ singular values are set to zero. That is if SVD of $A = UDV^T$, then

$$B = UD^*V^T, \tag{1.35}$$

where D^* is the diagonal matrix with the diagonal elements d_i, such that

$$d_1 \geq d_2 \geq \ldots \geq d_k > d_{k+1} = d_{k+2} = \ldots = d_m = 0. \tag{1.36}$$

Another way to express this is if a_{ij}, b_{ij} and c_{ij} are the elements of A and B (as defined above) respectively, and C is any rank *k* matrix, then Eckart-Young theorem states that the solution to the problem

$$\sum_i \sum_j (a_{ij} - c_{ij})^2 \rightarrow min, \tag{1.37}$$

is given by

$$\sum_i \sum_j (a_{ij} - b_{ij})^2 . \tag{1.38}$$

See [115, 116] for therorem and proof

1.7 Iterative Algorithm for calculating Principal Components

The naïve computation of PCA requires the covariance matrix. Computing the covariance matrix requires $O(nm^2)$ operations and when *n, m* are large this could be computationally expensive. Roweis (1998) developed a method that avoids computing the covariance matrix explicitly [91]. This algorithm was based on the classical expectation/maximization iterative algorithm which was first introduced by [21] and it is of order $O(nmk)$ (where $k \le m$ is the number of principal component of interest). The algorithm is presented below (also see [40], [91]):

1. $\mathbf{v}_0 = M_F(X)$. (i.e. the zero order principal component is the mean point of *X)*. For centered data we set $\mathbf{v}_0 = 0$.
2. Choose randomly $\mathbf{v}_1 \neq 0$
3. Iterate the following steps:
4. Calculate $a_i = \frac{\langle \mathbf{x}_i - \mathbf{v}_0, \mathbf{v}_1 \rangle}{\| \mathbf{v}_1 \|^2}$, *i*=1,...,*n*;
5. Given a_i, find new $\mathbf{v}_1$, such that $\sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{v}_0 - \mathbf{v}_1 a_i)^2 \to \min_{\mathbf{v}_1}$

   i.e. $\mathbf{v}_1 = \frac{\sum_{i=1}^{n} \mathbf{x}_i a_i - \mathbf{v}_0 \sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i^2}$;
6. Re-normalize $\mathbf{v}_1 = \mathbf{v}_1 / \| \mathbf{v}_1 \|$.
7. If the direction of angle $\mathbf{v}_1$ changes by less or equal to some small angle ε then stop else go to step 3

To calculate the second principal components, a deflation approach is applied: after finding $\mathbf{v}_1$, we deflate the data and calculate new $X^{new} = X - \mathbf{v}_0 - \mathbf{v}_1 \langle \mathbf{x}, \mathbf{v}_1 \rangle$ and the algorithm is applied to find the PCA on X^{new}. This procedure is repeated for further principal components.

As can be noted (see [21], [40]), this iterative algorithm indeed performs singular value decomposition of the data and finds the right and left singular vectors one after the other.

The standard convergence proof of [21] applies to this algorithm; therefore it always reaches a local maximum of likelihood. Also, [101] [102] have shown that the only stable local extremum is the global maximum at which the true principal subspace is found. Therefore, the algorithm converges to the result.

Remark: As observed by Gorban et al (2015) [112], any point in high dimension are almost orthogonal to other points and therefore the algorithm above may suffer in high dimension especially at the projection steps.

1.8 PCA, K-means and Principal Objects.

In PCA, data approximation is achieved by the projection of data of high dimension to linear manifold of lower dimension such that $MSD(X,\mathbf{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{N}\left\|\mathbf{x}_i - P_{\mathbf{y}}\mathbf{x}_i\right\|_2^2}$ is minimized as given in definition 1. Another approach is to approximate a dataset by a finite set of points $Y = \{\mathbf{y}_1,\ldots,\mathbf{y}_k\}$ with $k << n$ where each $\mathbf{x}_i \in X$ is approximated by the closest $\mathbf{y}_i \in Y$ this leads to the popular *k-mean* clustering method. This idea date back to Hugo Steinhaus (1957) and was developed further by James Macqueen (1967), Stuart Lloyd (1957) and many extension and adaptation over the years.

Theoretically, PCA and *k-mean* clustering are linked [41]. First we consider the generalization of the notion of mean value as given by Fréchet (1948) [28], as a set which minimizes the mean square distance to the set of data samples.

$$M_F(\mathbf{x}) = \arg\min_{\mathbf{y}\in D} \sum_i dist(\mathbf{y},\mathbf{x}_i)^2 . \tag{1.39}$$

Using the definition above, we can find a mean or a set of means from a space *D (*which does not necessarily coincide with the space of the dataset) provided distance between $\mathbf{d} \in D$ and $\mathbf{x}_i \in X$ can be measured. This mean point or set of means are called principal

objects. We should mentioned that it is not guaranteed that the mean point(s) will be unique.

For example, if we chose D to be the space of *k-element* sets of vectors in R^m *(k-tuple)* and distance between $\mathbf{d} \in D$ and a point $\mathbf{x}_i \in X$ defined as

$$dist(\mathbf{x}_i, \mathbf{y}) = \min_{j=1,..k} \| \mathbf{x}_i - \mathbf{d}_j \|_2 , \tag{1.40}$$

where $\mathbf{d}_j$ is the jth element of the tuple, then the Fréchet mean corresponds to the optimal position of centroids in the *k-mean* clustering method.

From the definition of Fréchet mean, PCA can be seen as the case where D is chosen as the space of linear manifold embedded in R^m and the distance between a point $\mathbf{x}_i \in X$ and D is defined as the distance between $\mathbf{x}_i$ and its orthogonal projection $\mathbf{d} \in D$, using the Euclidean distance. Hence, PCA and *k-means* clustering can be seen as two extreme cases of principal objects where *k*-means is unstructured and PCA has a rigid linear structure.

1.9 PCA and Predictive Modelling

PCA is often used in the pre-process stage of data analysis before further predictive modelling is performed. As mentioned earlier, one of the applications of PCA is to remove variance that is considered distracting (noise) from the data. Removing distracting variance is a problem in dimension reduction which helps to reduce the number of parameters in the predictive model and to overcome problems such as multi co-linearity and overfitting.

Given a set of independent variables (predictors) $X_1, X_2, ..., X_m$ and a dependent variable Y, regression analysis seeks to estimate the relationship between the independent variables and the dependent variables. That is to model

$$Y = f(X_1, X_2, ..., X_m). \tag{1.41}$$

Regression analysis seeks the function f which relates the independent variables with the dependent ones.

Now let us consider the case of multiple linear regression models. It is assumed that the relationship f between the independent variables and dependent variables is linear and additive (the additive assumption can be removed by considering the interaction of the variables), and the model is given as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_m X_m + \varepsilon . \tag{1.42}$$

The scalars β_i are the parameters of the model (regression coefficient) , $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma I)$ is the error term which is assumed to be normally distributed and *I* is the identity matrix. The error term is used to capture all departure from the deterministic relationship between the dependent and independent variables which could arise from different sources such as measurement error, unmeasured variables that could contribute to the prediction etc. A multiple linear regression can either be used as a predictive model or as a tool for inference when the objective is to understand how the independent variables relate with the dependent variables. For ease of computation and without loss of generality we assumed that all the variables are mean centered.

The parameters are estimated from the sample dataset. If we view the independent variables $X_1, X_2, \ldots, X_m$ as columns of an $m \times n$ matrix X such that the *i*-th sample observation of $(X_1, X_2, \ldots, X_m)$ is the *i*-th row of X , Then we can use matrix notation to express (1.42). Let the dependent variable be given by the vector $\mathbf{y}$ with elements y_i such that $(\mathbf{x}_i, y_i)$ form a pair which represents a sample observation of independent variables and its corresponding dependent variable, thein in matrix notation

$$\mathbf{y} = X\beta + \boldsymbol{\varepsilon}. \tag{1.43}$$

where β is the m -dimensional vector with elements β_α, $\alpha = 1, \ldots, m$ and $\boldsymbol{\varepsilon}$ is the vector of error with elements ε_i which orresponds to y_i . It is assumed that the errors have the same variance σ and are uncorrelated. That is $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \Sigma)$. The estimate of the parameters using least square is given as

$$\hat{\beta} = \left(X^T X\right)^{-1} X^T \mathbf{y} \tag{1.44}$$

One major problem with multiple linear regressions is that of multi co-linearity. Typically there exist some forms of correlation between variables in a given dataset, however when there is a near constant linear function of two or more independent

variables, multi co-linearity is said to occur. A near constant linear function of a subset of the independent variables implies that one can be predicted from the others with a high degree of accuracy. This is undesirable in regression analysis as it causes the matrix $(X^T X)^{-1}$ in equation1.44 to be ill-conditioned. The variance of the parameter estimate becomes very large thereby affecting the stability of the model which affects the result of further predictions based on the model estimator. The variance of the estimate is given as:

$$\text{var}(\hat{\beta}) = \sigma(X^T X)^{-1}. \tag{1.45}$$

To see the effect of multi co-linearity on the estimate, we substitute the spectra-decomposition of the covariance matrix $X^T X$ into equation 1.45 to get

$$\text{var}(\hat{\beta}) = \sigma^2 \sum_{\alpha=1}^{m} \lambda_\alpha^{-1} \mathbf{e}_\alpha \mathbf{e}_\alpha^T \tag{1.46}$$

Where $\mathbf{e}_\alpha$ is the α-th eigenvector of $X^T X$ and λ_α is the corresponding eigenvalue (i.e. the variance of the α-th principal component). In the case of multi co-linearity in the data, it appears as a principal component (PC) with small variance λ_α with large inverse λ_α^{-1} which leads to large variance for the elements of $\hat{\beta}$.

However since PCA de-correlate the data set as given in *definition* 3; regression can be done on the principal components of the original data thereby ensuring that there is no multi co-linearity in the predictor set; this method is called Principal Component Regression (PCR). Let the principal component be $Z = XE$, where E is the matrix whose columns are the eigenvectors of the covariance matrix of X. Then the regression problem using the principal components as predictor can be expressed as:

$$\mathbf{y} = Z\eta + \boldsymbol{\varepsilon}, \tag{1.47}$$

where η is the vector of regression coefficients. If $Z = XE$, then $X = ZE^T$ and substituting this into equation 1.43 gives

$$\mathbf{y} = ZE^T \beta + \boldsymbol{\varepsilon}. \tag{1.48}$$

Comparing equation 1.48 with 1.43 we have $\eta = E^T \beta$ and $\beta = E\eta$.

It should be noted that if all the PCs are included in the predictor set, then the model obtained will be equivalent to that of least square on the original variables and hence the problem of large variance is still present. Therefore in order to reduce the variance, bias is introduced into the model by not including all the principal components in the model. The notion of bias–variance trade-off is popular in data modelling in which bias is introduced into a model in order to reduce the variance of the model with the aim of improving the stability and performance of the model. Since multi co-linearity usually appears as PC with small variance, one attempt to introduce bias is to exempts PCs with low variance. Therefore the biased estimate $\hat{\beta}$ of β will be given as

$$\hat{\beta} = \sum_{\alpha=1}^{k} \lambda_\alpha \mathbf{e}_\alpha \mathbf{e}_\alpha^T X^T \mathbf{y} \tag{1.49}$$

Equation 1.49 is obtained by using the spectra decomposition of $(X^T X)^{-1}$ and setting $\lambda_{k+1} = \lambda_{k+2} = \ldots = \lambda_m = 0$.

In addition to (PCR), other methods have been developed which also seek to reduce the variance due to multi co-linearity in a regression model by introducing some bias into the model. Such methods include the ridge regression and Lasso (see [53], [99]).

However, even though in PCA, principal components with high variance are selected based on the rationale that principal components with high variance are most informative, it should be noted that PCR does not take into account the dependent variable during the pre-processing step using PCA. The objective of PCA is to find the principal components that maximize the variance of the explanatory variables only; this may not necessarily have a satisfactory predictive performance. In fact [59] corrected the misconception that principal components with low variance are not important in prediction and show that they could be as important as principal components with large variance when it comes to regression analysis. Therefore, it is important to note that it is not the size of the variance of each of the principal components but the correlation between the principal components and the dependent variable that is of more importance in enhancing the predictive performance of the regression model. However, PCR can be made more efficient in predictive modelling by appropriate selection of the principal components used as predictors.

Other methods have been developed which take into account both the independent and the dependent variable during the dimension reduction step. An example of such methods is Partial Least Square method (PLS) which generalizes and combines features from PCA and multiple regression. Also, if the data is not well approximated by PCA (for example, data that are non-linear, disconnected or branched) then the performance of the predictive model could suffer if PCA is used for dimension reduction in the pre-processing step. To correct this, in [24], an analogue method to PCR but in which the data is projected to the principal curve approximating the data (using some tangent approximation to the principal curve or manifolds) was proposed. This method was called Projection Based Regression Tree. We remark also that PCA is not robust to the influence of outliers which could be present in data and PCR suffers from this drawback. One way this has been dealt with was to find a low dimension approximation of the data using some robust PCA. This approach was combined with robust linear discriminant analysis in [57] to improve the predictive performance of models.

In addition to the use of PCA in resolving the issue of multi co-linearity in datasets, using principal components as the predictor set make the computation of the parameters easier due to the orthogonality of the principal components. In a typical dataset, even when multi co-linearity is not present in the dataset, there are usually some level of correlation between the variables in the predictor set and the inclusion (or exclusion) of some variables in the model affects the parameter estimate of other variables. However, this is not the case when the principal components are used as the predictor due to the fact that the principal components are de-correlated.

In classification problem such as Linear Discriminant Analysis (LDA), PCA is employed as a tool for dimension reduction and also for data visualization. Data visualization is useful for revealing the structure of the data and the separation between the classes that exist in the data. The application of PCA as a tool for dimension reduction and data visualization also extend to clustering methods. However, it should be noted that in LDA and clustering problems the principal components with large variance do not necessary have to coincide with the direction in which the classes (groups) are separated. Therefore in dimension reduction, caution should be made in discarding principal components with low variance as it could provide information

about the class separation. But, if the inter class variance in the dataset is greater than the intra class variance then the direction of the first PC will provide information about the class separation since the direction of the first PC will aligned with the direction of maximum variance of data projection.

Finally, we remark that various distortions of the data space are possible during dimension reduction. For data which are labelled into classes, this could affect class structure (both inter-class and intra-class structure) of the data. This distortion could affect the performance of decision model for separating the classes. See [41] on how to analyse this distortion and for some examples.

1.10 Big Data

Virtually every field of human endeavour is driven by technology. Innovation in technology over the last few decades has led to an explosion in the volume of data generated and collected leading to what is now called big data. According to the reports [13], [93] [34] [87] [33] [31] [32] [78] [11], one could establish that there is an exponential growth in the volume of data and this has led to new challenges in managing and analysing these data.

Big data can be defined as datasets whose size is beyond the ability of typical database software tools to capture, store, manage and analyse [80]. A lot of organizations are beginning to pay attention to big data in order to leverage their businesses and increase production, competitiveness, reducing waste and to support human decision making. This is because it is expected that the more data that is available, the more information one can elicit from the data and also the more accurate the result of analysis of such data. Thus big data is now a major factor of the economy.

However, in addition to the volume of data generated, big data also comes with varying degrees of complexities; in fact, this century is called the century of complexity [48]. One of the complexities that have to be dealt with is the dimensionality of big data. We know that many statistical learning methods behave irrationally in high dimensional space which is termed the curse of dimensionality [10] as previously discussed. Therefore, there is a need to reduce the dimension of the data in order to understand the data and to learn from this data. Reducing the dimension of data is also needed for data visualization.

Finally, one other reason for dimension reduction is the need to manage the data. As mentioned earlier, the growth of data is exponential however the development in the infrastructures necessary to manage these data is not commensurate. An example of the application of dimension reduction in big data management is in data storage, the data can be approximated by an appropriate lower dimension object and stored.

1.11 Problem Statement and Structure of the Thesis

The problem of approximating a system of multidimensional points by a linear manifold of lower dimension led to the development of PCA [85]. This has been extended to approximating data using nonlinear manifolds, finite set of points or other principal objects. The success of data approximation is crucial to knowledge discovery and modelling as data approximation allows for data visualization and many models perform poorly on very high dimensional data. It is essential that while approximating data by objects of lower dimension or complexity, the dynamics and structure of the data is preserved.

However approximating data usually leads to some form of distortion of the data. For example, data points which are far from each other can be closely projected when approximated while some datapoints that are close to each other in the original dataspace may be projected far from each other when approximated. In the case of PCA, it favours directions with large distances (which correspond to direction with large variance) which can lead to distortion of other structures in the data which exist at smaller distances and sometimes completely obfuscating such structures. Therefore it is not just sufficient to approximate data by objects of lower dimension but it is also important to evaluate the distortion of the data as such distortion could have an impact on further analysis. In particular for PCA, there is the need to reveal the various structures that exist at various distances, to analyse the structures and to evaluate the distortions to these structures by the approximation.

The distortion during data approximation can be undesirable as data visualization using such approximation could be misleading. For most manifold modelling methods, the quality of approximation depends crucially on the initial conditions as most manifold modelling methods adopt the expectation maximization approach (or its variant). The first part of this thesis focuses on the effect of

initialization on the quality of approximation of manifold modelling methods. Using Self Organising Maps (SOM) as a case study of manifold modelling techniques we will analyse the effect of two popular initial approximation approaches on the final approximation of data using various examples. The two initialization methods compared are: initialization from the space of principal component of the data and random initialization. To understand the dynamics of manifold modelling methods, the dataset will be further classified as linear, quasilinear and nonlinear.

Also in this thesis, we will address the problem of revealing intrinsic structure(s) which PCA may not reveal. Since PCA finds the global structure of data, we will attempt to study the PCA of data at various localizations in order to reveal intrinsic structures. We will look at localization in two different directions. The first direction is an extention of definition 4 in which we will find the subspaces that approximate the data for various restricted distributions of pairwise distances. The PCA structures resulting from this localization will be embedded into appropriate space and analysed to reveal intrinsic structures. For the various stuctures that will be revealed we will further evaluate the quality of approximation of such structures in term of preservation of local and global structures, distortion of the dataset structures and preservation of class structure for labelled data.

The second direction to localization will be to perform PCA locally in the dataspace. PCA is a non parametric analysis; given a dataset, the resulting principal components are unique and reflect the global structure of the dataset. However for some complex data, there may be different structure(s) locally in the dataspace. We analysed the local PCA structures in order to deduce the dynamics and structure of the data in the various region of the dataspace.

Finally, we will combine these two localization approaches to find robust approximation of data.

Structure of the Thesis

There are six chapters in this thesis. In chapter 1 we have already given a brief history and development of PCA, the various derivation of PCA, some assumptions and limitations of PCA, and some properties of PCA. We have looked at some applications of PCA and numerical approach to finding principal components. We introduce the use of PCA in the pre-processing step of predictive modelling and also the recent challenges with big data.

The remaining part of this thesis is organised as follows:

Chapter 2: In this chapter we will review two directions in which PCA have been generalized and extended especially to cope with complex data. Using definitions 1 and 4 which derived PCA using distance function, PCA was generalized by assigning weights either to the datapoints or to the pairwise distances of datapoints. We review how this can be put into a general framework using generalized SVD. The second direction in which PCA has been generalized is by replacing the linear function in the approximation with a nonlinear function. An example of this is principal curve and manifold which analogous to PCA (which seeks hyperplane that passes through the middle of system of points) find smooth curves and manifolds which pass through the middle of the system of data points, using the notion of self consistency to define the middle of the data cloud. Next we review Self Organising Maps (SOM), an artificial neural network which approximate a system of points by a finite set of points (neurons) of smaller size (usually arranged in rectangular grid of 1 or 2 dimension size) to form a topological mapping of the data. We will also review the kernel PCA which first maps data nonlinearly into a reproducing kernel Hilbert space before finding the principal component. We will also review another development in nonlinear approximation which provides a general framework for constructing principal objects of various dimensions and topologies using the metaphor of elastic membrane and plates. These are called elastic nets and elastic maps respectively. This was extended using a topological grammar to develop principal graphs. This method produces approximators of various geometric, structural and construction complexities which can cope with branching data. We will also review the development so far in approximating data locally in the dataspace using PCA and the application of PCA to tensor objects. For tensor objects, the approach of vectorising the tensor objects before finding the

principal components does not take into consideration the natural structure of the tensor objects approximation which may destroy the important structure of the tensor objects hence the need for perfoming PCA directly on the tensor space. We will also discuss the advantages and disadvantages of projection methods and manifold modelling methods. Finally, we look at a standard problem in manifold learning which is the problem of initial approximation as everything depends crucially on the initial approximation. In this chapter we investigate the effect of initialization on manifold modelling methods using SOM as a case study. We compare the performance of two initialization approaches (Random Initialization (RI) and principal component initialization (PCI)) which are popularly used for manifold learning methods and SOM in particular. To further understand the performances of these initialization methods, the data were classified as quasilinear and nonlinear based on the manifold of the data. The performance of the initialization methods were compared for the various classifications of the data using fraction of variance unexplained as a criterion. The results showed that the widely accepted presumption about the advantage of PCI SOM initialization is not universal and in the case of SOM this presumption is definitely wrong for essentially nonlinear data.

Chapter 3: The goal of this chapter is to study the structure(s) of data at various distances thereby revealing some hidden structure(s) in the data which conventional approach may not reveal. Using weighted PCA, we find the sequences of subspaces that best approximate the data for various distributions of pairwise distances (scales) which we call Multiscale Principal Component Analysis (MPCA). The resulting principal components are scale dependent and we will further analyse these principal components in order to establish the structures in the data. We show that representing principal components using the orthogonal loading vectors (orthonormal k frame) does not preserve some important properties of the principal components. It turns out that principal components are lines rather than vectors and are thus points in the projective space. More generally, when we consider multiple principal components, these are points in the Grassmannian space. To study points in the Grassmannian space we embed the points into a suitable vector space, in our case the space of orthogonal projection matrices. We study some properties of this representation to ensure consistency with the properties of principal components. To reveal the structures in the data, we analyse the PCA structures at various scales. We defined the distances between

two points on the scales as the distance between its PCA structures and by examples we show that for data with clear multiscale structure, MPCA reveals the structures. We introduce the ratio of distortion as a criterion for measuring the distortion of distance structure and we discuss the effect of scaling on MPCA as MPCA is not scale invariant.

Chapter 4: In this chapter we study the PCA structure of data at various localization of the dataspace. For complex data which can be described as nonlinear, 'branched' 'disconnected' or generally as complex data, data is best approximated locally. Local PCA can be seen as tangent approximation to nonlinear methods. Using a kernel function, we formulate the problem of approximating PCA locally in space as a weighted PCA. We also study the PCA structure at various localizations to understand the structure of the data using the representation introduced in chapter 3. The quality of the analysis depends on the quality of the partition of the data. Partitioning of data is a classical problem in clustering therefore we discuss some options for selecting the partition and we also discuss the recursive local PCA algorithm. Finally we discussed how to combine MPCA and PCA localization in space to provide robust analysis and to reveal geometric structures.

Chapter 5: In this chapter we analyse both artificial and real data to demonstrate how the analysis of the MPCA structures of a dataset (as discussed in chapter 3) and local PCA structures in space (as discussed in chapter 4) can be used to reveal intrinsic structures in the dataset. We also discuss the issue of overfitting. For each cluster (resulting from the clustering of scales), we select representative scales to describe the cluster. We further analyse the representative scales to evaluate how each cluster preserve local structure at various scale compared to PCA using intersection of *k*-nearest neighbour of the data and the data approximation for selected points. We examine how each cluster preserves global structure in the dataset compared to PCA using correlation analysis of the distance structure in the original dataspace and the subspace of approximation. However due to dependence in the sample of pairwise distances, the correlation was performed on selected points which are chosen to be independent. This is called NatPCA and introduced by Gorban et al in [41]. For each cluster, we also compare the distortion of the data compared to PCA using ratio of distortion and for labelled data we compare how each cluster preserves the class structure of the data compared to PCA during approximation. Finally we perform local

PCA on the Iris dataset to reveal how the PCA structure changes as the radius of neighbourhood changes. We also cluster the local PCA structures for the Energy efficiency dataset to identify regions in the dataspace with similar PCA structures.

Chapter 6: This is the final chapter and it discusses some results obtained from the work in this thesis and also provides some critical analysis of the methods developed. It also highlights areas of further research.

The results have been presented at the following conferences and seminars:

- 2nd International Conference on Mathematical Modeling in Physical Sciences 2013 (IC-MSQUARE 2013). Held in Prague.
  - "Mulitscale Principal Component Analysis"
- European Conference on Data Analysis (ECDA) 2013. Held in Luxembourg.
  - "Is PCA good for SOM Initialization?"
- Workshop in Geometrical Structures in Statistics 2013. Department Mathematics, University of Durham.
  - "Multiscale Principal Component Analysis"
- Midlands Postgraduate Probability workshop 2013.
  - "Revealing Geometric Structures in Data using Multiscale Principal Component Analysis"
- The Internal Applied Seminar in Department of Mathematics in University of Leicester 2012.
  - "Initialization of Self-Organizing Maps: Principal Components versus Random Initialization. A case study"

And partially published in

- Journal of Physics Conference Series 07/2013; 490(1).
  DOI:10.1088/1742-6596/490/1/012081
  "Multiscale Principal Component Analysis".

- Information Sciences (2015)
  DOI: 10.1016/j.ins.2015.10.013
  "SOM:stochastic initialization versus principal components".

Chapter 2

Generalization and Extension of PCA

2.1 Introduction

Several approaches have been taken to generalize PCA and to adapt it in different contexts. In this chapter, we briefly review some extensions and generalizations of PCA. We will concentrate on two popular approaches in which there has been considerable development. The first approach is based on methods that generalize PCA by introducing weights on datapoints, variables or distances between datapoints. The second approach is based on methods that generalize the functional form of the PCA using non-linear curves and manifolds to approximate a system of points. Finally, in this chapter we will consider a standard problem common to most manifold learning methods, which is the problem of initial approximation of a method. The quality of approximation depends crucially on the initial approximation. See for example [7, 94].

2.2 Weighted PCA

Firstly, we note that PCA as introduced in definitions 1 and 4 are defined with the use of distance function. This allows for generalization by applying weights to the respective distance function. This can be used to incorporate external knowledge into the analysis of principal components. Some examples of such external knowledge could be of importance of certain data points and similarities or dissimilarities between the data points.

Let us consider the PCA as given in definition 1. Using the Euclidean distance, this problem was stated as

$$D_X = \frac{1}{n}\sum_{i=1}^{n} \| \mathbf{x}_i - P_L \mathbf{x}_i \|_2^2 \to \min .$$

One way to generalize this is to apply weights to the distances between a data point $\mathbf{x}_i$ and its projection $P_L \mathbf{x}_i$. Hence we solve the problem given as

$$D_X^w = \frac{1}{n}\sum_{i=1}^{n} w_i \parallel \mathbf{x}_i - P_L\mathbf{x}_i \parallel_2^2 \rightarrow \min, \tag{2.1}$$

where $w_i \geq 0$ is the weight applied to the distance between the data point $\mathbf{x}_i$ and its projection $P_L\mathbf{x}_i$. We note that the case where $w_i = 1 \; \forall i$ is the PCA as seen in definition 1. Choosing $w_i = 0$ for some *i* correspond to exempting such data points from the analysis. Weighted PCA can be used to exempt certain influential datapoints such as outliers and thus improve the robustness of PCA.

Another direction is to weight the distances between data projections. We recall from definition 4 that PCA seeks the set of orthognormal vectors $\mathbf{v}_\alpha$ which maximizes

$$D_X = \sum_{i<j}\left[\sum_{\alpha=1}^{k}\left\langle \mathbf{v}_\alpha, (\mathbf{x}_i - \mathbf{x}_j)\right\rangle^2\right].$$

Therefore, PCA can be generalized as follows:

$$D_X^w = \sum_{i<j}\left[\sum_{\alpha=1}^{k} w_{ij}\left\langle \mathbf{v}_\alpha, (\mathbf{x}_i - \mathbf{x}_j)\right\rangle^2\right]. \tag{2.2}$$

That is, we seek to maximize the weighted pairwise distances of the data projection. The particular case where $w_{ij} = 1 \; \forall i, j$, leads to PCA as given in definition 4.

Weights can be used to incorporate external knowledge of similarity or dissimilarity into the data analysis. Koren and Carmel (2004) gave some applications of weighted PCA as given by equation (2.2) and one of these applications is to enhance the robustness of PCA to outliers by underweighting distant datapoints. They proposed choosing $w_{ij} = \frac{1}{dist(\mathbf{x}_i, \mathbf{x}_j)}$ or $w_{ij} = \frac{1}{dist(\mathbf{x}_i, \mathbf{x}_j)^2}$ or some decaying exponential function of the pairwise distances of the datapoints. Another application of this which applies to labelled data is called supervised PCA in which the weights are choosen to emphasize the discrimination between clusters [71].

$$w_{ij}^{\;labeled} = \begin{cases} t.w_{ij} & i \; and \; j \; have \; the \; same \; label \\ w_{ij} & i \; and \; j \; have \; different \; label \end{cases}$$

Where $0 \leq t \leq 1$.

Greenacre (1984) in his paper [49] put this into a general framework in which generalized PCA was defined through generalized SVD (note that weighted PCA as defined by (2.2) does not fit into this). As mentioned in section 1.7, Principal component analysis is usually done through SVD. The SVD of a matrix $X = UDV^T$ where $U^T U = I_k$ and $V^T V = I_k$ and I_k is the identity matrix of rank k. Given positive definite matrix Ω and Φ, the Generalized SVD of a matrix X is given as

$$X = AMB^T \tag{2.3}$$

Where A satisfies the condition $A^T \Omega A = I_k$, and $B^T \Phi B = I_k$. We observe that when Ω and Φ are I (the identity matrix) then we have SVD.

Analogously, Greencare defined generalized PCA as having the loading vectors given by the columns of *B*, the principal components as the projection of the data onto this loading vectors. Similar to Eckart-Young Theorem for SVD, if Ω and Φ are chosen to be diagonal matrix with diagonal elements w_i, $i = 1,...,n$ and ϕ_j, $j = 1,..m$ respectively, then the solution to the problem of approximating X with a rank *k* matrix in the sense of minimizing

$$D_X = \sum_i \sum_j w_i \phi_j (x_{ij} - \hat{x}_{ij})^2 \rightarrow min \tag{2.4}$$

is given by choosing the approximating matrix $\tilde{X}$ to be the generalized SVD under the constraint that the lowest $m - k - 1$ singular values are set to zero. That is, if generalized SVD of $X = AMB^T$, then

$$\tilde{X} = AM^* B^T, \tag{2.5}$$

where M^* is the diagonal matrix with diagonal elements d_i, such that

$$d_1 \geq d_2 \geq ... \geq d_k > d_{k+1} = d_{k+2} = ... = d_m = 0.$$

When Ω and Φ are diagonal matrices, then we see that the elements of Ω introduce weights to the datapoints and the elements of Φ introduce weights to the variables. If we chose $\phi_j = 1, \forall j$ then (2.4) becomes

$$D_X = \sum_i \sum_j w_i (x_{ij} - \hat{x}_{ij})^2 \rightarrow min. \quad (2.6)$$

And if we define the generalized PCA from the generalized SVD, then the optimization probem 2.6 leads to 2.1. This is the case where the observations are weighted before performing PCA. One example of the application of weighted PCA is in the analysis of microarray data [18], where the weights are introduced to the datapoints (observations).

Also if we chose $w_i = 1, \forall i$, then (2.4) becomes

$$D_X = \sum_i \sum_j \phi_j (x_{ij} - \hat{x}_{ij})^2 \rightarrow min, \quad (2.7)$$

which leads to a weighted PCA in which weights are applied to the variables. PCA defined using correlation matrix (rather than covariance matrix) can be seen as a special case of Generalized PCA of the form 2.7, where $\phi_j = 1/s_{jj}$ and s_{jj} is the sample variance of the jth variable [58]. The use of a correlation matrix for PCA usually arises in the event that the variables have different measurement units. It could become appropriate to normalize the variables to ensure that they are dimensionless; performing PCA using these normalized variables lead to eigen-decomposition of the correlation matrix.

One example of application of weighted PCA is to wavelength kinetic experiments [17], where weights are applied to both the observation and the variables.

2.3 Nonlinear Generalization of PCA

The need for a non-linear generalization of PCA arises from the fact that PCA is a linear multivariate analysis technique and may not be adequate for approximating data with non-linear relationships. There has been considerable development of non-linear techniques for data approximation over the last two decades. We briefly review in this section a few major non-linear approaches which generalize PCA and also some extensions of these methods.

According to definition 1, PCA seeks the linear function that minimizes the MSD as given below

$$\sqrt{\frac{1}{n}\sum_{i=1}^{N}\|\mathbf{x}_i - f(\mathbf{x}_i)\|_2^2} \rightarrow min. \tag{2.8}$$

When $f(\mathbf{x}_i)$ is chosen to be the projection of $\mathbf{x}_i$ onto a subspace of dim $k < m$ of $\mathbb{R}^m$, we have PCA and $f(\mathbf{x})$ can be represented as $P^T\mathbf{x}$ where P is the projector matrix which has as its columns the eigenvectors of the sample covariance matrix of the dataset. One attempt to generalize PCA non-linearly is to replace $f(\mathbf{x})$ with a non-linear function of $\mathbf{x}$ that minimizes the MSD or that optimizes some other objective functions. Non-linear extension of PCA has also allowed for the extension of PCA to nominal and ordinal data, for example, the Gifi's method [36].

Gnanadesikan (1977) proposed introducing non linearity into PCA by using an extended vector $\mathbf{x}_+$. Where $\mathbf{x}_+$ extends the vector $\mathbf{x}$ by including functions of the elements of $\mathbf{x}$. For example given $\mathbf{x}^T = (x_1, x_2)$, we can choose $\mathbf{x}_+^T = (x_1, x_2, x_2^2, x_2^2, x_1x_2)$ and then PCA is performed on $\mathbf{x}_+$ [39]. He focused on using a quadratic form of the elements of $\mathbf{x}$. This can be seen as kernel PCA (see section2.3.3) with a quadratic kernel. The kernel method was made popular in the 1990s when it was used to extend the support vector machine algorithm to have non linear decision boundary. This was based on the previous works of Vladimir N. Vapnik and Alexey Ya. Chervonenkis (1963). Other non-linear functions that have been advocated include a logarithmic transformation of the elements of $\mathbf{x}_i$ [92], powers of the elements of $\mathbf{x}_i$ and Splines [74]. Principal component analysis using correlation matrix can also be viewed as a particular case of non-linear PCA where each variable is transformed by normalizing to unit variance of the variables. The choice of non-linear function to use is problem specific and consideration should be given to the suitability of such function to the analysis.

2.3.1 Principal Curves and Manifolds

In PCA, we seek to approximate a system of points using the best lines and planes which passes through the data cloud. Hastie and Stuelze (1989) proposed approximating a system of points using smooth curves and manifolds which pass through the data cloud, this was called Principal Curves and Manifolds [51]. We now discuss the mathematical background needed to find the principal curves and manifolds.

The derivation of PCA by Pearson (1901) was based on a geometric argument; however with the development of probabilistic interpretation of statistic, we have another view to the dataset. Let a multidimensional probability distribution $F(\mathbf{x})$ define the probability of appearance of a sample in the point $\mathbf{x} \in \mathbb{R}^m$, a dataset X can be interpreted as one particular independent and identitcally distributed sample from $F(\mathbf{x})$, this interpretation allows for the definition of many statistical notions. One of such statistical notions of fundamental importance is the notion of self-consistency [96] [23].

Definition of Self-consistency: Given a probability distribution $F(\mathbf{x})$ and a set of vectors Y, we say that Y is self-consistent with respect to $F(\mathbf{x})$ if $\mathbf{y} = \mathrm{E}(\mathbf{x} / P_Y(\mathbf{x}) = \mathbf{y})$ for every $\mathbf{y} \in Y$. Where $P_Y(\mathbf{x})$ is the projection of $\mathbf{x}$ to the set of vectors *Y*. This means Y passes through the middle of the data cloud since every $\mathbf{y} \in Y$ is the mean of all vectors projected to it.

Tarpey et al (1995) showed that for elliptical distribution, the self-consistent points exist only in the principal component subspaces [98]. In [97] the self-consistency of principal component subspaces of a large class of symmetric multivariate distributions was examined.

Hastie and Stuelze (1989) defined a principal curve for a distribution $F(\mathbf{x})$ as a non-intersecting, self-consistent smooth curve Y.

Definition*:* Let G be the class of differentiable 1-dimensional curves in $\mathbb{R}^m$, parameterized by $\lambda \in \mathbb{R}^1$ and without self- intersection (i.e. if $\lambda_1 \neq \lambda_2 \Rightarrow G(\lambda_1) \neq G(\lambda_2)$. The principal curve of the probability distribution $F(\mathbf{x})$ is such $Y(\lambda) \in G$ that is self-consistent. Where λ_i is the projection index of a point $\mathbf{x}_i$ to the curve Y which satisfies

$$\lambda_i = \sup_{\lambda}\left\{\lambda : \| \mathbf{x}_i - Y(\lambda) \| = \inf_{\mu} \| \mathbf{x}_i - Y(\mu) \|\right\}. \quad (2.9)$$

Equation 2.9 implies that λ_i is the value of λ for which $Y(\lambda)$ is closest to $\mathbf{x}$, and if there are several such values, the largest one is chosen. This definition can be extended to a 2-dimensional surface in R^m, see [40].

Usually the distribution $F(\mathbf{x})$ is not known and for a finite dataset in particular, given a point $\mathbf{y} \in Y$, usually few points (typically one) or even no point is projected to it. Therefore, it is not feasible to calculate the conditional mean given in the definition. Hence the need for the coarse grained self-consistency notion, which is estimating the conditional expectation using the local averaging of observations projecting into a neighbourhood of the estimate of the curve. The size of this neighbourhood controls the complexity of the resulting approximation Y

Hastie and Stuelze showed connection between principal curves and the first principal component. One of such proposition is that if a straight line is self-consistent for a probability distribution, then it is a principal component. The proof of this and other interesting connections between PCA and principal curves can be found in [51]. While PCA seeks the line passing through the middle of a dataset, principal curve generalizes this by seeking for the curve passing through the middle of the dataset where middle is defined using the notion of self-consistency. For elliptical distributions and multivariate normal distributions, the first principal component defines the principal curve (though the principal curve is not unique). Hence we can view non-linear principal curves as a generalization of the first principal component for some other probability distribution.

Algorithm for finding a principal curve for a finite dataset.

This algorithm follows the expectation/maximization algorithm as introduced by [21].

1) Initialization Step: an initial smooth curve $Y(\lambda)$ is chosen. This is usually set as $Y(\lambda) = \bar{\mathbf{x}} + \lambda \mathbf{u}_1$, where $\bar{\mathbf{x}}$ is the mean point and $\mathbf{u}_1$ is the loading vector of the first principal component.

2) Projection Step: project all data points $\mathbf{x}_i$ onto $Y(\lambda)$: that is for each $\mathbf{x}_i$ find λ_i that satisfies equation 2.9. Because $Y(\lambda)$ is determined in a finite number of points, there is a need to interpolate.
3) Expectation Step: Calculate new $Y'(\lambda) = E(\mathbf{x} / P_Y(\mathbf{x}) = Y(\lambda))$. As mentioned earlier, for finite dataset, typically zero or one observation is projected onto $Y(\lambda_i)$. Therefore coarse grained self-consistency is used, that is the local average of points $\mathbf{x}_i$ and some other points that have close to λ_i projection onto Y .
4) Update Step: Reassign $Y(\lambda) \leftarrow Y'(\lambda)$
5) Iteration: Repeat step 2-4 until Y does not change or the change is below some set threshold.

To carry out the projection step given above, we define d_{ik} as the distance between $\mathbf{x}_i$ and its closest point on the line segment joining each pair $\left(Y^j(\lambda_k^j), Y^j(\lambda_{k+1}^j)\right)$ where $Y^j()$ and λ^j, represent the curve *Y* and projection index λ produced at the j*-th* iteration. Hence corresponding to each d_{ik} is a value $\lambda_{ik} \in \left[\lambda_k^j, \lambda_{k+1}^j\right]$. Therefore we set $\lambda_i = \lambda_{ik*}$ if $d_{ik*} = \min_{\forall k} d_{ik}$. Corresponding to each λ_i is an interpolated $\mathbf{Y}_i$; using this value to represent the curve, we replace λ_i by the arc length from Y_1^j to Y_i^j. See [51] for further reading on arc length.

Hastie and Stuelze also found the principal curve to be biased for the functional data model. This means that if a sample is generated from the model $\mathbf{x} = Y(\lambda) + \varepsilon$, where $Y \in G$ and $E(\varepsilon) = 0$, then the principal curve is not necessarily $Y(\lambda)$. However, there is some evidence that the bias is small. Ways to reduce this bias was discussed in [100].

Also principal curve as define above leads to the following questions:

1) For what probability distributions does principal curve exist?
2) How many principal curves exist for a give probability distribution and their properties?
3) How does the algorithm proposed by Hastie and Stuelze perform for distribution for which principal curves do not exist?

The question about existence makes it difficult to analyse the consistency and convergence rates of any estimation scheme for the principal curve [51] [63] [72]. To resolve this problem, Kegl et al (2000) proposed a new definition of principal curve by incorporating a length constraint, combining vector quantization with principal curves. They showed that even though uniqueness was not guaranteed, however principal curve of length $L > 0$ always exist provided that X has second finite moments (i.e. $\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T < \infty$) [63].

Definition: A principal curve $Y^*_L(\lambda)$ of length L is such a curve that the MSD from data X to the curve $Y_L(\lambda)$ is minimal over all curves of length less than or equal to L.

$$Y^*_L = \sum_{i=1}^{n} dist(\mathbf{x}_i, P(\mathbf{x}_i, Y_L)) \rightarrow \min_{Y_L}. \tag{2.10}$$

Due to the computational complexity of finding the principal curve as defined, Kegl et al [63] proposed the polygonal line algorithm which is a sub-optimal method but more computationally tractable. This method uses the first principal component as the initialization for the algorithm. It should be noted that for complex data with uneven and (or) sparse distributions, initialization plays a role in guaranteeing that the algorithm converges to the principal curve. In [72], other problems that can arise from this algorithm are highlighted.

In [51], Principal curves was applied in aligning the magnet of the Stanford linear collider and also to study two different assays for gold contents in several samples of computer chip waste, revealing a structure which PCA did not reveal.

2.3.2 Self-Organising Maps (SOM)

Another approach to approximation of data is SOM which can be considered as a non-linear PCA [107]. Inspired by biological neural networks, Kohonen (1982) [67] developed the SOM which is a type of artificial neural network that uses an unsupervised learning algorithm with the additional property that it preserves the topological mapping from input space to output space making it a great tool for visualization of high dimensional data in a lower dimension. Originally developed for

visualization of distribution of metric vectors [66], SOM found early applications in speech recognition [66].

The SOM uses a set of neurons (nodes) with cardinality M , usually in a 1, 2 or 3-dimensional rectangular or hexagonal planar grid with regular spacing to form a discrete topological mapping of a dataset. Training this network utilizes competitive learning. For every input pattern, the neurons compete for ownership and the winner is adjusted to learn this input pattern. To ensure that the local spatial properties of the input data are preserved, in addition to the winning neuron, neurons within a given neighbourhood of the winning neuron are also adjusted as necessary.

The SOM algorithm

As proposed by Kohonen, the SOM algorithm can be summarised as follows:

i) Initialization: Initial weights $\mathbf{w}_j(0)$ are assigned to all the neurons, where $\mathbf{w}_j$ is a weight vector with same dimension as the dataset.

ii) Competition: all neurons compete for the ownership of the input pattern. Using the Euclidean distance function, the neuron with the minimum-distance wins.

$j^* = \arg\min_k \| \mathbf{x}(k) - \mathbf{w}_j \| \; j = 1,2,...,M$, where $\mathbf{x}(k)$ is the input pattern at time k .

iii) Cooperation: the winning neuron also excites its neighbouring neurons (topologically close neurons). An example of neighbourhood function often used is the Gaussian neighbourhood function,

$\eta_{*i}(k) = \alpha(k) \exp\left(- \frac{\| r_* - r_i \|^2}{2\sigma^2(k)} \right)$, where $\alpha(k)$ is a monotonically decreasing learning factor at time k and r_i is the position of neuron i .

iv) Learning Process (Adaptation): The winning neuron and its neighbours are adjusted with the rule given below:

$$\mathbf{w}_j(k+1) = \mathbf{w}_j(k) + \alpha(k)\eta_{*i}(k)[\mathbf{x}(k) - \mathbf{w}_j(k)].$$

Hence, the weight of the winning neuron and its neighbours are adjusted towards the input pattern however the neighbours have their weights adjusted with a value less than the winning neuron. This action helps to preserve the topology of the data.

As $k \to \infty,\ \eta_{*i}(k) \to 0.$

The quality of learning of the Self-Organizing Map (SOM) is influenced by initial weights of the map. Initial weights are often selected randomly or selected as vectors from the space of the principal components of the dataset. In section 2.6, the result of an empirical study of the performance of these initialization approaches will be presented. Other initial conditions that influence the quality of learning of SOM include: the neighbourhood function, the learning rate, sequence of training vectors and number of iterations [7] [15] [94].

Another issue with SOM is determining the number of neurons suitable for approximating a dataset. Growing Self Organising Map (GSOM) was developed to address this problem. GSOM starts with a minimal number of neurons and grows new neurons on the boundary until specified stopping criteria are satisfied.

SOM was not formulated as an optimization problem; it does not have an exact cost function. An attempt to overcome most of the significant limitations of the SOM led to the development of a principled alternative to SOM called Generative topographic Mapping (GTM) [12]. For convergence property for SOM see [113].

2.3.3 Kernel PCA (KPCA)

Schölkopf et al. (1998) proposed a non-linear extension of PCA in which the data is first mapped nonlinearly to a feature space of dimension higher than the dimension of the data (a Reproducing Kernel Hilbert Space) and PCA is performed in this feature space. This approach was called Kernel PCA [92].

Let $\Phi(\mathbf{x}_i)$ be the mapping of $\mathbf{x}_i$ to a feature space - F with considerable higher dimension than m.

$$\Phi : \mathbf{x}_i \to F \ , \ \dim(F) >> m$$

The PCA is perfomed in F by finding eigen-decomposition of the sample covariance matrix $\overline{C}$ in F

$$\overline{C} = \frac{1}{n-1} \sum_{i=1}^{n} \Phi(\mathbf{x}_i)\Phi(\mathbf{x}_i)^T \,. \tag{2.11}$$

The kernel method can be applied to algorithms that can be formulated exclusively in terms of dot product. The trick is to represent the dot product by a kernel function which computes the dot product in some possibly high dimensional feature space (derived space). Linear methods applied in such derived space turn out to be nonlinear in the original data space because of the nonlinear mapping from the data space to the derived space. An example of such is the support vector machine method. Since the computation of PCA in the feature space can be formulated exclusively in terms of a dot product, we can apply a kernel function in the feature space and we do not require $\Phi(x_i)$ in explicit form. Therefore we require the eigenvalue λ and eigenvector $\mathbf{e}$ such that

$$\overline{C}\mathbf{e} = \lambda\mathbf{e} \,. \tag{2.12}$$

Since all solutions $\mathbf{e}$ lie in the span of $\{\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \ldots, \Phi(\mathbf{x}_n)\}$, $\exists$ coefficients $\alpha_1, \ldots, \alpha_n$ such that

$$\mathbf{e} = \sum_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i) \,. \tag{2.13}$$

Let us consider the equivalent system

$$\left\langle \Phi(\mathbf{x}_k) \cdot \overline{C}\mathbf{e} \right\rangle = \lambda \left\langle \Phi(\mathbf{x}_k) \cdot \mathbf{e} \right\rangle, \quad k = 1,2,\ldots,n \,. \tag{2.14}$$

Let the kernel function be defined as

$$K_{ij} = \left\langle \Phi(x_i), \Phi(x_j) \right\rangle, \tag{2.15}$$

substituting equations 2.13 and 2.15 into 2.14 give

$$K^2\alpha = n\lambda K\alpha \,. \tag{2.16}$$

To find the solution of (2.16), we solve

$$K\alpha = n\lambda\alpha \tag{2.17}$$

Since (2.16), (2.17) have the same solution.

We normalize the solutions α^k belonging to non-zero eigenvalue by normalizing the corresponding vectors in *F*, $\left\langle \mathbf{e}^k, \mathbf{e}^k \right\rangle = 1$

$$1 = \sum_{i,j=1}^{n} \alpha_i{}^k \alpha_j{}^k \left\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \right\rangle = \alpha_i{}^k K \alpha_j{}^k = \lambda^k \left\langle \alpha_i{}^k \alpha_j{}^k \right\rangle .$$

The principal component in the feature space of a given test point $\mathbf{x}$ is the projection of $\Phi(\mathbf{x})$ to $\mathbf{e}^k$

$$\left\langle \mathbf{e}^k, \Phi(\mathbf{x}_i) \right\rangle = \sum_{i=1}^{N} \alpha_i{}^k \left\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \right\rangle, \tag{2.18}$$

where the dot product in 2.18 can be computed using the kernel function 2.15.

The computation complexity of KPCA does not grow with the dimensionality of the feature space that we are implicitly working in. KPCA also provides a better understanding of what kind of non-linear features are being extracted since the feature space is fixed a priori by choosing the kernel function. While PCA depends entirely on the first and second moment of the data, KPCA does not [16]. KPCA finds application in face recognition, image de-noising and fault detection [72].

2.3.4 Elastic Nets and Maps

Principal Manifolds are lines or surfaces (2-d manifolds) passing through "the middle" of the data distribution. When the principal manifold is linear, we have a principal component analysis, and when it is a curve with "middle" defined based on the notion of self-consistency of the curve with respect to the data distribution we have principal curves.

In a series of papers [42][43][44][45], a general framework for constructing principal objects of various dimension and topology using the metaphor of elastic membrane and plates to construct 1,2 and 3 dimensional principal manifold of various topology was developed. Elastic nets are systems of springs embedded in data space. This system forms a regular grid so that it can serve as approximation of some low dimension manifold. The purpose of elastic nets is to introduce point approximation to manifold. In

this section, we review the method of elastic nets and elastic maps for approximating principal manifolds.

Let G be a simple undirected graph with set of vertices V and set of edges E.

Definition: $k-star$ in a graph G is a subgraph with $k+1$ vertices $v_{0,1,\ldots,k} \in V$ and k edges $\{(v_0, v_i)\ i=1,\ldots,k\} \in E$. A 2-star is called a rib.

Definition: Suppose that for each $k \geq 2$, a family S_k of $k-star$ in G has been selected. An elastic graph is defined as a graph with selected families of $k-star$ S_k and for which for all $E^i \in E$ and $S_k^i \in S_k$, the corresponding elasticity moduli $\lambda_i > 0$ and $\mu_{kj} > 0$ are defined.

Definition: Let $E^i(0), E^i(1)$ denote two vertices of the graph edge E^i and $S_k^j(0),\ldots,S_k^j(k)$ denote vertices of a $k-star$ (where $S_k^j(0)$ is the central vertex to which all other vertices are connected). Let us consider a map $\phi: V \to \mathbb{R}^m$ which describes an embedding of the graph into a multidimensional space. The elastic energy of the graph embedding in the Euclidean space is defined as

$$U^{\phi}(G) = U_E^{\phi}(G) + U_R^{\phi}(G), \tag{2.19}$$

$$U_E^{\phi}(G) = \sum_{E^i} \lambda_i \left\| \phi(E^i(0)) - \phi(E^i(1)) \right\|_2^2, \tag{2.20}$$

$$U_R^{\phi}(G) = \sum_{S_k^j} \mu_{kj} \left\| \phi(S_k^j(0)) - \frac{1}{k} \sum_{i=1}^{k} \phi(S_k^j(i)) \right\|_2^2, \tag{2.21}$$

where λ_i is the coefficient of stretching elasticity of every edge E^i and μ_{kj} are the coefficient of bending elasticity of every rib S_k^j.

Figure 2. A node, an edge and a rib (2-star)

Definition: Elastic net is a particular case of elastic graph which contains only ribs (2-stars) and the vertices of this graph form a regular small-dimensional grid

We want a map $\phi: V \to \mathbb{R}^m$ with good approximation to the dataset X and with low elastic energy. Good approximation of the data is one with minimum mean square distance between the data and its projection to the vertex position.

Approximation error is given as

$$U_A^{\phi}(G,X) = \frac{1}{\sum_{\mathbf{x}\in X} w(\mathbf{x})} \sum_{v\in V} \sum_{\mathbf{x}\in K^v} w(\mathbf{x}) \left\| \mathbf{x} - \phi(v) \right\|_2^2 \tag{2.22}$$

Where $w(\mathbf{x}) \geq 0$ are weights attached to points, which reflect the importance attached to some points in the data and $K^{v_i} = \left\{ \mathbf{x}_i \,/\, v_j = \arg\min_{v_k \in V} \left\| v_k - \mathbf{x}_i \right\|_2^2 \right\}$.

Therefore elastic net seek to the optimal map ϕ_{opt} which minimizes the total energy function

$$U^{\phi} = U_A^{\phi}(G,X) + U^{\phi}(G) \tag{2.23}$$

The elastic net is characterized by internal dimension $\dim(G)$. Every node v_i is indexed by discrete values of internal coordinates $\{v_1^i,\ldots,v_{\dim(G)}^i\}$ such that nodes close on the graph have similar internal coordinates.

The quadratic form of the smoothness penalty given by the elastic energy makes this method computationally efficient and the geometric approach to the construction allows dealing with missing data. When the elastic coefficient is zero (zero elasticity), we have a completely unstructured *k-means* clustering and for rigid rectangular elastic net (i.e. elastic nets with high bending and low stretching energy) we have estimators close to PCA. Varying the elastic energy between these two extreme cases produces non-linear approximation to the principal curve with varying complexity.

Elastic Maps

In elastic net, data are projected to the closest nodes leading to a grid approximation of the data. The grid approximation of data enhances the speed of the projection without losing too much information when the grid resolution is good enough. However, this leads to estimation bias; elastic map method was developed in order to reduce this bias.

Definition: Elastic map is a continuous manifold $Y \in \mathbb{R}^m$ constructed from the elastic net as its grid approximation using some between-node interpolation procedure. For example a piecewise linear elastic map can be built by using piecewise linear interpolation between nodes.

Definition: Elastic principal manifold of dimension *s* for a dataset X is an elastic map, constructed from elastic net Y of dimension *s* embedded in $\mathbb{R}^m$ using such an optimal $\phi_{opt}: Y \in \mathbb{R}^m$ that corresponds to minimal value of the functional

$$U^{\varphi}(X,Y) = MSD_w(Y,X) + U^{\varphi}(G). \tag{2.24}$$

Where the weighted mean squared distance $MSD_w(Y,X)$ from the dataset X to the elastic net Y is calculated as the distance to the finite set of vertices $\left\{y^1 = \phi(v_1),\ldots,y^k = \phi(v_k)\right\}$

Elastic Maps have been applied to visualization of microarray [41] in which it outperforms PCA in terms of data approximation, representation of between point distance structures, preservation of local point neighbourhood and representing point's class in low- dimension spaces. It has also been applied in visualization of economic and sociological tables, natural and genetics text and recovery of missing values in geophysical time series.

2.3.5 Local PCA

We have seen that certain complex datasets are not well approximated (in terms of MSD) by PCA. An example is the case of a dataset in which the covariates have nonlinear relationship which led to the need for nonlinear approximation of such dataset as previsously discussed.Other complex datasets could be characterized by the term 'disconnected' in space of data or 'branched' and PCA may not provide the best

approximation to such datasets, leading to the need to approximate the data locally in dataspace.

Even though there are several ways in which the term 'local' has been interpreted, early approach to local PCA can be traced to [14][29], in which dataset were partitioned into clusters and then PCA performed on each cluster. This cluster-wise PCA was developed as an exploratory data analysis tool to understand the intrinsic structure of data. This can be seen as a generalization of the *k-means* clustering method, in the sense that in *k-means* method, clusters centre on points, but in cluster-wise PCA, the cluster is around a hyperplane segment.

Since cluster-wise PCA require the partitioning of the dataset into clusters usually using *k-means* algorithm, it is also plagued by the problem of initialization. To overcome the problem with initialization, Einbeck et al [24] developed Recursive Local PCA in which the partitions are built up from a single partition. This idea is akin to classification and regression trees (CARTs). This algorithm yield disconnected lines and hyperplane segments and can be seen as finding tangent approximation to principal curve. This algorithm can cope with branched datasets and disconnected datasets. Also similar work by Verbeek et al (2002) developed the *k-segment algorithm* for finding principal curves in which lines segment are fitted to the data and connected using polygonal lines as an approximation of principal curve [104].

In [20] a 'bottom-up' approach was introduced called principal oriented points (POP) based on the variance-maximization definition (definition 2). For a point $\mathbf{x} \in X$ we find the conditional mean of the hypeplane which minimizes the variance of the normal distribution conditioned to belong to that hyperplane. This hyperplane is orthogonal to the first principal component and the first principal component passes through this conditional mean point. Repeating this for different $\mathbf{x} \in X$ generate several conditional means called the principal oriented points (POPs) and the one-dimensional curve running through this points is called principal curve of oriented points (PCOPs). POPs and PCOPs can be seen as an adaptation of localized PCA where 'locality' consists in calculating local mean points and local principal direction and locality is defined by using kernel functions to define the effective radius of

neighbourhood in the data space. POPs do not use Eigen-decomposition of the data and can be computationally expensive due to large number of cluster analysis.

A simpler approach based on local tracing of principal curve was developed in [24] and called Local Principal Curves (LPC). The algorithm is given below

Algorithm for Local principal curve

1. Select a starting point $\mathbf{x}_0$ randomly and step size t_0 . Set $\mathbf{x} = \mathbf{x}_0$.
2. Calculate the local centre of mass $\boldsymbol{\mu}^{\mathbf{x}}$ at $\mathbf{x}_0$.
    a. $\boldsymbol{\mu}^{\mathbf{x}} = \sum w_i^{\mathbf{x}} \mathbf{x}_i$
    b. where $w_i^{\mathbf{x}} = K_H(\mathbf{x}_i - \mathbf{x}_0) \Big/ \sum_{i=1}^{n} K_H(\mathbf{x}_i - \mathbf{x}_0)$, $K_H(\cdot)$ is a *m-dimensional* kernel function and H is a $m \times m$ bandwidth matrix.
3. Estimate the local covariance matrix $\sum^x = (\sigma_{jk}^x)$ at $\mathbf{x}$ using
    a. $\sigma_{jk}^x = \sum_{i=1}^{N} w_i^x (\mathbf{x}_{ij} - \boldsymbol{\mu}_j^x)(\mathbf{x}_{ik} - \boldsymbol{\mu}_k^x)$ and compute the Eigen-decomposition.
    b. Let $\mathbf{v}^{\mathbf{x}}$ be the loading vector for the first principal component computed locally at $\mathbf{x}_0$,
4. Setting $\mathbf{x} := \boldsymbol{\mu}^{\mathbf{x}} + t_0 \mathbf{v}^{\mathbf{x}}$, one finds the updated value of $\mathbf{x}$.
5. Repeating steps 2 to 4 until the sequence of $\boldsymbol{\mu}^{\mathbf{x}}$ remains approximately constant (which implies that the end of the dataset is reached). Then set again $\mathbf{x} = \mathbf{x}_0$, set $\mathbf{v}^{\mathbf{x}} = \mathbf{v}^{\mathbf{x}}$ and continue step 4

This algorithm only produces a 1-dimensional curve approximation of the data. To generalize this idea, Einbeck suggested using a *d-dimensional* mesh much like the elastic net algorithm of [41] [45] but in a 'bottom-up' method thereby requiring no initialization.

Local PCA finds application in image feature extraction and recognition [65].

2.3.6 Branching Principal Component

For certain complex dataset which can be classified as 'branched', linear PCA and principal curve do not provide satisfactory approximation since it does not effectively approximate the branches.

Kegl et al (2002) extended the polygonal line algorithm to approximate dataset which are 'branched'. This idea was called Piecewise Linear Skeletonization [64]. It starts with an initialization method to capture the approximate topology of the dataset and restructuring operations to improve the structural quality of the graph. This method was applied to isolated handwritten digits and images of continuous handwriting.

In series of papers [40] [46][70], Gorban et al extended their previous works on elastic nets and maps in order to model branching datasets more effectively using topological grammar. Topological grammar is a set of operations which can be applied to modify the structure of principal graphs after which the elastic energy (see equations 2.19, 2.20, 2.21) of the graph is minimized. Due to the growing nature of the principal graph the class of graphs considered in the optimization step is limited to graphs whose embedding satisfies certain restriction on the following complexity criteria:

Geometric complexity- which measure how far a principal object deviates from 'ideal configuration'. This can be seen as a measure of non-linearity.

Structural complexity–penalizes for structural elements. This is usually a non-decreasing function of the number of vertices, edges and *k-stars* of different orders.

Construction complexity- defined with respect to graph grammar measures the number of elementary transformations needed to construct the graph from the simplest graph (one vertex, zero edges)

They propose a pluriharmonic graph as 'ideal configuration' where pluriharmonic graph is defined as a map $\phi: v \rightarrow \mathbb{R}^m$ defined on vertices of graph *G* such that for any *k-star* $S_k^j \in S_k$ with central vertex $S_k^j(0)$ and neighbouring vertices $S_k^j(i), i = 1,\ldots,k$ the equality holds

$$\phi(S_k^j(0) = \frac{1}{k}\sum_{i=1}^{k}\phi(S_k^j(i)) \tag{2.24}$$

The simple case of topological grammar {“add a node”, “bisect an edge”} is equivalent to the construction of principal trees which are acyclic primitive elastic principal graphs. This work was further generalized to principal cubic complexes. For further reading see [40] and [46].

This method has application in visualization of datasets. Notable example of this is in visualization of gene expression in human tissues and visualization of microarray data [47].

2.4 Tensor PCA

PCA has been extended to deal with dimension reduction problem for tensor objects. A tensor is a multidimensional array and an *N*th-order tensor can be defined as the element of tensor product of *N* vector spaces; this is a generalization of vectors and matrices. In recent years, especially in the field of computer vision and pattern recognition, the objects of interest are best described as tensors; examples of such tensor objects are 2-D/ 3-D images and video sequences. A colored image for example is an object with column, row and colour mode and can be considered as a 3rd order tensor. One of the properties of a tensor object is that entries are often highly correlated with surrounding entries and the presence of redundancy in the data suggests that we can find a suspace of lower dimension which approximates the tensor object while retaining most of the information contained and also preserving the underlying structure of the system represented by the tensor objects.

One approach to dimension reduction for tensor objects is to vectorise the tensor objects and then use PCA for dimension reduction. However vectorising tensor objects lead to high dimensional vector representation of each tensor object, which can suffer from the curse of dimensionality [10] especially since in high dimension all feasible training samples sparsely populate the input space. To demonstrate this, let us consider tensor objects with dimension $(100\times 50\times 30)$ this will be vectorize to vectors of dimension $(150{,}000\times 1)$. Even more importantly, PCA on vectorised tensor objects does not take into account the natural structure of the tensor obects (the fact that the objects are tensors) and hence could destroy the important structure(s) of the objects being approximated. Therefore there is a need to perform PCA on the tensor space directly.

There have been several developments in the approximation of tensor objects; for example [105], [52] and [19].

Another approach to dimension reduction of tensor objects is to compute PCA directly in the tensor space, using multiple orthogonal transformations to transform tensor objects to other tensor objects with lower dimension while capturing as much variance in the tensorial data as possible. Let us briefly discuss this approach to computing PCA for tensor objects as formulated in [76]. Given an *N*th- order tensor space $T = \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \ldots \otimes \mathbb{R}^{I_N}$ where $\mathbb{R}^{I_i}$ is the *i*th vector space of dimension I_i; any tensor $A \in \mathrm{T}$ can be decomposed as follows:

$$A = S \times_1 U^{(1)} \times_2 U^{(2)} \times \ldots \times_N U^{(N)}, \tag{2.25}$$

where $S = A \times_1 U^{(1)^T} \times_2 U^{(2)^T} \times \ldots \times_N U^{(N)^T}$ and $U^{(i)} = (\mathbf{u}_1^{(i)} \mathbf{u}_2^{(i)} \ldots \mathbf{u}_{I_i}^{(i)})$ is a $I_i \times I_i$ orthogonal matrix. This decomposition can be written as linear combination of $I_1 \times I_2 \times \ldots \times I_N$ rank-1 tensors given as

$$A = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \ldots \sum_{i_N=1}^{I_N} S(i_1, i_2, \ldots, i_N) \mathbf{u}_{i_1}^{(1)} \circ \mathbf{u}_{i_2}^{(2)} \circ \ldots \circ \mathbf{u}_{i_1}^{(N)}. \tag{2.26}$$

To reduce the dimension of the tensor object *A*, for each mode *i,* we seek $P_i < I_i$ orthonormal basis vectors of the *i*th mode vector space to form a tensor subspace $\hat{T} = R^{P_1} \times R^{P_2} \ldots \times R^{P_N}$; the tensor object *A* is then projected to the tensor subspace. Let $\tilde{U}^{(i)}$ be the $I_i \times P_i$ matrix containing the P_i orthonormal *i*-mode basis vector, the projection of *A* to the subspace $\hat{T}$ is defined as

$$Y = A \times_1 \tilde{U}^{(1)^T} \times_2 \tilde{U}^{(2)^T} \times \ldots \times_N \tilde{U}^{(N)^T}. \tag{2.27}$$

Now given a set of *M* tensor samples $\{A_i, i = 1, \ldots, M\}$, the total scatter of these set is defined as

$$\psi_A = \sum_{m=1}^{M} \| A_m - \bar{A} \|_F^2, \tag{2.28}$$

where $\bar{A} = \frac{1}{M} \sum_{m=1}^{M} A_m$ is the mean tensor and $\|\cdot\|_F$ is the Frobenius norm.

Therefore the Tensor PCA as formulated in [76] seeks to define a multilinear transformation that maps the original tensor space T into a tensor subspace $\tilde{T}$ such that it captures most of the variation observed in the original tensor objects where the variation is measured by the tensor scatter. That is PCA on tensor seek to determine N projection matrices $\{\tilde{U}^{(i)} \in R^{I_i \times P_i}, i = 1,\ldots,N\}$ that maximize the scatter ψ_A:

$$\{\tilde{U}^{(i)}, i = 1,\ldots,N\} = \arg \max_{\tilde{U}^{(1)},\tilde{U}^{(2)},\ldots\tilde{U}^{(N)}} \psi_A . \tag{2.29}$$

In practice, an iterative approach is taken to find the solution to the optimation problem 2.29. However we remark that while PCA produces uncorrelated features, tensor PCA features are not uncorrelated in general (although the transformation in each mode is orthogonal).

2.5 Projection Methods versus Manifold Modelling Methods

Finally we give a comparison between approximation of data using projection methods and manifold modelling methods. By projection methods, we mean methods that project the data to some linear subspace, which include the classical PCA and also weighted PCA. By manifold modelling methods, we mean methods that try to model the topology of the data using curves and surfaces such as principal curves, elastic maps, SOM and others. It should be noted that PCA can also be said to be a manifold modelling technique.

Some advantages of projection methods compared with Manifold modelling methods

1) In projection methods, one can quantify the importance of each of the principal directions. For example, with variance maximization as the goal, one can quantify the contribution of each principal component to the total variance of the dataset. This is not the case in manifold modelling methods especially since the resulting nonlinear manifold sometimes can be embedded in all dimensions of the usually high–dimensional dataspace.

2) In projection methods, it is easier to approximate a dataset with a subspace of dimension higher than three. This is usually computationally expensive and sometimes not feasible with manifold modelling techniques.
3) The principal components in projection methods are easier to interpret since they are a linear combination of the original data. The principal components of manifold modelling methods are sometimes not easy to interpret. See [71].
4) Sometimes the manifold modelling methods can unrecognizably deform the topology of the data. A good example is the application of principal curve or SOM to data distributed along a spiral. See [24].
5) Computation complexity of projection methods is usually low compare to manifold modelling methods. In fact, the projection matrix in projection methods can be stored in memory and can be used when new data undergo the same transformation.

Some advantages of Manifold modelling methods compared with projection methods.

1) As previously mentioned, application of linear method to nonlinear data may be inefficient. For example curved dataset can be approximate by manifold methods with much lower dimension compared to approximating such dataset with projection methods. Also modelling nonlinear data with linear methods can be misleading when nonlinear structure is of interest [84].
2) Dimension reduction leads to distortion of data. PCA in particular favours direction with large distances since its principal component align along the directions with large variance. The implication of this is that the preservation of local structures is not guaranteed. However this sometimes can be improved in manifold modelling since it models the topology of the dataset. For further reading on this see [41].
3) Due to the better local neighbourhood preservation that can be achieved for certain kind of dataset, for such datasets, manifold method can be a better pre-processing step to adopt before classification task.

2.6 Initial Approximation for Manifold Learning Methods – A case study

In manifold learning methods, one of the standard problems is determining the initial approximation as everything depends crucially on the initial approximation. In this section we will investigate the effect of initialization on manifold modelling methods. However, because it is impossible to solve this problem for all manifold modelling methods, Self-Organising Maps (SOM) has been selected for a case study on initialization of manifold learning methods.

We compared the performance of two initialization methods which are popularly used for manifold learning methods and SOM in particular. The two initialization methods are Random Initialization (RI) and principal component initialization (PCI). To further understand the performance of these initialization methods, the datasets were classified as linear, quasilinear or nonlinear based on the topology of the data. The performance of the initialization methods were compared for the various classifications of the data and we demonstrate in this case study for SOM that the performance of each of the initialization methods depends on the class of data and for some class of data RI performs better than PCI.

2.6.1 SOM-Background and Algorithm

As discussed in section 2.3.2, SOM can be considered as a non-linear generalization of PCA [106]. It approximates a high dimension data manifold using a regular low dimensional grid (called maps) using a neighbourhood function to preserve the topological properties of the data. SOM is a type of artificial neural network which uses unsupervised learning and was introduced by Kohonen (1982) [67].

Like clustering algorithms [86], the quality of learning of SOM is greatly influenced by the initial conditions: initial approximation (initial weight of the map), the neighbourhood function, the learning rate, sequence of training vector and number of iterations. [7], [94]. Several initialization methods have been developed over the years and can be broadly grouped into two classes: random initialization and data analysis based initialization [7].

For random initialization method, the initial weights are selected randomly from the dataspace. Due to many possible initial configurations, several attempts are usually made and the best initial configuration is chosen. However, for the data analysis based methods, certain statistical data analysis and data classification methods are used to determine the initial configuration; a popular method is selecting the initial weights from the same space spanned by the linear principal component of the data. Modification to this method was done by [7] and over the years other initialization methods have been proposed. For example see [26].

We will be comparing the performance of two initialization methods in terms of the quality of learning of the SOM that result from using the initialization methods on the same dataset. The quality of learning is determined by the fraction of variance unexplained (FVU) [83]. To ensure an exhaustive study, synthetic datasets distributed along various shapes of only 2-dimensions are considered in this study and the map is 1-dimensional. 1-dimensional SOMs are very important, for example, for approximation of principal curves. The experiments were performed using the PCA, SOM and GSOM applet available online [83] and can be reproduced by anybody.

Since the performance of SOM depends on several factors (as earlier mentioned), in order to marginalize the effects of other factors that can influence the result, the learning processes have been subjected to the same condions. Based on this, the SOMs learning has been done with the same neighbourhood function and learning rate for the initialization methods studied. To marginalize the effect of the sequence of training vectors, the applet adopts the batch learning SOM algorithm [26] and [79] described in the next section. For each dataset and initialization method, the data set was trained using three or four different values of neuron k.

Background

Next we discuss the batch algorithm and the SOM algorithm used for the case study.

The SOM is an artificial neural network which has a feed-forward structure with a single computational layer. See section 2.3.2 for the SOM algorithm as proposed by Kohonen.

The Batch Algorithm

In approximating a dataset using SOM, the input vectors are presented sequentially during the training step. The sequence in which the vectors are presented during training influences the resulting map. For some sequences the resulting map may not be globally optimal. One way to deal with this is using batch algorithm. The batch algorithm is a variant of the SOM algorithm in which the whole training set is presented to the map and afterwards the weights are adjusted with the net effect over the samples [27][82]. The algorithm is given below.

Put the set of data point associated with each neuron equal to empty set: $C_i = \emptyset$.

1. Present an input vector $\mathbf{x}_s$ and find the winner neuron, which is the weight vector closest to the input data.

$$i = \arg\min_{1 \le j \le k} \left\| \mathbf{x}_s - \mathbf{w}_j(t) \right\|, \quad C_i \leftarrow C_i \cup \{s\}.$$

2. Repeat step 1 for all the data points in the training set.
3. Update all the weights as follows

$$w_i(t+1) = \sum_{j=1}^{k} \eta_{ij}(t) \sum_{s \in C_j} \mathbf{x}_s \Big/ \sum_{j=1}^{k} \eta_{ij}(t) \tag{2.30}$$

where $\eta_{ij}(t)$ is the neighbourhood function between the *i*-th and *j*-th neuron at time t, and k is the number of neuron. For this case study the batch algorithm was adopted for SOM learning.

SOM learning algorithm used by the applet

Before learning, all C_i are set to the empty set ($C_i = \emptyset$), and the steps counter is set to zero.

1. Associate data points with neurons (form the list of indices

$$C_i = \left\{ l : \left\| \mathbf{x}_l - w_i \right\| \le \left\| \mathbf{x}_l - w_j \right\| \forall i \ne j \right\}$$

2. If all sets C_i evaluated at step 1 coincide with sets from the previous step of learning, then STOP.

3. Calculate the new values of coding vectors (weights) by formula (2.30)
4. Increment the step counter by 1.
5. If the step counter is equal to some specified number (for this study 100), then STOP.
6. Return to step 1.

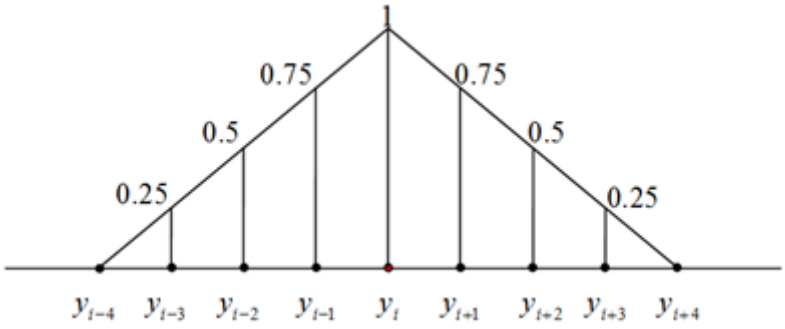


Figure 3: The B-spline neighbourhood function with $h_{\max}$=3.

The neighbourhood function used for this applet has the simple B-spline form given as a B-spline with $h_{\max}=3$: $\eta_{ij}=1-|i-j|/(h_{\max}+1)$ if $|i-j|<h_{\max}$ and $\eta_{ij}=0$ if $|i-j|\geq h_{\max}$.

2.6.2 Fraction of Variance Unexplained

For this study, data are approximated by broken lines (SOM) [83]. The dimensionless least square evaluation of the error is the *Fraction of Variance Unexplained* (FVU). It is defined as the fraction: [The sum of squared distances from data to the approximating line / the sum of squared distances from data to the mean point].

$$FVU = \sum_{i=1}^{n} d(\mathbf{x}_i)^2 \Big/ \sum_{i=1}^{n} \|\mathbf{x}_i - \bar{\mathbf{x}}\|^2, \qquad (2.31)$$

where $\bar{\mathbf{x}}=\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$ is the mean point and the distance d_i^2 from data point $\mathbf{x}_i$ to the approximating line is given as the length of $\mathbf{x}_i$ to its closest point on the approximating line. This definition allows us to evaluate FVU for SOM using PCI and RI as initialization methods.

For the given array of coding vectors $\{y_i\}$ $(i=1,2,\ldots,m)$, we have to calculate the distance from each data point $\mathbf{x}$ to the broken line specified by a sequence of points

$\{y_1, y_2, \ldots, y_m\}$. For the data point $\mathbf{x}$, its projection onto the broken line is defined, that is, the closest point. The square of distance between the coding vector y_i and the point $\mathbf{x}$ is $p_i(\mathbf{x}) = \| \mathbf{x} - y_i \|^2 \ (i = 1,2,\ldots,m)$.

Let us calculate the squared distance from the data point $\mathbf{x}$ to the segment $[y_i, y_{i+1}]$ $(i = 1,2,\ldots, m)$. For each *i*, we calculate projection of a data point onto a segment. $l_i(x) = \langle \mathbf{x} - y_i, y_{i+1} - y_i \rangle / \| y_{i+1} - y_i \|_2^2$. If $0 < l_i(x) < 1$ then the point, nearest to $\mathbf{x}$ on the segment $[y_i, y_{i+1}]$, is the internal point of the segment. Otherwise, this nearest point is one of the segment's ends.

Let $0 < l_i(x) < 1$ and c be a projection of $\mathbf{x}$ onto segment $[y_i, y_{i+1}]$, then $\| c - y_i \|^2 = (l_i(x) \| y_{i+1} - y_i \|)^2$ and from Pythagorean Theorem, the squared distance from $\mathbf{x}$ to the segment $[y_i, y_{i+1}]$ is $r_i(x) = \| \mathbf{x} - y_i \|^2 - (l_i(x) \| y_{i+1} - y_i \|)^2$. (see figure 4).

Let $p(\mathbf{x}) = \min\{p_i(\mathbf{x}) \mid i = 1,2,\ldots,m\}$ and $r(\mathbf{x}) = \min\{r_i(\mathbf{x}) \mid 0 < l_i(\mathbf{x}) < 1,\ 0 < i < m\}$, then the squared distance from $\mathbf{x}$ to the broken line specified by the sequence of points $\{y_1, y_2, \ldots y_m\}$ is given as $d(\mathbf{x}) = \min\{p(\mathbf{x}), r(\mathbf{x})\}$.

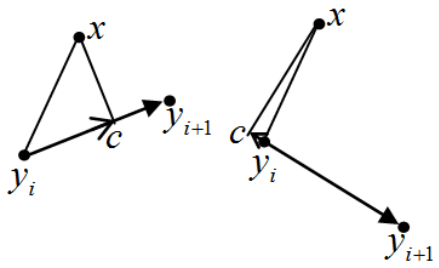


Figure 4. A distance from a point to a segment: two versions of the projection.

2.6.3 Initialization Methods

The objective of this case study is to consider the performance of two different initialization methods for SOM using the FVU as the criterion for measuring the performance or the quality of learning. The two initialization methods compared are:

- *PCA initialization (PCI):* The weight vectors are selected from the subspace spanned by the first *p* loading vectors of the principal components. For this study, the weight vectors are chosen as a regular grid on the first principal component, with the same variance as the whole dataset. Therefore, given the number of weight vectors *k*, the behaviour of SOM using PCA initialization is completely deterministic and results in a

single configuration. PCI does not take into account the distribution of the linear projection results. It can produce several empty cells and may need a post-processing reconstitution algorithm [6]. However, since the PCA initialization is better organized, SOM computation can be made order of magnitude faster comparing to random initialization [77].

Random Initialization (RI): k weight vectors are selected randomly, independently and equiprobably from the data points. The size of the set of possible initial configurations given a dataset increases with the size n of the dataset. The possible choice of initial configuration for a given k (the number of nodes) can becomes enormous (n^k). However, given an initial configuration, the behaviour of the SOM becomes completely deterministic.

2.6.4 Linear, Quasilinear and Nonlinear Data Models

Datasets can be modelled using linear or nonlinear manifold of lower dimension. According to [43] a class of quasilinear model data set was identified. In this study, datasets will be classified as linear, quasilinear or nonlinear. The non-linearity test for PCA [72] can be used to determine whether a linear or nonlinear model is appropriate for modelling a given dataset.

- **Linear Model** – A dataset is said to be linear if it can be modelled using a sequence of linear manifolds of small dimension (in 1-D case, they can be approximated by a straight line with sufficient accuracy). This dataset can be easily approximated by the principal components without SOM. We do not consider such data in this study.

- **Quasilinear Model** – A dataset is called quasilinear [43] if the principal curve approximating the dataset can be univalently and linearly projected onto the linear principal component. It should be noted that the principal curve is projected to the lines and not the nodes. For this study, datasets which fall in the border between nonlinear and quasilinear (in which over 50% of the data can be classified as quasilinear) will also be classified as quasilinear. See examples in figures 5a and 5b.

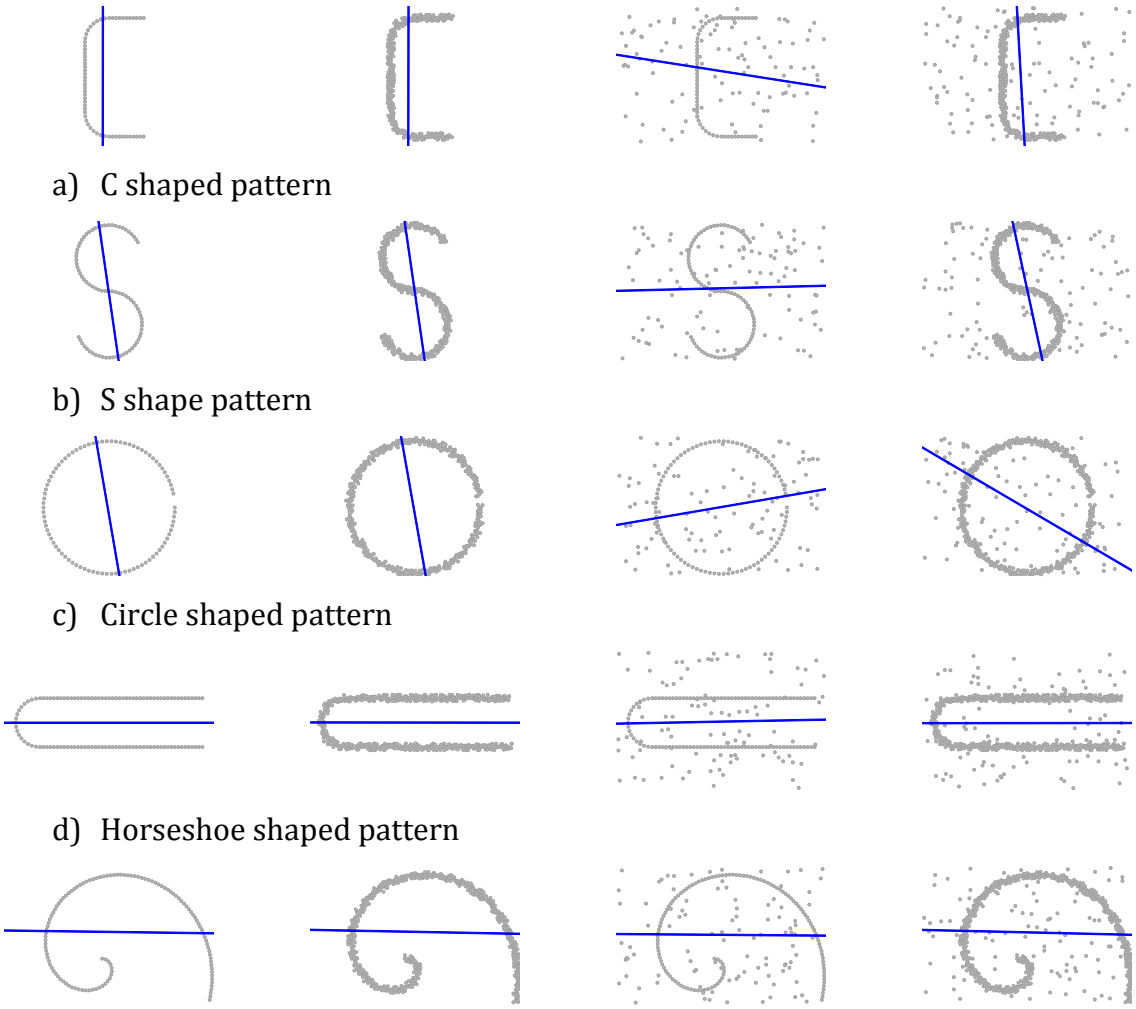


Figure 5. (a) Quasilinear data set; (b) a border case between nonlinear and quasilinear dataset. (c, d, e) nonlinear data set; The first principal component approximations are shown (blue line). The left column contains clear patterns, the second column from the left contains scattered patterns, the second column from the right contains clear pattern with added noise and the right column contains the scattered patterns with added noise.

Nonlinear Model – In this study, we call the essentially nonlinear datasets which do not fall into the class of quasilinear datasets just nonlinear data. See example in figures 5c, 5d and 5e.

Table 1. Classification of patterns models (figure 5).

Etalon	***Clear***	***Scattering***	***Noise***	***Noise and Scattering***
C	quasilinear	quasilinear	nonlinear	quasilinear
S	quasilinear	quasilinear	nonlinear	quasilinear
Circle	nonlinear	nonlinear	nonlinear	nonlinear
Horseshoe	nonlinear	nonlinear	nonlinear	nonlinear
Spiral	nonlinear	nonlinear	nonlinear	nonlinear

2.6.5 Experiments and Analyses

The performance of both initialization methods on datasets with data distributed along different shapes (see figure 4) was studied at values of k = 10, 20, 50 (unless otherwise stated).

Drawing up the Probability Distribution of FVU

For the PCI as mentioned earlier, its yield just one initial configuration given K (this is because equidistant nodes are selected from the subspace of principal component such that the variances are equal).

In drawing up the probability distributions for the RI method, a sample of 100 initial configurations from the space of possible initial configurations for each dataset and each value of k was taken and the resulting FVU computed. . The probability distribution of the FVU was described in terms of mean, median, standard deviation; minimum and maximum (see Table A in the Appendix).

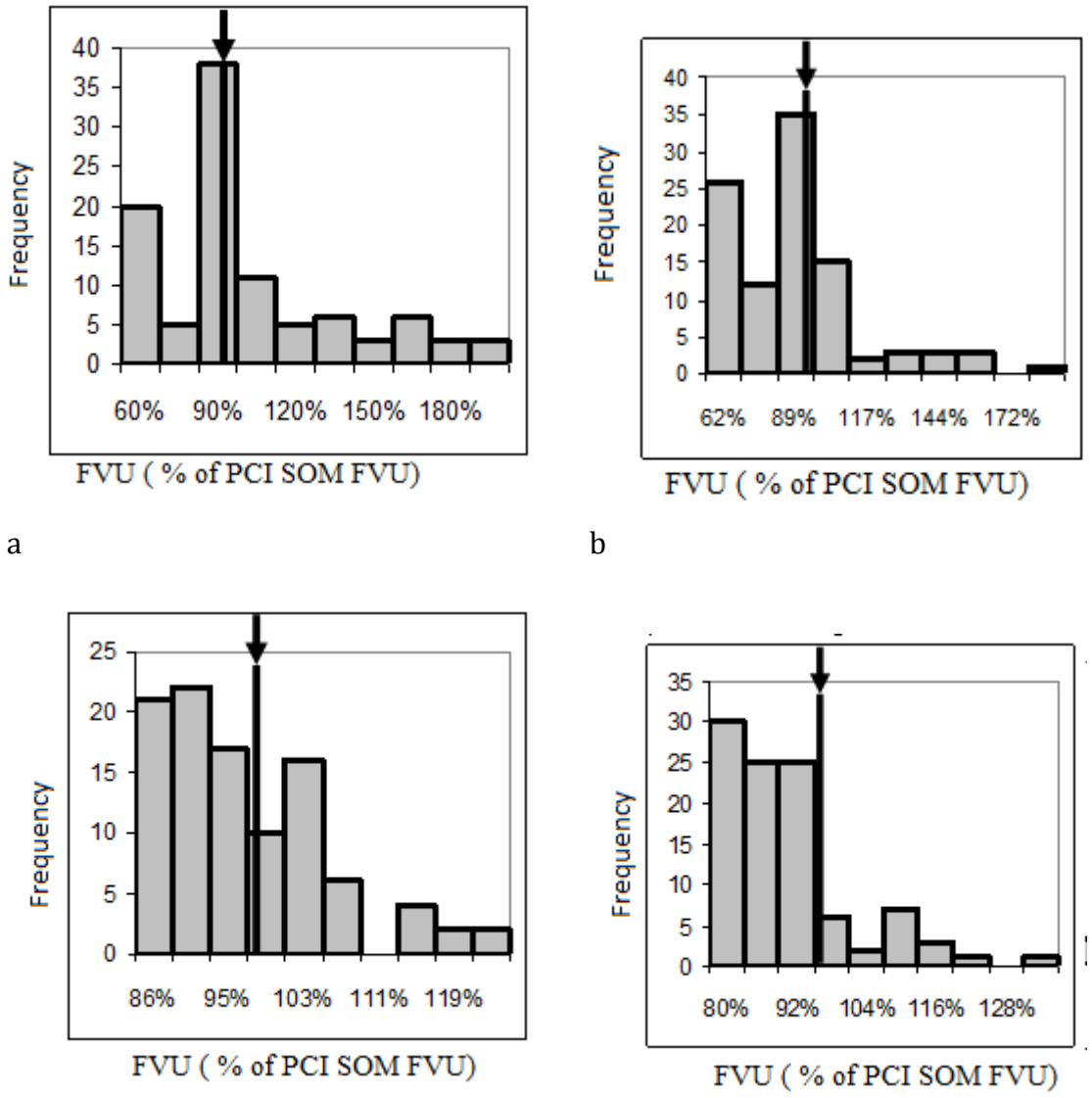


Figure 6. A typical example of distribution of RI SOM FVU in percent of PCI FVU. Vertical line with arrow above corresponds to PCI SOM FVU. Axis's *x* labels are the value which corresponds to left boundary of interval. All four histograms are illustrating the distribution of RI SOM FVU for the spiral pattern with 20 nodes: (a) clear spiral, (b) scattered spiral, (c) noised spiral and (d) scattered and noised spiral

To compare the performance of the two initialization approaches in terms of the FVU, for each *k* we find the number of RI SOM with FVU that is less or equal to PCI SOM (i.e. proportion of resulting map for which the RI outperform PCI) since the PCI have just one configuration and RI have many configurations. In the tables, results are averaged

for various types of pattern smearing (table 2) and for different pattern models (table 3).

Table 2. The results of testing for different kind of patterns

Pattern	***Average fraction of RI SOM with FVU better than for PCI***
Clear	35.00%
Scattered	44.56%
Noised	55.52%
Scattered and noised	64.60%

Table 3. The results of testing for different models

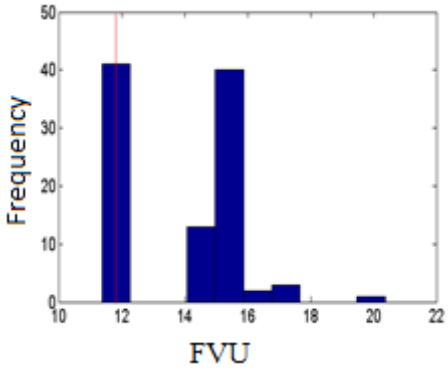
Pattern model	***Average fraction of RI SOM with FVU better than for PCI***
Quasilinear	36.62%
Nonlinear	60.89%

Analysis of Performance and Discussion

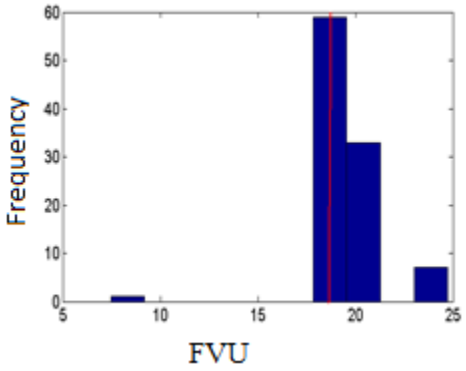
In eight tests (from 100) all RI SOM have FVU that is equal or greater than that of PCI SOM: clear C with 10 nodes, scattered C with 10 nodes, clear circle with 10 nodes, scattered circle with 10 nodes, scattered S with 20 nodes, scattered and noised spiral with 10 nodes, noised circle with 75 nodes and clear spiral with 50 nodes.

Analysing the performance shows that RI tend to perform quite well for nonlinear datasets. An interesting result was obtained for the spiral dataset (figure. 7a, b). For 10 nodes, 41% of RI realisations give better value of FVU than PCI, for 20 nodes this

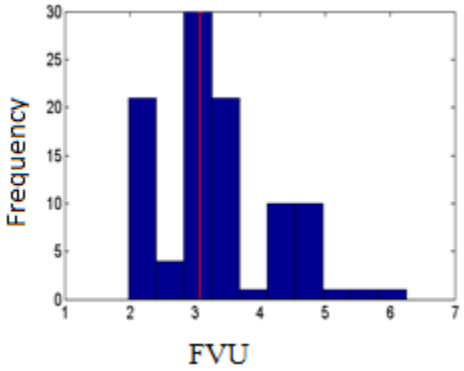
percent increases to 55% but for 50 nodes PCI gives better result than 99% of RI (figure 7c).



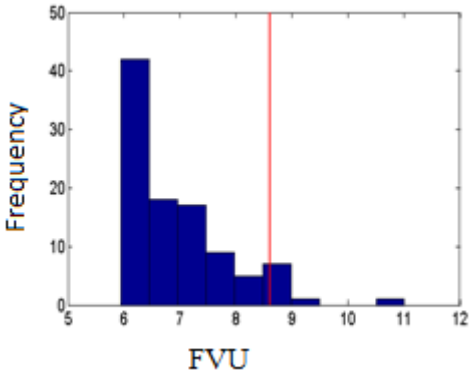
a1) SOM approximation using 10 Nodes



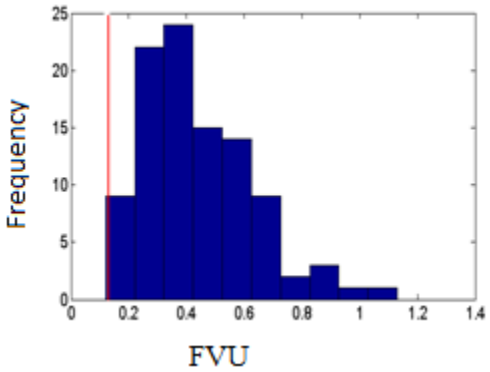
b1) SOM approximation using 10 Nodes



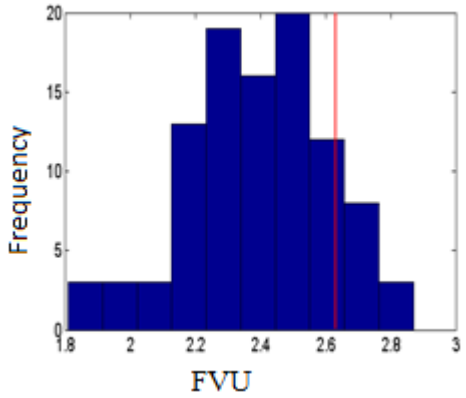
(a2) SOM approximation using 20 Nodes



(b2) SOM approximation using 20 Nodes



(a3) SOM approximation using 50 Nodes



(b3) SOM approximation using 50 Nodes

Figure 7(a1-a3). Spiral Data Set.

Figure 7(b1-b3). Spiral Data Set With Noise.

We can conjecture that the SOM learning dynamics has many Stable Steady States SOMs (SSSSOMs). Sometimes the PCI can hit into a basin of attraction of a SSSSOM with low value of FVU. We have no different possible explanation of the surprising result presented in figure 7a3.

However as can be observed in figure 7b, the presence of noise affects the performance of the initialization methods. In particular, the surprisingly good performance of PCI seen in figure 7a3 is destroyed by noise (figure 7b3), and with noise the relative performance of PCI monotonically decreases with the number of nodes. In general, we can conclude that for essentially nonlinear datasets PCI performs not better or even worse than the median of RI. The role of noise will be discussed later. However, the performance of RI is inconclusive regarding quasilinear datasets. While the performance was good for the S shaped dataset the performance for the C shape was not as expected. For the approximation of the C shaped dataset by 1D SOM with 10 nodes all the results of RI were better than PCI. Nevertheless, it should be mentioned that the difference between the values of FVU for this case is rather small. It does not exceed 4% of the minimal value of FVU. In that sense, we can say that the performance of PCI almost coincides with the quality of RI

Further analysis was performed to determine factors that could influence the performance of the initialization methods. By considering the effect of the underlisted factors on the proportion of RI that outperforms PCI and using regression analysis the following were observed:

a) Increase in Nodes (k): there was no relationship that could be established which indicates that increase in *k* significantly influence the performance of RI compared to PCI

b) Number of unique final configurations in sample: Even though the number of unique final configurations in the population is not well defined, there is a significant correlation between the number of unique final configurations in the sample and the performance of RI for quasilinear datasets. This correlation however does not exist for nonlinear data. See the result in table 4 below for quasilinear datasets. The data is given in table C in the Appendix.

c) Increase in the data points (N): increase in the dataset size does not significantly influence the performance of RI compared with PCI.

Table 4. The correlation between the number of unique final configurations in the sample and the performance of RI for quasi-linear datasets.

Model		*Unstandardized Coefficients*		*Standardized Coefficients*		
		B	*Std. Error*	*Beta*	*t*	*Sig.*
1	(Constant)	.713	.121		5.911	.000
	Unique	-.010	.003	-.749	-3.576	.005

a. Dependent Variable: Proportion

d) Presence of noise: It was observed that the presences of noise in the spiral dataset tend to influence the performance of PCI. Further studies show that the presence of noise in quasi-linear data sets affects the performance of PCI. This is because noise can affect the principal component and also the principal curve of the dataset, which can affect the classification of dataset especially for quasi-linear datasets. An illustration is given in figure 8.

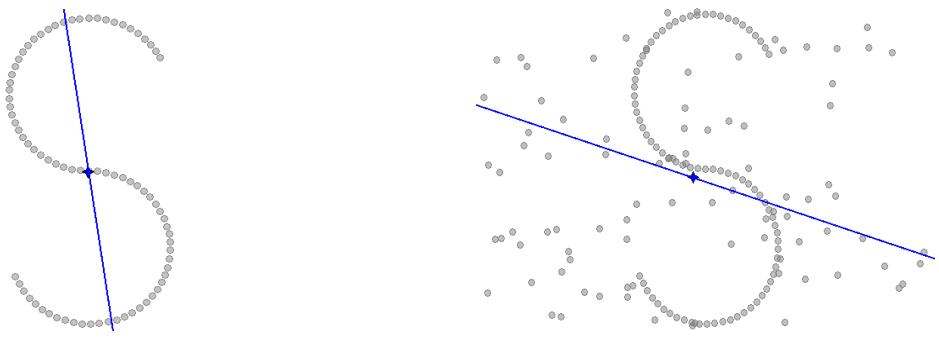


Figure 8. S shaped dataset (It is almost quasilinear). S shape dataset with noise (it

becomes essentially nonlinear). The straight blue lines are the first principal component approximation line.

The results of tests show that the RI SOM may perform better than PCI SOM for any models and any kind of pattern. Nevertheless, there exist a small fraction of patterns for which RI SOM does not overperform PCI SOM. Let us estimate the number of RI SOM which we can learn to obtain the FVU less than that of PCI with probability 90%. Let us have pattern with quasilinear model. In this case we estimate the probability of obtaining RI SOM with FVU worse than for PCI SOM is 63.38% (100-36.62). Probability of obtaining 5 RI SOM with FVU not less than for PCI SOM is $0.6338^5 \approx 0.10$. Therefore, it is sufficient to try 5 RI SOM to obtain FVU less than for PCI SOM with probability $\approx 90\%$. All these numbers are valid for our choice of patterns and their smearing (see figure 5).

We can conclude that the widely accepted presumption about advantages of PCI SOM initialization seems not to be universal. The frequency of RI SOM with FVU that is less than FVU for PCI SOM is 61% for nonlinear patterns in our case study. This means that three random initializations are sufficient to obtain the FVU which is less or equal to PCI SOM FVU with probability 95% in these cases. For quasilinear patterns it is sufficient to try RI SOM 5 times to obtain FVU less than PCI SOM with probability 90%. The computational resource which is necessary to calculate principal components and to learn SOM appears to be approximately equal but the quality of the best RI SOM is often, after several generations, better than that of PCI SOM.

2.7 Conclusion

In this chapter we examined some generalizations and extensions of PCA. In partiticular we examine two major approaches in which there has been considerable development. The first is the methods that generalize PCA by introducing weights on data points, variables in the dataset or distances of data projections. We noted that the use of correlation matrix (rather than covariance matrix) can be framed in this context. We also mentioned that the use of weights can be used to incorporate prori knowledge available into the analysis of principal components. An example is using weight to preserve class structures for labelled data or to minimize the effect of outliers or some other influential data points.

The second approach is methods that generalize PCA by using nonlinear functions to approximate the dataset. Such methods tend to model the manifold of the dataset. Especially for dataset with nonlinear relationship among the covariates, manifold modelling methods generally seek to take advantage of this nonlinear relationship in the dataset to produce a more efficient approximation. Examples of such methods include principal curves, SOM, elastic nets and maps, kernel methods e.t.c. We also consider methods that can approximate efficiently dataset which are 'branched' or 'disconnected'.

Finally we conducted an empirical study on initialization problem of manifold modelling techniques using SOM as a case study. We were able to compare the performance of two initialization approaches which is common in manifold learning. The classification of dataset into linear, quasilinear or nonlinearclass has been important for understanding the dynamics of manifold learning and for selection of initial approximation. We have shown that the widely accepted view about advantages of PCI SOM initialization seems not to be universal and in case of SOM, we can conclude that the hypothesis about advantages of the PCI is definitely wrong for essentially nonlinear datasets.

Chapter 3

Multiscale Principal Component Analysis (MPCA)

3.1. Introduction

Principal component analysis of a dataset reveals the underlying structure of the data. However, according to *definition 2,* PCA favours structures with large variance as directions with large variance are assumed to be informative. Equivalently PCA seeks subspace of the original data in which the pairwise distance distortion is minimized for the data projection to such subspace, which means that PCA seeks to maximized the sum of the pairwise distances of data projection (see *definition 4)*. The implication of this is that PCA favours structures with large distances. For certain complex data which have different structures at different distances, PCA is only able to identify structure with large distance and may completely obfuscate other interesting structures represented by smaller distances in the data.

We illustrate this with a simple example of a dataset with different structures at different distances. Let us consider a dataset which contains outliers, where outlier is defined as a point with large distance to other points in the dataset. As can be observed in figure 9a and 9b, if outliers are removed then this data distributed along a line.

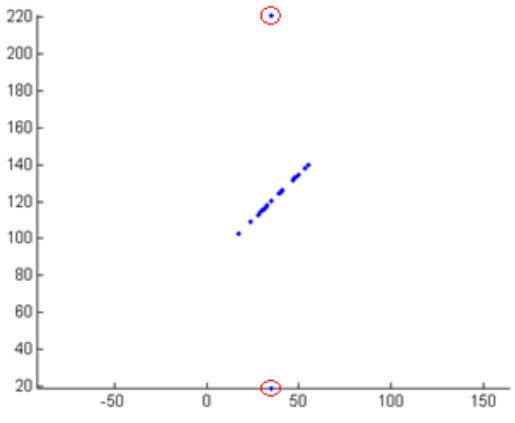


Figure 9a. Scatter plot of a dataset distributed along a line with 2 outliers (marked with red circle).

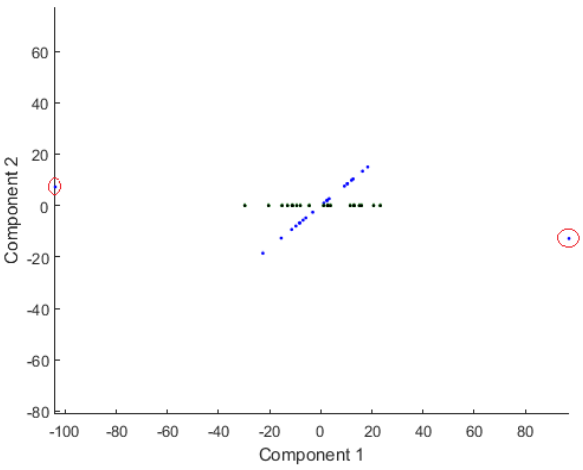


Figure 9b. Scatter plot of the principal components of the data (in blue colour) and scatter plot of the principal components of the data with outliers removed (in black colour)

The biplot of the example above is given in figure 10. A biplot is useful for visualizing the magnitude (represented with the blue lines), and sign of each variables' contribution to the first two or three principal component and how each data point (represented as red point) is represented in terms of those components.

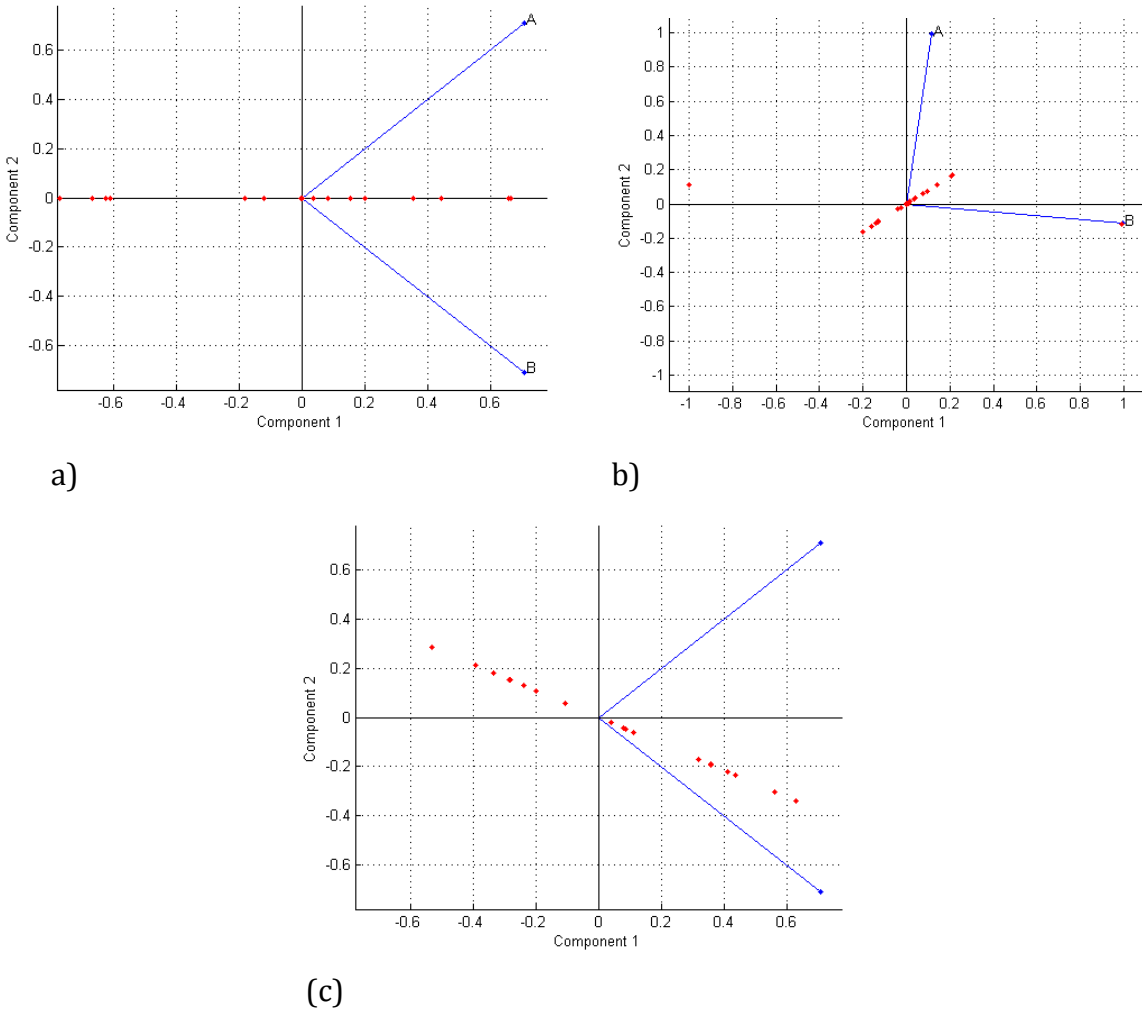


Figure 10. (a) is the biplot of the data without outliers, (b) is the biplot of the dataset with outliers and (c) is the biplot of the dataset when normalized to unit variance. We observe that in figure (a) the dataset is distributed along the first PC (1 dimension) whereas this was not captured in figure (b) and (c)

From figure 10 above, we see that the first principal component align in the direction of the outliers and in this case at the expense of the more interesting underlying structure of the data. Of course if the first principal component does not recognise the more interesting structure in the data, other principal components may not be able to detect

such structure(s) due to the orthogonality constraint imposed on principal components which ensure that all subsequent principal components are selected orthogonally to previous ones. Thus PCA may miss out on trends, patterns and other underlying structure(s) of the data which exist at smaller distances. One popular approach employed to reduce the effect of outliers on PCA is to normalize the variables to unit variance. However we see that this may not necessarily reveal the structure in the data as we observe in figure 10c. We should comment that detection of outliers and other influential data points in high dimension can be challenging.

We can therefore say that PCA being a non-parametric statistical tool, focus on the global structure of the dataset and to explore for underlying structures there is a need to introduce some form of localization into the analysis. Much work has been done in the area of localization of PCA in the space of the data vectors; see [14] [29] [24] [104] [20] and also the discussion in section 2.3.5. In this chapter, using an extension of *definition 4*, we propose a new form of localization called "localization in scale" to study the underlying structures of data at various distribution of pairwise distances.

3.2 Mathematical Background

As earlier stated, PCA seeks the *k*-dimensional projection that maximizes

$$\frac{1}{n}\sum_{i<j}\left\|P_L(\mathbf{x}_i)-P_L(\mathbf{x}_j)\right\|_2^2. \tag{3.1}$$

Let $\mathbf{x}_i$, $i=1,\ldots,n$ be data points where $\mathbf{x}_i \in \mathbb{R}^m$ and let the data points be arranged as the rows of a $n\times m$ matrix X such that the m coordinates is given by the columns of X. As earlier stated, the coordinates will be represented by Greek indices while the observations (data points) will be represented by Latin indices (i.e. $\mathbf{x}_{i\alpha}$ is the αth coordinate of the ith observation). For all computations, we assume that the data is centered which can be achieved by simple translation of the data.

Using the Euclidean distance, the problem 3.1 can be stated as

$$D_X = \sum_{i<j}\| P_L(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 \rightarrow max,$$

where $i, j = 1,...,n,$, $L = \sum_{\alpha=1}^{k} a_\alpha v_\alpha$, $a_\alpha \in \mathbb{R}^m$, $\alpha = 1,2,...,k$ and $k \le m$. Also $\mathbf{v}_\alpha \in \mathbb{R}^m$ and $\langle \mathbf{v}_\alpha, \mathbf{v}_\beta \rangle = \delta_{\alpha\beta}$ (δ is Kronecker delta). The projection of a vector $\mathbf{x}$ to a plane denoted by $P_L(\mathbf{x}) = \sum_{\alpha=1}^{k} \mathbf{v}_\alpha (\mathbf{v}_\alpha, \mathbf{x})$.

We have shown in definition 4 that this problem reduces to (1.19)

$$\max_{v_1,...,v_k} \sum_{\alpha=1}^{k} \mathbf{v}_\alpha^T \tilde{S} \mathbf{v}_\alpha, \quad \text{Subject to } (\mathbf{v}_\alpha, \mathbf{v}_\beta) = \delta_{\alpha\beta} \ \ \alpha, \beta = 1,2,..,k. \tag{3.2}$$

Where $\tilde{S}$ is a symmetric positive semi-definite matrix given as:

$$\tilde{S} = \sum_{i<j} \left[(\mathbf{x}_i - \mathbf{x}_j) \otimes (\mathbf{x}_i - \mathbf{x}_j) \right],$$

and each element of $\tilde{S}$ is given as

$$\tilde{S}_{\alpha\beta} = \sum_{i<j} \left[(\mathbf{x}_{i\alpha} - \mathbf{x}_{j\alpha})(\mathbf{x}_{i\beta} - \mathbf{x}_{j\beta}) \right]. \tag{3.3}$$

We also showed if $\lambda_1 \ge ... \ge \lambda_m$ is the sorted eigenvalues of the matrix $\tilde{S}$ and $\mathbf{e}_1,...,\mathbf{e}_m$ is the corresponding eigenvectors, from theorem 1.1, a maximizer of the constrained maximization problem (1.19) is $\mathbf{e}_1,...,\mathbf{e}_m$. Finally from lemma 1.1, we showed that $\tilde{S} = n^2 \operatorname{cov}(X)$ and since the eigenvectors of a matrix does not change when multiplied by a positive constant then $\mathbf{e}_1,...,\mathbf{e}_m$ is also the eigenvectors of the covariance matrix of the data $\operatorname{cov}(X)$.

If there are $q < m$ distinct eigenvalues $\lambda_1 \ge ... \ge \lambda_q$ of the matrix $\tilde{S}$ such that λ_i is of multiplicity n_i and $\sum_{i=1}^{q} n_i = m,$ we have a case called eigenvalue degeneracy. For each λ_i with multiplicity $n_i > 1,$ the eigenvectors lie in a n_i dimensional subspace orthogonal to the subspace spanned by the non-degenerate eigenvalues. For symmetric matrix, these n_i eigenvectors will be linearly independent and using Gram-Schmidt procedure we can find n_i orthogonal vectors that span this subspace.

Proposition 3.1

The matrix $\widetilde{S} = \sum_{i<j}\left[(\mathbf{x}_i - \mathbf{x}_j)\otimes(\mathbf{x}_i - \mathbf{x}_j)\right]$ *can be written as* $X^T LX$, *where* $L = [L_{ij}]$, $L_{ij} = [n\delta_{ij} - 1]$ *and* δ_{ij} *is the Kronecker delta.*

Proof:

From equation (1.18)) we have

$$\widetilde{S}_{\alpha\beta} = \sum_{i<j}\left[(\mathbf{x}_{i\alpha} - \mathbf{x}_{j\alpha})(\mathbf{x}_{i\beta} - \mathbf{x}_{j\beta})\right],$$

$$= \frac{1}{2}\left[\sum_{i=1}^{n} n\mathbf{x}_{i\alpha}\mathbf{x}_{i\beta} + \sum_{j=1}^{n} n\mathbf{x}_{j\alpha}\mathbf{x}_{j\beta} - \sum_{i=1}^{n}\mathbf{x}_{i\alpha}\sum_{j=1}^{n}\mathbf{x}_{j\beta} - \sum_{j=1}^{n}\mathbf{x}_{j\alpha}\sum_{i=1}^{n}\mathbf{x}_{i\beta}\right]$$

$$= \frac{1}{2}\left[2\sum_{i=1}^{n} n\mathbf{x}_{i\alpha}\mathbf{x}_{i\beta} - \sum_{i=1}^{n}\mathbf{x}_{i\alpha}\sum_{j=1}^{n}\mathbf{x}_{j\beta} - \sum_{j=1}^{n}\mathbf{x}_{j\alpha}\sum_{i=1}^{n}\mathbf{x}_{i\beta}\right]$$

$$= \sum_{i=1}^{n} n\mathbf{x}_{i\alpha}\mathbf{x}_{i\beta} - \sum_{i=1}^{n}\mathbf{x}_{i\alpha}\sum_{j=1}^{n}\mathbf{x}_{j\beta}.$$

$$= \sum_{i,j=1}^{n}(n\delta_{ij} - 1)\mathbf{x}_{i\alpha}\mathbf{x}_{j\beta} = \sum_{i,j=1}^{n} L_{ij}\mathbf{x}_{i\alpha}\mathbf{x}_{j\beta}\,. \tag{3.4}$$

In matrix notation, the quadratic form (3.4), can be written as

$$\widetilde{S}_{\alpha\beta} = (X_\alpha{}^T LX_\beta) \tag{3.5}$$

$$\widetilde{S} = (X^T LX) \tag{3.6}$$

Where $L = [L_{ij}]$, $L_{ij} = [n\delta_{ij} - 1]$ and δ_{ij} is the Kronecker delta. L is an $n \times n$ symmetric positive-semi definite matrix with zero column and row sum and this is useful for describing the pairwise relationship between data elements see lemma 3.1 and [71].

Lemma 3.1*: Let* L *be as defined above and let* $\mathbf{x} \in \mathbb{R}^n$ *then*

$$\mathbf{x}^T L\mathbf{x} = \sum_{i<j}^{n} - L_{ij}(\mathbf{x}_i - \mathbf{x}_j)^2.$$

And for k coordinate vectors we have:

$$\sum_{\alpha=1}^{k}\mathbf{x}_\alpha{}^T L\mathbf{x}_\alpha = \sum_{i<j}^{n} - L_{ij}(\sum_{\alpha=1}^{k}(\mathbf{x}_{i\alpha} - \mathbf{x}_{j\alpha})^2$$

$$= \sum_{i<j}^{n} - L_{ij} \parallel \mathbf{x}_i - \mathbf{x}_j \parallel_2^2.$$

Hence we see that quadratic form associated with the matrix L is the weighted sum of all pairwise squared distances. The matrix given by L is useful for describing the pairwise relationship between data elements.

3.2.1 Weighted PCA - Revisited

Definition 4 allows for some flexibility in the analysis of principal components. Since we have control over the pairwise distances of data projection, by assigning weights to these pairwise distances, we can manipulate the result of PCA of the data.

We now consider this generalization of PCA using weighted pairwise distances of data projection.

$$\sum_{i<j} w_{ij}[dist^2(P_L\mathbf{x}_i, P_L\mathbf{x}_j)].$$

Using the Euclidean distance function, the problem can be stated as:

$$D_X = \sum_{i<j} w_{ij} \| P_L(\mathbf{x}_i - \mathbf{x}_j) \|_2^2 \rightarrow \max$$
$$\text{Subject to: } (\mathbf{v}_\alpha, \mathbf{v}_\beta) = \delta_{\alpha\beta} \tag{3.7}$$

Where $w_{ij} = w_{ji}$ is the non-negative weight assigned to the distance between elements i and j and $w_{ij} = 0$, for $i = j$. The equation (3.7) reduces to maximizing the equation below:

$$D_X = \sum_{i<j} w_{ij} \left[\sum_{\alpha=1}^{k} (\mathbf{v}_\alpha, \mathbf{x}_i - \mathbf{x}_j)^2 \right] \tag{3.8}$$

This is the same as

$$D_X = \sum_{\alpha=1}^{k} \left[\sum_{i<j} w_{ij} (\mathbf{v}_\alpha, \mathbf{x}_i - \mathbf{x}_j)^2 \right]$$

The expression in the bracket given as

$$\sum_{i<j} w_{ij} (\mathbf{v}_\alpha, \mathbf{x}_i - \mathbf{x}_j)^2 = \sum_{i<j} w_{ij} (\mathbf{v}_\alpha, \mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j, \mathbf{v}_\alpha)$$
$$= \mathbf{v}_\alpha^T \tilde{M} \mathbf{v}_\alpha \tag{3.9}$$

Where $\tilde{M}$ is a symmetric positive semi-definite matrix given as

$$\tilde{M} = \sum_{i<j} w_{ij} \left[(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j) \right]. \tag{3.10}$$

Proposition 3.2: *The matrix* $\tilde{M} = \sum_{i<j} w_{ij}\left[(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)\right]$ *can be written as* $X^T L^w X$ *where* $L^w = [L^w{}_{ij}] = \left(\delta_{ij}(\sum_{j=1}^{n} w_{ij}) - w_{ij}\right)$ *and* δ_{ij} *is the Kronecker delta.*

Proof:

Let $R_i = \sum_j w_{ij}$ and let $C_j = \sum_i w_{ij}$, from equation (3.10) we have

$$\tilde{M} = \sum_{i<j} w_{ij}\left[(\mathbf{x}_i - \mathbf{x}_j)\otimes(\mathbf{x}_i - \mathbf{x}_j)\right] = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\left[(\mathbf{x}_i - \mathbf{x}_j)\otimes(\mathbf{x}_i - \mathbf{x}_j)\right] \tag{3.11}$$

$$= \frac{1}{2}\left[\sum_{i=1}^{n} R_i(\mathbf{x}_i \otimes \mathbf{x}_i) + \sum_{j=1}^{n} C_j(\mathbf{x}_j \otimes \mathbf{x}_j) - \sum_{i,j=1}^{n} w_{ij}[(\mathbf{x}_i \otimes \mathbf{x}_j) + (\mathbf{x}_j \otimes \mathbf{x}_i)]\right] \tag{3.12}$$

$$= \frac{1}{2}\left[\sum_{i=1}^{n} (R_i + C_i)(\mathbf{x}_i \otimes \mathbf{x}_i) - \sum_{i,j=1}^{n} w_{ij}[(\mathbf{x}_i \otimes \mathbf{x}_j) + (\mathbf{x}_j \otimes \mathbf{x}_i)]\right]. \tag{3.13}$$

Each element is given as

$$M_{\alpha\beta} = \frac{1}{2}\left[\sum_{i=1}^{n} (R_i + C_i)\mathbf{x}_{i\alpha}\mathbf{x}_{i\beta} - \sum_{i,j=1}^{n} w_{ij}(\mathbf{x}_{i\alpha}\mathbf{x}_{j\beta} + \mathbf{x}_{j\beta}\mathbf{x}_{i\alpha})\right], \tag{3.14}$$

$$M_{\alpha\beta} = \left[\sum_{i=1}^{n} R_i(\mathbf{x}_{i\alpha}\mathbf{x}_{i\beta}) - \sum_{i,j=1}^{n} w_{ij}(\mathbf{x}_{i\alpha}\mathbf{x}_{j\beta})\right], \tag{3.15}$$

because $R_i = C_i$ and $\sum_{i,j=1}^{n} w_{ij}(\mathbf{x}_{i\alpha}\mathbf{x}_{j\beta}) = \sum_{i,j=1}^{n} w_{ij}\mathbf{x}_{j\beta}\mathbf{x}_{i\alpha}$.

Therefore we can write equation (3.15) as

$$M_{\alpha\beta} = \left(\delta_{ij}R_i - w_{ij}\right)\mathbf{x}_{i\alpha}\mathbf{x}_{j\beta}$$

$$M_{\alpha\beta} = \left(\delta_{ij}(\sum_{j=1}^{n} w_{ij}) - w_{ij}\right)\mathbf{x}_{i\alpha}\mathbf{x}_{j\beta} \tag{3.16}$$

Let $L^w = [L^w{}_{ij}] = \left(\delta_{ij}(\sum_{j=1}^{n} w_{ij}) - w_{ij}\right)$. We can write L_{ij} in the form

$$L_{ij}{}^{w} = \begin{cases} \sum_{j=1}^{n} w_{ij} & i = j \\ -w_{ij} & i \neq j \end{cases}$$

Where $w_{ij} = 0$, for $i = j$.

In matrix notation, the quadratic form (3.16), can be written as

$$\tilde{M}_{\alpha\beta} = (X_\alpha{}^T L^w X_\beta), \qquad (3.17)$$

$$\tilde{M} = (X^T L^w X).$$

Therefore the problem given by (3.7) is reduced to

$$\max_{v_1,\ldots,v_k} \sum_{\alpha=1}^{k} \mathbf{v}_\alpha^T \tilde{M} \mathbf{v}_\alpha \qquad (3.18)$$

$$\text{Subject to } (\mathbf{v}_\alpha, \mathbf{v}_\beta) = \delta_{\alpha\beta} \;\; \alpha, \beta = 1,2,..,k.$$

Where $\tilde{M}$ is a symmetric positive semi-definite matrix, and from theorem 1.1, the eigenvectors corresponding to the sorted eigenvalues of the matrix $\tilde{M}$ is a maximizer of the constrained maximization problem 3.18. In the case of degenerated eigenvalues, the set $\mathbf{e}_1,\ldots,\mathbf{e}_m$ is not uniquely defined.

3.3 Multiscale PCA (MPCA)

In this section, we introduce the Multiscale PCA (MPCA) algorithm to enhance the robustness of PCA and to study the structures of data at various distances thereby revealing some hidden structure(s) in the data which conventional PCA might not reveal. MPCA compute principal components by maximizing the sum of pairwise distances between data projection for only pairs of datapoints for which the distance is within a chosen scales. This is achieved by assigning a weight of 1 to the pairwise distance of projections of any pair of data points with distance within the chosen scale and a weight of 0 otherwise.

$$\begin{cases} w_{ij} = 1 & l \le \| \mathbf{x}_i - \mathbf{x}_j \|_2 \le u \\ w_{ij} = 0 & otherwise, \end{cases} \qquad (3.19)$$

where l Is the lower limit of the scale and u is the upper limit. Let $d^{\min}$ be the minimum pairwise distance greater than zero and $d^{\max}$ be the maximum pairwise distance in the dataset, then we should choose l and u such that $l \in L = \{a : 0 \le a < d^{\max}\}$, $u \in U = \{b : d^{\min} \le b \le d^{\max}\}$ and $l < u$ where a, b are real numbers.

With control over the pairwise distances of data projection, we are able to study the structure of data at various localization of distances (scales). Analyzing the changes in the principal component decomposition of the data at different levels of localization

in scale can reveal interesting underlying structure(s) that may be present in data. For example, reducing the upper limit of the scale while keeping the lower limit at 0 translate to finding the subspace that best approximate the data while preserving the smaller distances structure of the dataset. In such scenario, since large distances (larger than the scale) have been assigned a weight of 0 (i.e. excluded from the analysis), there could be a distortion of the structures represented by such large distances. Given a dataset with outliers, this has the effect of minimizing without explicit exclusion the contributions of certain influential data elements in the analysis of the principal components.

3.3.1 The MPCA Algorithm

Given a dataset, the Multiscale PCA Algorithm is given below:

1. Centralize the data by subtracting the mean of the variables from each observation.
2. Find the dissimilarity matrix by computing the Euclidean distance.
3. Choose an appropriate scale between 0 and the maximum distance. For easy analysis, a scale between 0 and 1 could be chosen and then multiplied by the maximum distance. For this thesis when using scale between 0 and 1 we call it standard scale.
4. Calculated the binary weight as given in equation (3.19)
5. Calculate the matrix L^w as given below
6. $$L_{ij}{}^{w} = \begin{cases} \sum_{j=1}^{n} w_{ij} & i = j \\ -w_{ij} & i \neq j \end{cases}$$
7. Calculate the matrix $A = Y^T L^w Y$ where Y is the centralized data matrix.
8. Find the sorted eigenvalues of the matrix A in descending order of magnitude and project the data onto their corresponding eigenvectors. This will be the principal components at the selected scale.

Let us consider an example to illustrate this idea. Multiscale PCA on Data with repeated patterns.

Example 1

Let us consider the data sample with repeated underlying structure as shown in figure 11a-11e.

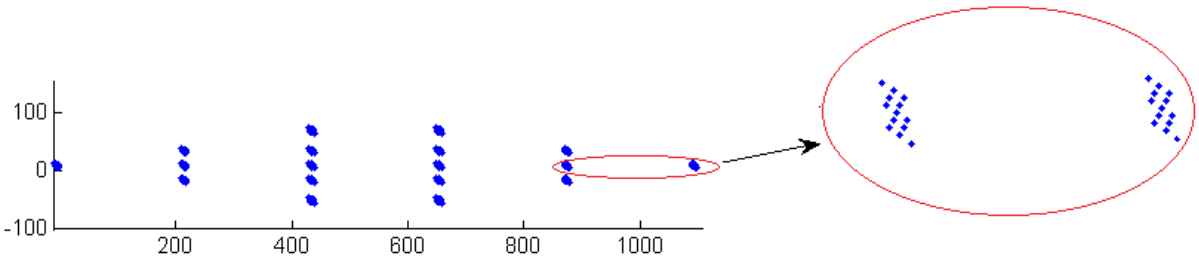


Figure 11a. Scatter plot of data with repeated pattern.

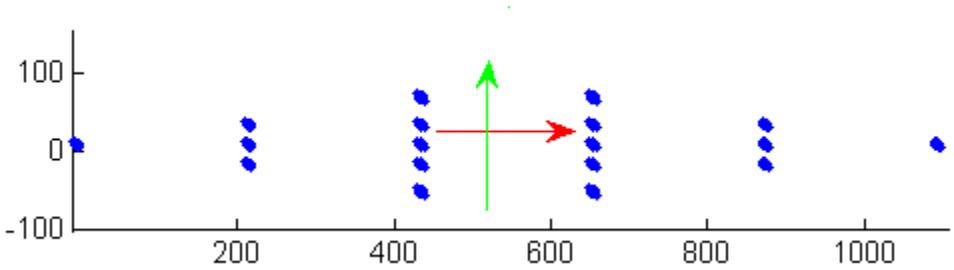


Figure 11b. The red and the green arrows indicate the direction of the first and second principal components respectively, at a scale of [0-1108] equivalent to standard scale [0-1]. This is the same as PCA using the sample covariance matrix.

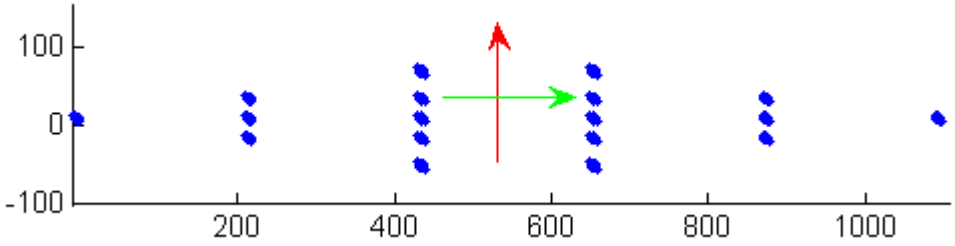


Figure 11c. The red and the green arrows indicate the direction of the first and second principal components respectively, at a scale of [0-200] equivalent to standard scale [0-.18].

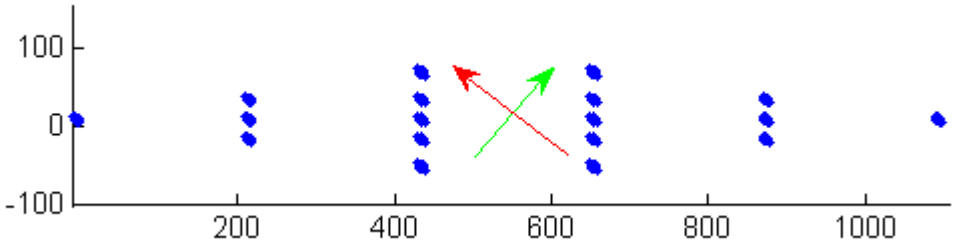


Figure 11d. The red and the green arrow indicate the direction of the first and second principal components respectively, at a scale of [0-12] equivalent to standard scale [0-0.01].

From figure 14 we observe that the PCA reveals the inner structure of the data. A better view of this inner structure and the PCA is given below in figure 10.

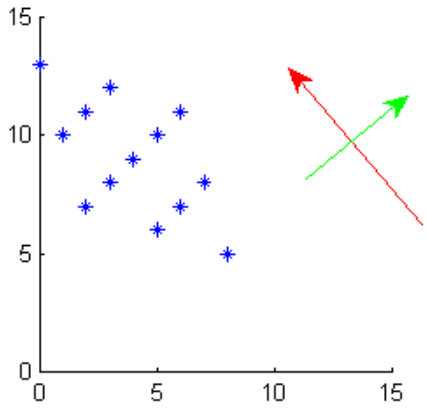


Figure 11e. Magnified view of one cluster in the dataset with the red and green arrow representing the direction of the first and second principal components respectively.

As observed from the figures above, the principal axes changed as the scale changed and this was able to reveal some underlying structures of the dataset. The changes in the first principal axis at various scales have been captured in figure 12. The plot is the angle between the first principal axis of PCA and the first principal axis of MPCA at various selected scales.

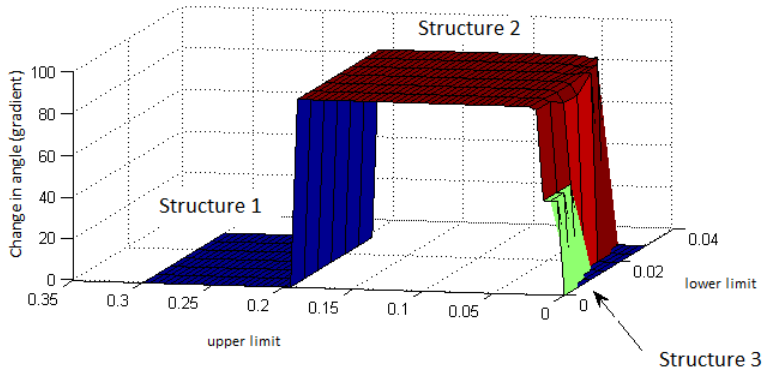


Figure 12. This diagram illustrate the changes in the angle of the principal components as the scale changed. The angles recorded here is the angle (in gradient) between the first principal axis of PCA and the first principal axis of MPCA at selected scale.

3.4 Representation of PCA Structures

In the previous section we have introduced the MPCA and its algorithm which is the analysis of the principal components of a dataset at various localizations of the pairwise distances (scale) of the data projection. Now we look at how to represent and analyse the loading vectors (corresponding to the principal axes) that are generated from multiscale principal component analysis of a given dataset.

Let us consider the interval of values $[l,u]$ where l = lower limit, u = upper limit and $l<u$. The scale (l,u) such that $l \in L = \{a : 0 \le a < d^{\max}\}$, $u \in U = \{b : d^{\min} \le b \le d^{\max}\}$ and $l<u$ (as defined earlier) can be represented as point in the plane $\mathbb{R}^2$ as shown in figure 13. The resulting principal component structures in MPCA depend on the points (l,u) on the plane. This implies that for every scale (l,u) there is an associated set of loading vectors which maximizes equation 3.18, where weights have been chosen using 3.19. This loading vectors represent the PCA structure at the scale (l,u).

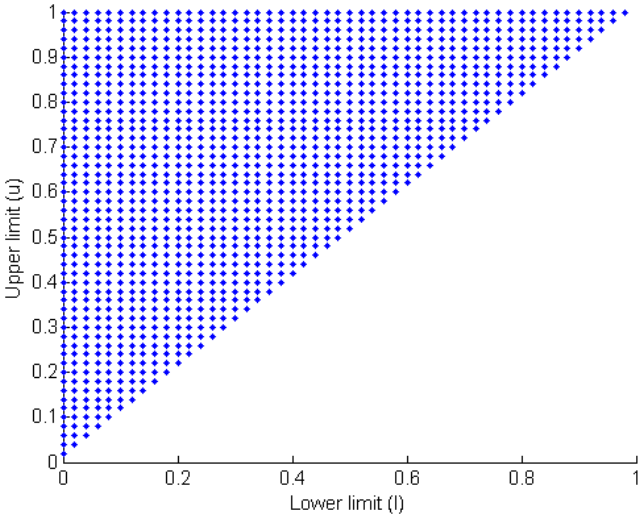


Figure 13. This diagram shows standard scale represented as points on the plane $\mathbf{R}^2$. The standard scale normalizes the pairwise distances of a dataset such that the maximum pairwise distance equal 1.

To explore the local structure(s) of a given dataset, we would like to analyze the PCA structure(s) of the dataset at various scales. For each point (l,u) on the scale, let us denote the associated set of loading vectors as $\mathbf{B}_{(l.u)} = \{b^1_{(l,u)}, b^2_{(l,u)} \ldots, b^m_{(l,u)}\}$ where the elements $b^k_{(l,u)}$ is the loading vector for the *k*-th principal component at the scale (l,u). An ordered orthonormal set of *k* vectors is called a *k*-frame, thus $\mathbf{B}_{(l.u)}$ is a *k*-frame since its element are the loading vectors which are orthonormal. Since the objective is to explore the local structure(s) of the dataset *X* at various scales using the MPCA structures, then there is a need to analyze the collection of $\mathbf{B}_{(l.u)}$ for all scales (l,u) at which MPCA has been performed on the dataset. From the definition of the scale, there is a continuum of points (l,u) in which the principal components can be computed, therefore in application there is a need to sample this space of scales.

The collection $\mathbf{B}_{(l.u)}$ is a subset of the Stiefel manifold as defined below. We will therefore consider some properties of the Stiefel manifold to decide if we can analyze $\mathbf{B}_{(l.u)}$ as set of points in the Stiefel manifold.

Definition: The Stiefel manifold denoted by $V_k(R^m)$ is the set of all orthonormal *k*-frame in R^m. If for each *k*-frame we arrange the vectors as the column of an $m \times k$ matrices then $V_k(\mathbb{R}^m) = \{A \in \mathbb{R}^{m \times k} : A^T A = I_k\}$.Where I_k is the identity matrix.

Now we consider some properties of the Stiefel manifold. In particular the case $k = 1$ is the set $V_1(\mathbb{R}^m) = \{\mathbf{v} \in \mathbb{R}^m : \mathbf{v}^T\mathbf{v} = 1\}$ which is a unit $m-1$ sphere. For example let us consider two elements $\mathbf{v}_1, \mathbf{v}_2 \in V_1(\mathbb{R}^m)$ where $\mathbf{v}_2 = -\mathbf{v}_1$ (the antipodal point on the unit sphere). These two vectors represent the same principal component; however the average of these two vectors is $\mathbf{0}$, which implies that the result of the average does not take into consideration the structure of the principal components. If we consider a scenario in which the first principal component of a dataset has been represented by normalized vector $\mathbf{v}$ corresponding to the first loading vector then from the example above, it implies that it averages to zero. Thus we see that in the case of equidistribution of a normalized vector $\mathbf{v}$ on $m-1$ sphere the expectation $E[\mathbf{v}] = \mathbf{0}$ due to spherical symmetry. The expectation is the vector in the sphere which is rotation invariant and that is $\mathbf{0}$. From the above, we conclude that the statistics of vector is not good and could be counter intuitive to represent the PCA structures of the dataset at various scales by the loading vectors. As the property of the Stiefel manifold does not capture this important property of the principal component, we seek another representation of the MPCA structures in order to further analyse the MPCA structures.

Since a vector $\mathbf{v} \in \mathbb{R}^m$ and its antipodal point $(-\mathbf{v})$ represent the same principal component, a more accurate representation of the loading vectors is as a set of orthogonal axial frames [6]. An Orthogonal axial frame is defined as the set of ordered orthogonal k-vectors which has the form $(\pm\mathbf{v}_1, \pm\mathbf{v}_2, \ldots, \pm\mathbf{v}_k)$, $k \leq m$. In fact it turns out that representing the first principal component by a vector is only convenient for calculations but strictly speaking the principal component is not a vector. Geometrically we see that the first principal component is a line rather than a vector and hence principal components are points in the space of straight lines called projective space. We briefly discuss this below.

3.4.1 Space of Lines and Linear Subspaces.

We know that given a vector space V of dimension $m+1$ over the real field $\mathbb{R}$ with the basis $U = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{m+1}\}$, then any point $\mathbf{v} \in V$ can be represented by the coordinate vector $[v_1, v_2, \ldots, v_{m+1}]$ relative to the basis U. This coordinate vector defines an isomorphism

between V and $\mathbb{R}^{m+1}$. A line L in the vector space V which is in the direction of a vector $\mathbf{v} \in V$ is the subspace spanned by the vector $\mathbf{v}$ and can be represented as $L = \{\lambda \mathbf{v},\ \lambda \in \mathbb{R} \setminus \{\mathbf{0}\}\}$.

The projective space $P^m(\mathbb{R})$ over a field $\mathbb{R}$ is the set of lines in a vector space V of dimension $m+1$.

Definition: More formally we define the real projective space $P^m(\mathbb{R})$ as a set of lines through the origin in $\mathbb{R}^{m+1}$. If $\mathbf{x} \in \mathbb{R}^{m+1}$, $\mathbf{x} = (x_0, x_1, \ldots, x_m)$ one can write

$$P^m(\mathbb{R}) = (\mathbb{R}^{m+1} \setminus \{\mathbf{0}\})/\sim \tag{3.20}$$

Where '$\sim$' is the equivalence relation defined by: $(x_0, \ldots x_m) \sim (v_0, \ldots, v_m)$ if there is a non-zero real number λ (i.e. $\lambda \in \mathbb{R},\ \lambda \neq 0$) such that $(x_0, \ldots, x_m) = (\lambda v_0, \ldots, \lambda v_m)$.

Thus any point $\mathbf{p} \in P^m(\mathbb{R})$ is a line L in the vector space V and $\mathbf{p}$ can be represented by the coordinate $[v_1, v_2, \ldots, v_{m+1}]$ of any non-zero point $\lambda \mathbf{v}$ on the line L ($\lambda \in \mathbb{R},\ \lambda \neq 0$). These are called homogenous coordinates of the projection point.

Slightly more generally, the definition *of real projective space* could be extended for a vector space V (over some field K, or even more generally a module V over some division ring). The coordinates $(x_0, \ldots, x_m)$ defined up to multiplication by $\lambda \neq 0$ are called the homogenous coordinates of $P^m(\mathbb{R})$. If $m = 1$, it is called projective line and for $m = 2$, it is called projective plane.

As mentioned earlier, geometrically the principal component is a line rather than a vector due to the fact that a loading vector $\mathbf{v}$ and its antipodal point $-\mathbf{v}$ represent the same principal component. Thus since the first principal component (and by extension each principal component) is a line spanned by the loading vector representing it, we conclude that each principal component is a point in the projective space.

Let us extend our discussion to multiple principal components (i.e. considering the first k- principal components of a dataset together). Just like the k-th principal component in the data space is the line spanned by the k-th loading vector, the first k-principal components considered together is a subspace of the data space spanned by

the first *k*-loading vectors. This leads us to consider the space of linear subspaces of a vector space next.

Given a vector space *V* of dimension *m*, the collection of all *k*-dimensional linear subspaces of the vector space *V* is called the Grassmannian space denoted by $G(k,V)$. A point $g \in G(k,V)$ correspond to a subspace in the vector space *V,* thus the Grassmannian is the space that parameterizes all linear subspaces of dimension *k* of a vector space *V*.

Definition: The Grassmannian space is the set of *k*-dimensional subspaces of *m*-dimensional space $\mathbb{R}^m$ going through the origin. The Grassmannian can also be denoted by $G(k,m)$ where *m* is the dimension of the vector *V.*

We observe that $G(1,m) = P^1(\mathbb{R})$ which is the projective line as earlier discussed. Hence the Grassmannian space is a generalization of the projective space $G(k,V)$ where the vector space is the data space.

The first *k* principal components considered together is a subspace of the data space spanned by the first *k*-loading vectors and this can be considered as a point in the Grassmannian space where the vector space *V* is the data space. By extension, any *k*-principal components considered together can be considered as a point in the Grassmannian space. The projective line and plane can be seen as a particular case of the Grassmannian where *k* =1 and *k* =2 respectively. Therefore for the purposes of representation and analyses, we shall consider the principal components in general (i.e of dimension *k, k*=1, 2…, *m*) as points in the Grassmannian space $G(k,V)$.

Recall that our objective is to represent and analyse the principal component structures resulting from the application of multiscale principal component analysis on a dataset *X.* This problem can be viewed as representing and analysing points in the Grassmannian space $G(k,V)$. We shall consider the statistics of the multiscale principal components as the statistics of points in the Grassmannian space.

One way to study the points in the Grassmannian space is to embed it in a suitable vector space and then analyse in this vector space; this is called the embedding approach [6] [81]. In considering a vector space to embed the Grassmannian, We note

that the projection operator has a one-to-one correspondence with the projective space and by extension in higher dimension; the projection operator has a one-to-one correspondence with the Grassmannian space. The projector operator (Projection matrix) is a linear transformation that maps a vector space *V* with dimension *m* to a subspace *W* of dimension $k,\ k \leq m$.

3.4.2 Projection Matrix

Given a subspace *W* of dimension *k* belonging to a vector space of dimension *m,* there is a unique operator B of orthogonal projection onto *W* [114]. This operator B (or its matrix representation) satisfies the following three conditions

- B is idempotent: $B^2 = B$.
- B is symmetric: $\langle Bu, v\rangle = \langle u, Bv\rangle$ for $u, v \in \mathbb{R}^m$. In order words the matrix representation satsifies $B^T = B$
- B has trace *k*: *trace*(B) = *k*.

Also, let $M\{m, \mathbb{R}\}$ denote the space of real $m \times m$ matrices (matrix space).The set of matrices $A(k, m) \subset M(m, \mathbb{R})$ defined by $B \in A(k, m)$ if and only if B satisfy the three condition given above is the operator of orthogonal projection to some *k*-dimensional subspace in $\mathbb{R}^m$.

Let $\mathbf{Gr}(k, \mathbb{R}^m)$ denote the Grassmannian of *k*-dimensional subspaces of $\mathbb{R}^m$ and let $A(k, m)$ be the set of operator of orthogonal projections as defined above, then $\mathbf{Gr}(k, \mathbb{R}^m)$ and $A(k, m)$ are homeomorphic, with the homeomorphism defined by the $\Phi: \mathbf{Gr}(k, \mathbb{R}^m) \rightarrow A(k, m)$. From the stand point of topology, homeomorphic spaces are essentially identical, sharing the same topological properties like connectedness, compactness and variable separation axioms. Therefore we can embed the Grassmannian manifold $\Phi: \mathbf{Gr}(k, \mathbb{R}^m)$ into the space $A(k, m)$ of the orthogonal projection for further analysis.

Since the principal components can be viewed as points in the Grassmannian of *k*-dimensional subspaces of $\mathbb{R}^m$ and since there exist and homoemorphism beween

$\mathbf{Gr}(k, \mathbb{R}^m)$ and the projection matrix $A(k, m)$, we therefore propose to represent the MPCA structures of a dataset as the sequence of projection matrices $P_k \in A(k,m)$ corresponding to the sequences of subspace $W_k \in \mathbf{Gr}(k, \mathbb{R}^m)$, $1 \le k \le m$ spanned by the loading vectors of the principal components at the scale (l,u). We should mention that statistics is more convenient in this space.

3.4.3 Properties of Projection Matrix Representation

Having proposed representing the MPCA structures of a dataset by the orthogonal projection matrix corresponding to the subspace, we will like to examine some properties of the orthogonal projection matrix representation of the PCA structure. This is to ensure that this representation have properties that are consistent with the properties of principal component and also desirable for analysis.

The matrix representation P_k of the k-th principal component which is the the orthogonal projection matrix to the k-th principal component subspace is the tensor product of the loading vectors $\mathbf{e}_i$. $P_i = \mathbf{e}_i \otimes \mathbf{e}_i$, This product is bilinear and we can confirm that

$$-\mathbf{e}_i \otimes -\mathbf{e}_i = \mathbf{e}_i \otimes \mathbf{e}_i . \tag{3.21}$$

If we recall that the principal component given by $\mathbf{e}_i$ is the same as the antipodal point $-\mathbf{e}_i$ (orthogonal axial frames), then we confirm from (3.21) that this property of the representation is consistent with the property of principal components.

$P_i X = \mathbf{e}_i(\mathbf{e}_i, X) = (\mathbf{e}_i \otimes \mathbf{e}_i)X$ is the projection of data X to vectors $\mathbf{e}_i$ and

$\sum_{i=1}^{k} P_i X = \sum_{i=1}^{k} \mathbf{e}_i(\mathbf{e}_i, X)$ is the data X projected onto the first k - principal components.

For any m orthonormal vectors $\mathbf{e}_1,\ldots,\mathbf{e}_m$, $\sum_{i=1}^{m} \mathbf{e}_i \otimes \mathbf{e}_i = 1$. If $\mathbf{e}$ is one of $\mathbf{e}_i$ with probability $\frac{1}{m}$,

then $E(\mathbf{e} \otimes \mathbf{e}) = \frac{1}{m}$. The rotation invariance gives the same result if $\mathbf{e}$ is equidistributed

on unit $m-1$ sphere.

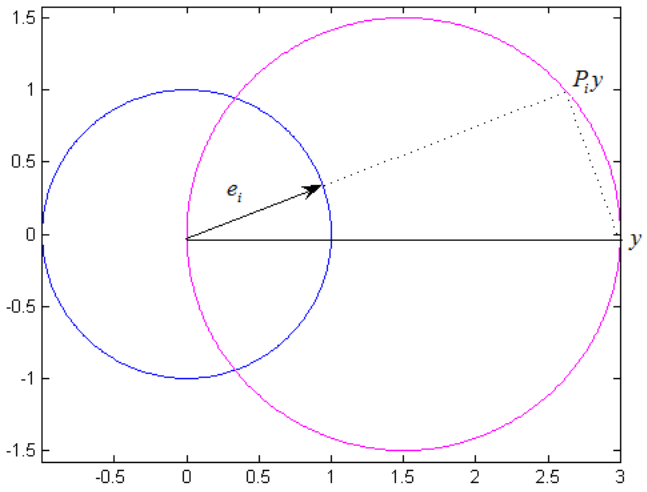


Figure 14. This diagram illustrates how the projection of vector *y* changes (given by the red sphere) as e_i moves along the blue solid 1-sphere.

It follows that the average projection of $E[\sum_{i=1}^{k} P_i X] = \frac{k}{m} X$.

The properties of the projection matrix considered capture the desired properties of PCA Structure and therefore we shall consider the statistics of PCA frames as the statistics of cortege of projection matrices. This implies that we shall represent the PCA structure of the data at any localization given by the scale (l,u) by the cortege of projection matrix corresponding to MPCA at that point.

There are two ways to represent the MPCA structures using the cortege of projection matrix. The first approach will be to represent each principal component given by $\mathbf{e}_i, i = 1,2,...,m$ by the rank one projector matrix $P_i = \mathbf{e}_i \otimes \mathbf{e}_i$. Using this representation, the full description of the PCA frame of a dataset *X* at a given scale (l,u) will be given by the cortege of projectors $P_1, P_2, ..., P_m$. Where the projection matrix P_i projects *X* to the 1-dimensional subspace spanned by $\mathbf{e}_i$.

The second approach is to represent the PCA frame using the cortege of projectors

$\rho_k = \sum_{i=1}^{k} \mathbf{e}_i \otimes \mathbf{e}_i$, $k = 1,2,\ldots,m-1$. Where ρ_k is a matrix of dimension *k*. The full description of the PCA frame of a dataset *X* at a given scale (l,u) will be given by the cortege of projectors $\rho_1, \rho_2, \ldots, \rho_{m-1}, \rho_m$. Using this representation, the matrix ρ_k projects the dataset *X* to the *k*-dimensional subspace spanned by the vectors $\mathbf{e}_i$, $i = 1,2,\ldots,k$.

Now If we arrange the $\mathbf{e}_i$, $i = 1,2,\ldots,k$ as columns of matrix E, then

$$\rho_k X = EE^T X \text{ and } \rho_k = EE^T . \tag{3.22}$$

In particular when $k = m$, $\rho_m = \mathbf{I}$ the identity matrix.

In this thesis we shall adopt the second approach based on the following motivation:

1) The first approach is not able to handle the situation where two or more principal components can interchange; a situation which arises in real application. Such situation can arise when there is eigenvalue degeneracy of the matrix $\tilde{M} = (X^T L^w X)$. As mentioned earlier, eigenvalue degeneracy is when there are $q < m$ distinct eigenvalues $\lambda_1 \geq \ldots \geq \lambda_q$ of the matrix $\tilde{M}$ such that λ_i is of multiplicity n_i and $\sum_{i=1}^{q} n_i = m$. For each λ_i with multiplicity $n_i > 1$, the eigenvectors lie in a n_i dimensional subspace orthogonal to the subspace spanned by the non-degenerate eigenvalues. The implication of this is that such eigenvectors are not stable since there is an additional freedom of rotation. This freedom of rotation could cause the loading vectors of the principal components to change thereby choosing any orthonormal basis of the eigenspace. In case of an interchange of principal components, since P_i are ordered then the representation using the first approach sees this as different object in the space of matrices. However since the second approach is the sum of the projection matrix $\sum_{i=1}^{k} P_k$ then the representation in such cases of interchange are the same.

2) Another motivation for choosing the second approach is that calculation and analysis is more tractable in the second approach. For example, analyzing PCA structure of *X* in a *k*-dimension subspace require one matrix ($\rho_k = \sum_{i=1}^{k} P_k$) which is the projection

matrix to the subspace, whereas in the first approach one require k-matrices ($P_i, i=1,\ldots,k$) with each P_i being the projection matrix to each basis in the subspace.

MPCA leads to scale dependent PCA structures and with the MPCA structures represented as the cortege of the projection matrix as proposed, we can study the structures in our data further by analyzing these projection matrices and considering various sub manifolds of the matrix space of orthogonal projection which represent the PCA structures at the various scales. For example, even though the full representation of a PCA structure of a dataset X at a given scale (l,u) is given by the cortege of projection matrices as mentioned earlier, we can decide for a case study to consider the k-dimensional sub manifolds and discarding the minor subspaces with small eigenvalue. This translates to studying ρ_k which could be analyzed over all scales (l,u). Another case study could be to analyse the k-dimension sub manifolds with smallest eigenvalues which translate to studying $\rho_{k*} = \mathbf{I} - \rho_k$. To understand the structure of the dataset X, we can study various sub manifolds of the space of projection matrix.

However it should be noted that while in PCA, the order of variance contribution of the principal components of a dataset X is based on the order of the corresponding eigenvalues, this is not the case for MPCA. Rather the eigenvalues in MPCA gives the order of minimal distortion of the dataset under the restriction of localization given by the scale (l,u) placed on the pairwise distances.

3.5 Analysis of PCA Structures - Clustering of Scales.

Each MPCA scale *(l,u)* defines a localization of PCA on a given dataset X. Our interest is to analyze the PCA structures at various scales for the dataset X with the intention of understanding the local structure(s) of the data and to reveal some hidden geometric structures of the data. This is a problem in unsupervised learning, in which we will like to identify patterns in the distribution of scales. This means that we will like to identify scales that have similar PCA structures which we can further cluster together leading to the notion of clustering of scales (see figure 14). To identify similar (dissimilar) scales, there is a need to define some kind of similarity function $s: [l,u] \times [l,u] \to \mathbb{R}$, which measures the similarity between two scales. In our case we want to measure similarity

between two PCA structures which are associated with two points $a, b \in [l, u]$ in the interval of scale.

We shall define the similarity function $s: [l, u] \times [l, u] \to \mathbb{R}$ of two scales as the distance between their corresponding PCA representations. This means that the measure of similarity between two scales is taken to be the distance between the orthogonal matrices of projection ρ_k corresponding to the PCA structures of the scales.

Clustering analysis of scales group together scales with similar PCA structures. The clusters correspond to some structures in the data. One way to describe each structure given by a cluster is to use the projection matrix corresponding to the medoid point of the cluster. The medoid point is proposed since the mean point cannot be guaranteed to belong to the space of orthogonal projection matrix and for finite set the medoid point can be seen as a point in the set which minimizes the Fréchet means as given in equation 1.39. However in section 3.8, we will introduce Ratio of Distortion which is another criterion that can be used to select cluster representative that describes the cluster.

For example, clustering analyses of scales for ρ_2 corresponds to cluster analysis of the PCA structures when the dataset is projected onto the loading vectors of the first 2-principal components at various scales. Another example can be to study the clusters for all scales for $1 - \rho_k = \sum_{i=k}^{m} \mathbf{e}_i \otimes \mathbf{e}_i$ which corresponds to the PCA structures of the data when projected to the *m-k* principal components with smallest eigenvalues.

Now let each point (l,u) in the interval of scale [*l*,*u*] be represented by χ_p, where $\chi_p = (l,u)$, $l \in L$, $u \in U$ such that $l < u$. We denote the projection matrix ρ_k at a point χ_p by ρ_{χ_p}. For any pair of points χ_p, χ_q in the space of scales we can compute the distance between the associated projectors $\rho_{\chi_p}, \rho_{\chi_q}$ for a given k using invariant norm. We recall that the Frobenius norm of a real matrix B denoted by $\| B_F \| = \sqrt{trace\{B^T B\}}$, therefore distance between projectors of any pair of points in the space of scale $dist(\rho_{\chi_p}, \rho_{\chi_q}) = \sqrt{trace\{(\rho_{\chi_p} - \rho_{\chi_q})^T (\rho_{\chi_p} - \rho_{\chi_q})\}}$.

Any standard clustering algorithm can be used to cluster the scale in order to reveal the PCA structures in the data. In this thesis, agglomerative hierarchical clustering was used. Deciding on the number of true clusters in clustering analysis is a classical

problem and one may want to compare various indices. A typical example of such is the $pseudo\,t^2$ statistic.

$$pseudo\,t^2 = \frac{[SSE_t - (SSE_a + SSE_b)](n_a + n_b - 2)}{SSE_a + SSE_b}.$$

Where SSE_a is the sum of square of cluster a, SSE_b is the sum of square of cluster b, SSE_t is the sum of square of cluster formed by joining clusters a and b, n_a and n_b are the number of elements in clusters a and b respectively. If a small value of the $pseudo\ t^2$ statistic at a step i of the hierarchical clustering is followed by a distinct large value at the step $i+1$, the cluster form at the step i is chosen as the optimal cluster. It is assume that the mean vector of the two clusters being merged at the step $i+1$ can be regarded as different and should probably not be merged.

3.6 Choice of Metric in the Space of Data

From (3.7) we recall that MPCA solves the optimization giving below

$$D_X = \sum_{i<j} w_{ij} \| P_L(\mathbf{x}_i - \mathbf{x}_j) \|_2^2 \to \max,$$

where the weights w_{ij} are chosen as

$$\begin{cases} w_{ij} = 1 & l \le \| x_i - x_j \|_2^2 \le u \\ w_{ij} = 0 & otherwise, \end{cases}$$

with points (l,u) chosen from the interval of scales $[l,u]$ as defined in section 3.3. These weights were chosen such that only pairwise distances within certain range of values (given by the scale) were preserved in the computation of PCA. In this case the distance function between pairs of points was chosen as Euclidean distance.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_2$$

However we can use other distance functions. The flexibility in the choice of distance functions or weights in general allows us to incorporate priori knowledge of similarity into the analysis of the principal components. For example, given a dataset *X*, if certain pairs of points in this dataset share some interesting characteristic which may or may not be measured on a continuous scale, we may wish to find an optimal

subspace such that the projection of the dataset to this subspace preserves this characteristic. In this situation we find the weighted PCA using the priori knowledge to decide the weight to be assigned to each pair of distances.

Often in application of PCA to real life data, variables of the dataset often have different unit of measurement and it becomes desirable to normalize the variable to unit variance for the purposes of comparison before performing PCA. Using the Euclidean distance as a measure of pairwise distances of points, then we can view the PCA on normalized data as weighted PCA on the un-normalized data where the distance metric between two datapoints is chosen as:

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \| \mathbf{x}_i^* - \mathbf{x}_j^* \|_2^2 \tag{3.23}$$

where $\mathbf{x}_i^* = \left[\frac{x_{i1}}{\hat{\sigma}_1}, \frac{x_{i2}}{\hat{\sigma}_2}, \ldots, \frac{x_{im}}{\hat{\sigma}_m} \right]$ is the vector $\mathbf{x}_i$ normalized to unit variance, and $\hat{\sigma}_i$ is the sample variance of variable *i*. and the weight chosen as:

$$\begin{cases} w_{ij} = 1 & l \le \| \mathbf{x}_i^* - \mathbf{x}_j^* \|^2 \le u \\ w_{ij} = 0 & otherwise. \end{cases} \tag{3.24}$$

This is the same as computing PCA using the correlation matrix of the dataset *X* (rather than the covariance matrix). Other normalization of the dataset can be used; an example of such is to normalize to unit mean.

The choice of normalization should be problem dependent and carefully chosen. However, we should point out that the topology of dataset is not invariant to scaling, so also the result of the MPCA of a dataset is not invariant to normalization especially when the choice of pairwise distance function is the Euclidean distance measure. A good example to illustrate this is to note that PCA result covariance matrix is different from PCA result using correlation matrix (where correlation matrix results from normalizing the variables in the dataset to unit variance).

Example 2

Let us consider the result of the cluster analysis of the data in example 1 for ρ_1 (i.e. projection onto first principal component). For illustration purpose, points from the subset of L and U have been selected. $l \in \{[0,0.4],[0.1,0.95]\}$ and $u \in \{[0.005,0.01],[0.11,0.19],[0.2,1]\}$. We visualize this cluster in figure 15 below.

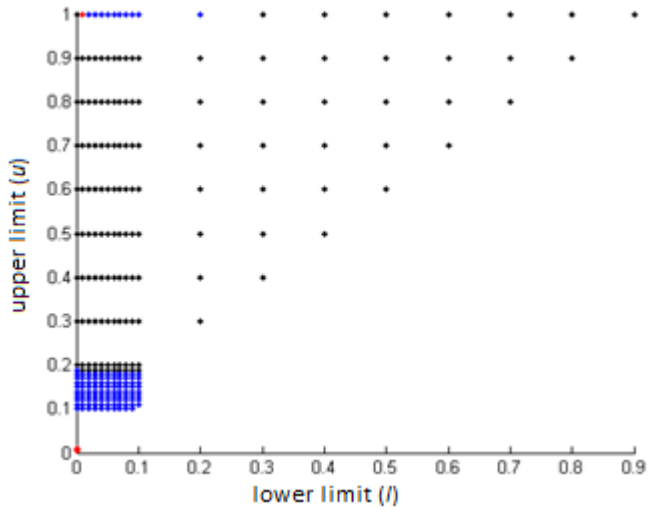


Figure 15. This diagram shows cluster of scales on the plane. Scales belonging to the same cluster are represented by the same colour.

The *pseudo* t^2 statistic indicates three meaningful clusters. This reaffirms the result displayed in figure 12.

We represent each of these structures by the eigenvector of the medoid point of the cluster representing it. The result is given in the table below.

Table 5. This table shows the description of each cluster. Each cluster has been described by the eigenvector of the projection matrix corresponding to the medoid point of the cluster.

Cluster	***Colour***	***Interval corresponding to Medoid point (Projector)***	***Eigenvector***
1	*Blue*	(0.1,0.19)	$\mathbf{e}_1 = [-0.0019, 1.0000]$
2	*Red*	(0,0.01)	$\mathbf{e}_1 = [-0.7071, 0.7071]$
3	*Black*	(0.3,0.8)	$\mathbf{e}_1 = [-1.0000, 0.0000]$

3.7 Examples

Example 3 - Multiscale PCA on Data with Outliers

The presence of outliers in our data serves to obfuscate the underlying structure of the data in PCA. MPCA can be used to reveal the underlying structure of data with outliers. By reducing the upper limit of the scale appropriately, we can effectively mitigate the effect of outliers in the analysis of the principal components without explicit exclusion of these outliers. To test the performance of MPCA on data with outliers, data were simulated along known plane and some outliers were added to this data. This data was embedded into a higher dimensional space and we seek to recover the original plane from the data by using PCA and MPCA (at various scales).

We consider a 3-dimensional data sample in which the elements are distributed uniformly on a plane (2-d) with the directional vectors given as;

$$\mathbf{u} = [0.8944,\ -0.4472,\ 0.0000]$$

$$\mathbf{v} = [0.1826,\ 0.3651,\ -0.9129];$$

with few outliers as can be seen in figure 16a. The result of the projection of the data to the first 2 principal components is shown in figure 16b. Figure 16c shows the result of MPCA at a scale of (0.0 - 0.8). This is found to have captured the data quite well.

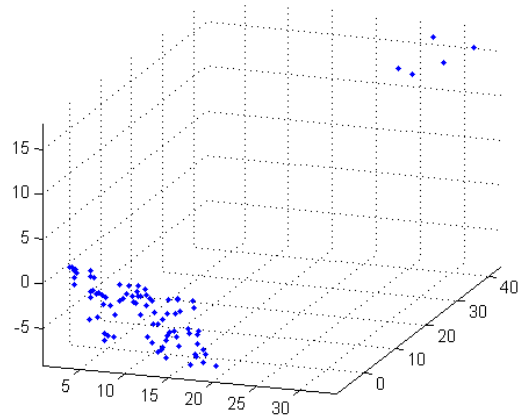


Figure 16a. Scatter plot of data in 3-dimension with a few outlying points.

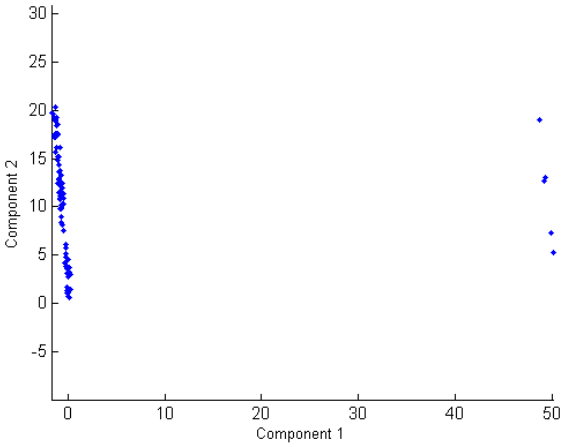


Figure 16b. The first 2 principal components using PCA. It can be observed that the outliers have influenced the result of the PCA.

See table D in the appendix for the difference in angle between the original directional vector and the loading vector of the first principal component of the Multiscale PCA at various scales. Clustering analysis suggest 2 cluster which we visualize in figure 16d.

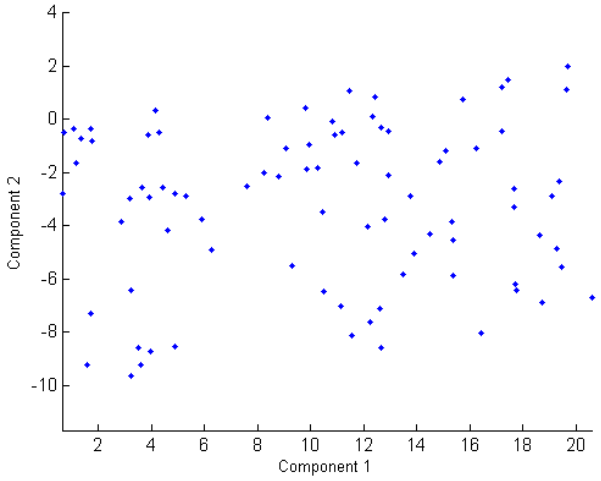


Figure 16c. Scatter plot of the first 2 principal components computed from MPCA at standard scale of (0, 0.8). The effects of the outliers have been mitigated and another structure of our data is seen here.

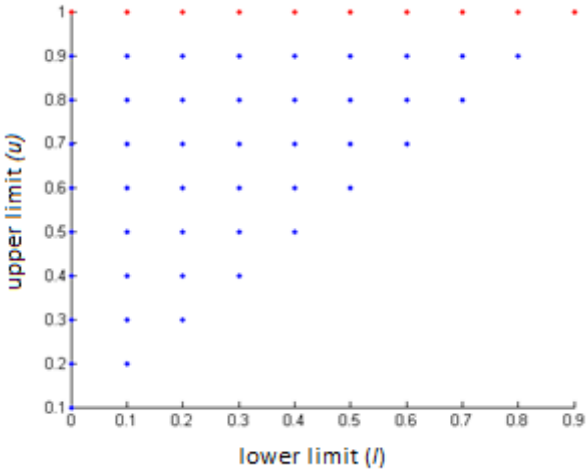


Figure 16d. This diagram shows the scales as points on the plane. Scales belonging to the same cluster are represented by the same colour.

The structure revealed by PCA has been influenced by the outliers, obfuscating the intrinsic structure revealed by the MPCA at scales (l,u), $0 \leq l \leq 0.8$, $0 < u \leq 0.9$, $l < u$. See figure 16b and 16c. MPCA given by cluster 1 (see blue cluster in figure 16d) mitigates the effect of the outliers and revealed a more interesting structure of the dataset as seen in figure 16d. The result of the cluster analysis is consistent with the difference in angle between the original plane and the first principal component computed using MPCA as given in table D in the appendix.

We represent each of these structures by the eigenvector of the medoid point of the cluster representing it. The result is given in table 6 below.

The table below is the result of the cluster analysis for ρ_2 .

Table 6. This table shows the description of each cluster. Each cluster has been described by the eigenvectors of the projector corresponding to the medoid point of the cluster.

Cluster	*Colour*	*Interval corresponding to Medoid point (Projector)*	*Eigenvector*
1	*Blue*	(0,0.1)	$\mathbf{e}_1 = [-0.9113, 0.3382, 0.2349]$ $\mathbf{e}_2 = [0.0538, -0.4679, 0.8821]$
2	*Red*	(0.4,1)	$\mathbf{e}_1 = [-0.3561, -0.8579, -0.3704]$ $\mathbf{e}_2 = [-0.8613, 0.4551, -0.2261]$

3.8 Ratio of Distortion

We have proposed describing each cluster of scales by the medoid projection matrix where distance between two scales is measured by the distance between their projection matrix. However, since one of the objectives of PCA is to approximate the dataset using a subspace of the dataspace, it becomes desirable to represent and describe each cluster by the projection matrix that 'best' approximate the data in some sense. Representing a cluster by the medoid projection matrix do not take into account how well the subspace represented by the medoid point approximate the dataset because the medoid is just the projection matrix with minimal average distance to all other projection matrices in the set under consideration.

Therefore we need a criterion for evaluating the projection matrix in a cluster in order to decide on which projection matrix is to be selected to represent and describe the cluster. If we recall that finding the principal components using definition 4 is equivalent to minimization of *mean squared distance distortion:*

$$\sum_{i,j=1}^{N} [dist^2(\mathbf{x}_i, \mathbf{x}_j) - dist^2(P_L\mathbf{x}_i, P_L\mathbf{x}_j)] \longrightarrow min$$

where the dimension k of L is strictly less than the dimension of the data. We define the ratio of distortion as

$$R_\rho = \frac{\sum_{i,j=1}^{N} \| (P_L \mathbf{x}_i) - (P_L \mathbf{x}_j) \|^2}{\sum_{i,j=1}^{N} \| (\mathbf{x}_i - \mathbf{x}_j) \|^2} \tag{3.25}$$

For all $\mathbf{x}_i, \mathbf{x}_j$ such that $l \leq \| \mathbf{x}_i - \mathbf{x}_j \|^2 \leq u$ where l is the lower limit of the scale and u is the upper limit.

Therefore we propose that that the criterion for choosing a representative projection matrix for a cluster should be the projection matrix with maximum ratio of distortion. If there are multiple projection matrices with maximum ratio of distortion, then one should be chosen at randomly. This criterion ensure that the matrix chosen to represent a cluster correspond to the subspace approximation with minimum distortion.

We remark that due to the restriction on pairwise distances which limit the pairwise distances included in the computation of R_ρ to those which fall within the interval represented by the scale (l,u), some pairwise distances are exempted. If the proportion of the exempted pairwise distances is large then MPCA could overfit the data and so also the ratio of distortion could be misleading. Therefore there is a need to put into consideration the number of pairwise distances exempted when deciding on ratio of distortion.

3.9 Discussion and Conclusion

3.9.1 Discussion

Reducing the upper limit to a very small number may cause MPCA to fit noise while increasing the lower limit only may cause MPCA to fit outliers if such is present in the data. It is important to note that by using MPCA, some pairwise distances are exempted in the analysis of principal component and the percentage of such exempted pairwise distance should be kept to a reasonable number. In order words, choosing too small a scale may result in low numbers of pairwise distances considered which may lead to loss of "information".

In example 3, we observed (see appendix E and F) that as the lower limit increased, the ratio of distortion appear to improve (even though the difference in angle is quite

large for some scales) but only because MPCA is fitting outliers. Therefore, in addition to the result of the ratio of distortion, the percentage of total pairwise distances exempted in the computation of the MPCA at different scales (especially when $l > 0$) should be considered in choosing an appropriate scale. A good scale for MPCA should be one with maximum ratio of distortion and least number of exempted pairwise distances.

Appendix G shows the percentage of pairwise distances of data points exempted in the computation of MPCA at various scales. It can be concluded that while reducing the upper limit is good for this data, increasing the lower limit while keeping the upper limit at 1 makes MPCA to fit outliers.

3.9.2 Conclusion

The result of MPCA of a dataset is a set of subspaces approximating the datasets depending on the scale chosen. For data with multiscale structures, clustering analysis of the MPCA structures at various scales reveal some interesting structures in our data which conventional PCA may not reveal due to the fact that such structures are obfuscated by other structures of higher variance. Each cluster corresponds to a structure and each structure is described by the medoid point of the cluster. Ratio of Distortion was also introduced. This can be used to select a scale for approximating a dataset and especially as a criterion for choosing the PCA structure to describe each cluster in the clustering analysis of scales. MPCA was tested on various data and for data with multi-scale structures, the method was able to reveal some underlying structure in data and also mitigate the influence of outliers on the analysis of principal component without having to exclude such outliers explicitly.

Chapter 4

PCA and Localization in Space

4.1 Introduction

A dataset X for which different structure exist at various localizations of the dataset will be referred to as a multi-structure dataset. Our objective in this thesis is to reveal some of these structures by studying the PCA structure(s) of the dataset at various localizations. However, localization of a dataset can be interpreted in many senses. In chapter 3, we considered localization in scale which was called multiscale principal component analysis (MPCA). In MPCA, we introduced localization by introducing weight to the pairwise distances of data projection and we then cluster the PCA structures of the dataset at various scales. We demonstrated using examples that for datasets with clear multiscale structure, MPCA helps reveal some structures in the datasets. However another interpretation of localization of PCA has been in the dataspace as discussed in section *2.3.5*, where the dataset is partitioned in the data space before performing PCA for example [14][29][24].

Usually in PCA, directions with low variance are regarded as noise and therefore considered distracting. When this assumption is correct, excluding such directions is useful in revealing the intrinsic structure in a dataset. However in a big dataset, it is expected that different regions of the data space may have different structures. In a multi-structure dataset, it may even be that the dataset have certain regions that have similar intrinsic structures or dissimilar intrinsic structures and it may become desirable to approximating such data locally in the dataspace. With proper partitioning of the dataspace and representation of the PCA structure for each partition, it may be possible to identify regions of the data space with similar structures and others with dissimilar structures.

For datasets such that the relationship between the covariates can be described as “non-linear”, “branched”, “disconnected” or generally as “complex”, usually it implies that different regions of the dataspace have different intrinsic structures. PCA being an unsupervised and non-parametric analysis will only give the global structure of the

dataset which may not efficiently approximate such datasets. Approximating complex data has led to the development of non-linear techniques and extension of PCA in order to deal with such complexities. For such complex datasets, it is expected that manifold modelling techniques like principal curves, SOM, elastic maps and others, should approximate the structure of this data at the various regions better. However, most manifold modelling techniques approximate data using objects of lower dimension *k* usually for $k \leq 2$, thus placing some restrictions on the approximation in a given region. An attempt to approximate a dataset with lower dimensional object of dimension $k > 3$ often increases the difficulty of manifold modelling methods.

To further explore the structures of a dataset, in this chapter, we will examine and analyse the PCA structure(s) at locally at various spatial locations in the data space. By localization in the data space, I mean that if we consider a target data point $\mathbf{x}_o$ in the dataset X, we will analyse the PCA locally around $\mathbf{x}_o$ by performing PCA only on the data within a given neighbourhood of the target data point. This localized PCA can be performed on several selected distinct target data points in the dataset and the localized PCA structures corresponding to these selected target points can be further analysed. The analysis of the local PCA structures may be able to reveal how the structure of the data changes and possibly help to identify spatial regions of the dataspace with similar intrinsic structures.

While we can see localized PCA as an approximation to manifold modelling in a neighbourhood of a data point $\mathbf{x}_o$, it is able to handle branched data and disconnected data better than many manifold modelling methods. This may not include methods designed for branched data as discussed in section 2.3.6) see [24]. In addition to this, localized PCA enjoys the advantages of PCA which includes being easy in terms of interpretation, computation (even when you approximate the data at various region of the data space using hyperplane of dimension up to $k < m$ where *m* is the dimension of the data space) and inference.

In this chapter, we will consider how to find localized PCA about various data points in the dataset and using the representation discussed in the previous chapter we will analyse this localized PCA structures to reveal some intrinsic structures in the

dataset, we will also look at how we can combine localization of dataset both in scale and space to further explore a given dataset and finally, we will discuss some examples.

4.2 Localization in the Data Space

Given a target data point $\mathbf{x}_o$ in the dataset X, to find the principal components of the dataset locally around the point $\mathbf{x}_o$ require that we perform PCA for observation close to the point $\mathbf{x}_o$. This localization is achieved by using a kernel function $\Phi_r(\mathbf{x}_o, \mathbf{x}_i)$ to assign weight to observation $\mathbf{x}_i$ based on its distance from the target point $\mathbf{x}_o$. The kernel function is usually parameterized by *r* which dictates the radius of the neighbourhood. There are several kernel functions that can be used, for example; the (117) Gaussian kernel, Nadaraya-Watson kernel, Epanechnikov quadratic kernel and others. However, for this thesis, we chose the Euclidean metric to measure distance between vectors and we will define the kernel function as

$$\Phi(\mathbf{x}_i, \mathbf{x}_o) = \begin{cases} 1 & when \ \dfrac{\| \mathbf{x}_i - \mathbf{x}_o \|_2}{r} \le 1 \\ 0 & otherwise \end{cases} \tag{4.1}$$

Where $d_{\mathbf{x}_o}^{\min} \le r \le d_{\mathbf{x}_o}^{\max}$, $d_{\mathbf{x}_o}^{\min}$ is the minimum pairwise distance (greater than zero) between $\mathbf{x}_o$ and any point $\mathbf{x}_i$ and $d_{\mathbf{x}_o}^{\max}$ is the maximum pairwise distance between the target point $\mathbf{x}_o$ and any point $\mathbf{x}_i$ in the dataset. The kernel $\Phi(\mathbf{x}_i, \mathbf{x}_o)$ as defined in (4.1) assigns the value 1 to the pairwise distance of any data point $\mathbf{x}_i$ within *r* neighborhood of $\mathbf{x}_o$ and it assigns the value 0 to any pairwise distance that is not within *r* neighborhood of $\mathbf{x}_o$.

Therefore PCA seeks the set of orthonormal vectors $\{\mathbf{v}_1, \mathbf{v}_2, ...\mathbf{v}_m\}$ which maximize

$$\max_{v_1,...,v_k} \sum_{\alpha=1}^{k} \mathbf{v}_\alpha^T \tilde{S}_{x_o} \mathbf{v}_\alpha \ .$$

$$\text{Subject to } (\mathbf{v}_\alpha, \mathbf{v}_\beta) = \delta_{\alpha\beta}$$

$$\alpha, \beta = 1,2,..,k \ .$$

Where the local covariance matrix $S_{x_o} = \sum_{i=1}^{n} \Phi_i^x (\mathbf{x}_{ij} - \boldsymbol{\mu}_j^x)(\mathbf{x}_{ik} - \boldsymbol{\mu}_k^x)^T$ and

$$\boldsymbol{\mu}^{x} = \frac{1}{\sum_{i=1}^{n} \Phi(\mathbf{x}_i, \mathbf{x}_o)} \sum_{i=1}^{n} \Phi(\mathbf{x}_i, \mathbf{x}_o)\mathbf{x}_i .$$

Proposition 4.1**:** *The local PCA of dataset* $\{\mathbf{x}_i\}$ *within r neighborhood of a target point* $\mathbf{x}_o$ *is obtained by finding for each dimension 1≤k≤m a k-dimensional subspace L of the dataspace, which maximizes*

$$D_X = \sum_{i<j} \Psi_{ij} \parallel P_L(\mathbf{x}_i - \mathbf{x}_j) \parallel_2^2 \tag{4.2}$$

where $\Psi_{ij} = \Phi(\mathbf{x}_i, \mathbf{x}_o)\Phi(\mathbf{x}_j, \mathbf{x}_o)$ and $\Phi(\mathbf{x}_i, \mathbf{x}_o)$ is as defined in (4.1).

Proof

From definition 4, subspace of *k* first principal components is found by maximizing $\sum_{i<j} \| P_L(\mathbf{x}_i - \mathbf{x}_j) \|_2^2$, $i, j = 1,2,..n$ on the manifold of *k*-dimensional subspaces. For local PCA, only points within *r* neighborhood of $\mathbf{x}_o$ are considered. The constraint imposed by localization implies that $P_L(\mathbf{x}_i - \mathbf{x}_j)$ is retained in the computation if $\mathbf{x}_i, \mathbf{x}_j \in \mathrm{N}_{x_0}^r$ where $\mathrm{N}_{\mathbf{x}_0}^r = \{\mathbf{x}_i : \| \mathbf{x}_i - \mathbf{x}_0 \| \le r\}$ and $P_L(\mathbf{x}_i - \mathbf{x}_j)$ is excluded whenever $\mathbf{x}_i, \mathbf{x}_j \notin \mathrm{N}_{x_0}^r$. To satisfy localization constraint, we introduce weight $\Psi_{ij} = \Phi(\mathbf{x}_i, \mathbf{x}_o)\Phi(\mathbf{x}_j, \mathbf{x}_o)$. Therefore local PCA about target point $\mathbf{x}_o$ is weighted PCA that seeks to maximize (4.2).

The weight Ψ_{ij} ensures that all pairwise distance considered in the optimization problem (4.2) are local to $\mathbf{x}_o$. For ease of computation, we will like to translate the data points within an *r*-radius neighborhood of $\mathbf{x}_o$ to have mean **0**; therefore we compute the local mean at point $\mathbf{x}_o$ as

$$\mu_{x_o} = \frac{1}{\sum_{i=1}^{n} \Phi(\mathbf{x}_i, \mathbf{x}_o)} \sum_{i=1}^{n} \Phi(\mathbf{x}_i, \mathbf{x}_o)\mathbf{x}_i \ . \tag{4.3}$$

The dataset *X* can therefore be centralized by subtracting μ_{x_o} from the observations in *X*. Henceforth for any analysis or computation that entails localizing about a target point, we shall assume that the datasets have been centered about μ_{x_o}.

Proposition 4.2: *The k-dimension projection that maximizes* $\sum_{i<j} \Psi_{ij} \| P_L(\mathbf{x}_i - \mathbf{x}_j) \|_2^2$ *is obtained by taking the direction vectors to be the p highest eigenvectors of the matrix* $X^T L^{\Psi} X$ *where*

$$L_{ij}^{\Psi} = \begin{cases} \sum_{j=1}^{n} \Psi_{ij} & i = j \\ -\Psi_{ij} & i \neq j \end{cases} \tag{4.4}$$

proof

Since maximizing $\sum_{i<j} \Psi_{ij} \| P_L(\mathbf{x}_i - \mathbf{x}_j) \|_2^2 = \sum_{\alpha=1}^{k} \left[\sum_{i<j} \Psi_{ij} (\mathbf{v}_\alpha, \mathbf{x}_i - \mathbf{x}_j)^2 \right]$ is an optimization problem for weighted PCA, as discussed in 3.2.1, the problem can be stated as maximize

$$\max_{v_1,\ldots,v_k} \sum_{\alpha=1}^{k} \mathbf{v}_\alpha^T \tilde{M} \mathbf{v}_\alpha$$
$$\text{Subject to } (\mathbf{v}_\alpha, \mathbf{v}_\beta) = \delta_{\alpha\beta} \quad \alpha, \beta = 1,2,..,k \tag{4.5}$$

where the weight w_{ij} here is given as Ψ_{ij}. From *proposition 3.2* we have that $\tilde{M} = \sum_{i<j} \Psi_{ij} \left[(\mathbf{x}_i - \mathbf{x}_j) \otimes (\mathbf{x}_i - \mathbf{x}_j) \right]$ can be written as $X^T L^{\Psi} X$ and from *theorem 1.1 we* have that the *p*-dimension projection that maximizes (4.5) is given by the the eigenvectors corresponding to the sorted eigenvalues of the matrix $\tilde{M} = X^T L^{\Psi} X$.

Propositions 4.1 and 4.2 allow us to treat PCA localization in dataspace as an extension of MPCA and therefore we can adapt the the methods and analysis developed in chapter three to PCA localization in dataspace.

4.3 Selection of Target Points

The principal components of local PCA depend on the target point $\mathbf{x}_o$ and the radius of neighbourhood *r*. If we want to explore the structure of a dataset for better

understanding, then it is important to select the target points carefully. Given a dataset X with n datapoints, one can consider the localized PCA of X for a given r neighborhood at every data point, however this can become cumbersome for large n. Since the objective of localized PCA is to understand the intrinsic structure(s) of the dataset using localized PCA, we propose some ways to do this based on heuristics.

The first approach proposed is to randomly select a subset $T = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t\}$, $t < n$ of the dataset X under the condition that for each point $\mathbf{x}_i \in T$, a corresponding set N_i^r, $i = 1, ..., t$, which is the set of datapoints in the r neighborhood of target point $\mathbf{x}_i$ is defined and the union of all the set N_i^r (i.e. $\bigcup_{i=1}^{t} \mathrm{N}_i^r = R^n$) cover all the data points of the dataset or cover the datapoints up to some small exclusion. The localized PCA for each $\mathbf{x}_i \in T$ is performed as discussed in section 4.2.

The second approach is to first partition the dataset into clusters using any standard clustering algorithm (such as k-means algorithm for sufficiently large k) and then perform a cluster wise PCA analysis using the center of mass of each cluster as the target point rather than datapoints (i.e. the elements in the set $T = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k\}$, is the center of mass for the cluster C_k, $k < n$. This approach was proposed as a tool for exploratory data analysis in [14] [29] and can be seen as a generalization of the k-mean algorithm in the sense that while k-mean is clustered around a point, now it is clustered around a hyperplane. Consequent upon the so called "curse of dimensionality", we should mention that clustering algorithm may not perform well for very high dimensional data.

It should be noted that for this approach, the radius of neighborhood may be different for each cluster and if we retain the kernel function as defined in (4.1), then in the situation in which a cluster C_k is bigger than N_k^r then we are using a subset of the cluster C_k, and the case where cluster C_k is smaller than N_k^r then elements of other clusters which are within the r neighborhood of target point $\mathbf{u}_k$ of C_k will be included in computing the localized PCA at $\mathbf{u}_k$.

However we can redefine the kernel function $\Phi(\mathbf{x}_i, \mathbf{x}_o)$ such that only the elements of the same cluster are included in the localized PCA analysis of a cluster with target point $\mathbf{u}_k$.

$$\Phi(\mathbf{x}_i, u_k) = \begin{cases} 1 & when\ \ x_i \in C_k \\ 0 & otherwise \end{cases} \tag{4.6}$$

In the situation that some clusters have more datapoints than other clusters, we can adjust the sample size for each cluster so that there is a balance by taking a sample of the larger clusters before performing localized PCA.

For the second approach, it should be noted that since the localized PCA are performed cluster wise, then the quality of the cluster affects the result of the localized PCA. We remind ourself that the quality of clustering algorithm is usually influenced by the initial condition such as the initialization of the cluster centre; if not well selected the algorithm could be trapped in local minima. The quality of the clustering algorithm is also influenced by the the number *k* of clusters as this as to be determined apriori.

To overcome this, Einbeck et al, [24] proposed the recursive local PCA. Starting from a single partition of the dataset, each partition is recursively split up into two partitions if the dataset can be more effectively approximated by two hyperplanes instead of one. Also any two neighbouring partitions can be joined together if it can be effectively approximated with a single hyperplane thus allowing the algorithm to get around some suboptimal local extrema. In order to test if a partition $R^{(q)}$ can be better approximated using two hyperplanes rather than one, Einbeck et al proposed that we find the eigenvalues $\lambda_1^{(q)}, \lambda_2^{(q)}, \ldots \lambda_m^{(q)}$ of the covariance matrix of the partition $R^{(q)}$ arranged in descending order, split the partition $R^{(q)}$ from the mean point orthogonally to the first principal component of the partition into two partitions $R^{(l)}$ and $R^{(r)}$. And the split is retained if

$$\frac{\lambda_1^{(q)} + \ldots + \lambda_k^{(q)}}{\lambda_1^{(q)} + \ldots + \lambda_m^{(q)}} < C \cdot \left(\frac{n^{(l)}}{n^{(q)}} \frac{\lambda_1^{(l)} + \ldots + \lambda_k^{(l)}}{\lambda_1^{(l)} + \ldots + \lambda_m^{(l)}} + \frac{n^{(r)}}{n^{(q)}} \frac{\lambda_1^{(r)} + \ldots + \lambda_k^{(r)}}{\lambda_1^{(r)} + \ldots + \lambda_m^{(r)}} \right). \tag{4.7}$$

Where $n^{(q)}$ is the number of observation in partition q and C is a constant usually chosen as 1. If any two neighboring partition does not satisfy the condition (4.7), then we join them together. Two partitions $R^{(l)}$ and $R^{(r)}$ are defined to be neighbors if for at

least one observation $\mathbf{x}_i$, the partitions $R^{(l)}$ and $R^{(r)}$ are amongst the $s > 2$ "closest" partitions. The recursive local PCA algorithm is discussed below.

Recursive local PCA algorithm

1) Start with a single partition $R^{(1)}$ containing all the data.
2) Iterate…
   a) Test each partition $R^{(q)}$ and split if criterion (4.10) is satisfied,

      Iterate

      i. For each partition $R^{(q)}$ $(q = 1,2,\ldots,Q)$, compute the "local" PCA and obtain the corresponding eigenvalues $\lambda_1^{(q)} \geq \lambda_2^{(q)} \geq \ldots \geq \lambda_k^{(q)}$

      ii. Update the partitioning $R^{(i)}$, $(i = 1,\ldots,Q)$ by allocating each observation $\mathbf{x}_i$ to the nearest partition, i.e. the partition whose hyperplane segment is closest to $\mathbf{x}_i$.

   b) For each pair of partition $R^{(l)}$ and $R^{(r)}$ that are "neighbors", test whether it should be joined together. A pair of partition is joined if it does not satisfy the criterion (4.7)
   c) Stop when there is no change in the allocation of observation to partitions.

This algorithm yields disconnected hyperplane segments and can be seen as finding tangent approximations to the principal curve or manifold. However this algorithm could also suffer for data in very high dimension as distances are not local in high dimension [10].

In [60], Kambhatla et al, using empirical evidence showed that dimension reduction using local PCA performs better than PCA and even nonlinear model built by five–layer autoassociative neural network. In addition the training time for local PCA was significantly faster than the neural network.

4.4 Representation of PCA Structures in Space

As mentioned earlier, we need to select several distinct data points and find the localized PCA structure at this point. We note that the localized PCA structure at a given target point $\mathbf{x}_o$ depends on the point $\mathbf{x}_o$ and the radius of neighbourhood *r*. This implies that for a given pair $(\mathbf{x}_i, r)$ there is an associated PCA structure given by the eigenvectors of the matrix $\tilde{M}$ as given in (4.7).

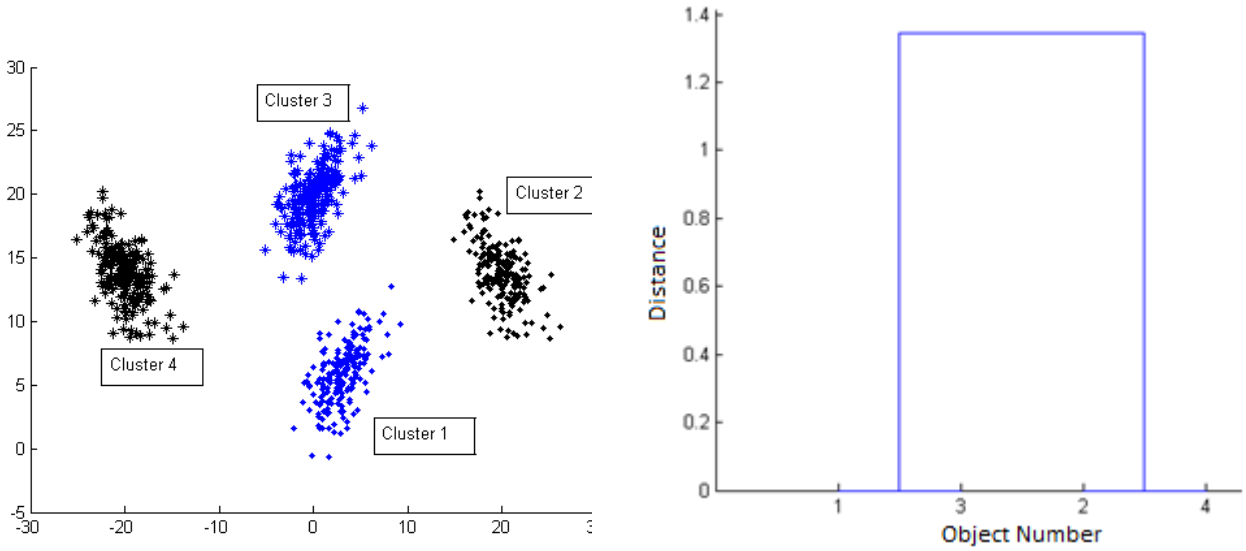
Based on PCA representation proposed in chapter 3, the PCA structure for a given pair $(\mathbf{x}_i, r)$ will be represented by the projector matrix $\rho_k = \sum_{i=1}^{k} \mathbf{e}_i \otimes \mathbf{e}_i$, $k = 1,2,\ldots,m-1$. Where ρ_k is a matrix of rank *k*. Therefore the full description of the PCA structure of a dataset *X* for a given pair $(\mathbf{x}_i, r)$ is given by the cortege of projectors $\rho_1, \rho_2, \ldots, \rho_{m-1}, \rho_m$ where the matrix ρ_k is the projection matrix to the *k*-dimensional subspace spanned by the vectors $\mathbf{e}_i$, $i = 1,2,\ldots,k$.

With the local PCA structures for each target point represented as discussed above (see section 3.4.3), we can further analyse these PCA structures in order to explore and understand the intrinsic structure(s) of the dataset. Of interest could be to identify certain regions in the data space with similar PCA structures or dissimilar PCA structures. To identify similar structure in the data space, we will cluster the PCA structures of the data space localization at the target points.

Let $T = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t\}$, $t < n$ be the set of target points for dataset *X*, we define the similarity between any two points $\mathbf{x}_i, \mathbf{x}_j \in T$ for a fixed *r*, as the distance between the orthogonal projection matrices representing the PCA structure of the localized PCA at the target points using an invariant norm such as the frobenius norm (see section 3.5). Therefore the result of the clustering algorithm on the data space groups together target points $\mathbf{x}_i$ with similar PCA structures. This can be interpreted as finding regions in the data space with similar PCA structures. For datasets with noise (with low variance), clustering of the PCA structure of targets points using ρ_k where *k* has been chosen to exclude directions with low variance can be interpreted as finding the regions in the data with similar intrinsic structures since the distracting variance have been eliminated by the choice of ρ_k.

Due to the presence of noise either in the data or the neighbourhood of some target points, two different regions of the dataset with the same PCA structure may not necessarily have the same projection matrix. That is the subspace spanned by the loading vectors are not necessarily parallel, however clustering algorithm should be able to identify such regions and group them in the same cluster as it will be expected that the PCA subspaces of such regions will not vary by a significant amount and hence the distances between the orthogonal projection representing such subspaces will be small compare to subspaces with different PCA structures.

We demonstrate localization in space with the example given in figure 17. Figure (a) show a 2 dimensional dataset with four well separated clusters. Using *k*-means algorithm to cluster the dataset, PCA was performed on each cluster. The PCA structure of each cluster C_k with mean point u_k, $k=1,\ldots,4$ is represented by the projection matrix ρ_1. We cluster the projector matrix to reveal clusters with similar PCA structures. The dendrogram, figure (b), clearly suggest two distinct PCA clusters. Clusters with similar PCA structures have been assigned the same colour in figure (a)



a) Scatter plot of simulated 2-d data. Clusters with similar PCA structures have the same colour

b) Dendrogram of Hierarchical clustering algorithm.

Figure 17

Also few data points (including the mean point) of the dataset have been selected for analysis. For each target point, we consider the local PCA at various scales of regular interval from 0 to 1. The scale 0 to 1 has been chosen such that the maximum pairwise distance $r_{\mathbf{x}_0}^{\max}$ between target point $\mathbf{x}_0$ and other point in the dataset has been scaled to 1. Therefore a given scale s corresponds to $s \times r_{\mathbf{x}_0}^{\max}$ radius of neighbourhood. The points selected are given in table 7

Table 7.

Target points	
A	[26.20, 9.56]
B	[-25.10, 16.43]
C (mean Point)	[0.80, 13.33]
D	[-13.8, 9.56]
E	[4.68, 8.06]

The target point A and B have been selected because it represent the pair of points with largest distance which can be viewed as the end points of the dataset. The target point C is the mean point while points D and E have been chosen randomly. For each target point $\mathbf{x}_i$, the PCA structure was clustered over the scale s and the result is shown in figure 18. For each target point, the inconsistency coefficients indicate 2 natural clusters of the PCA structures.

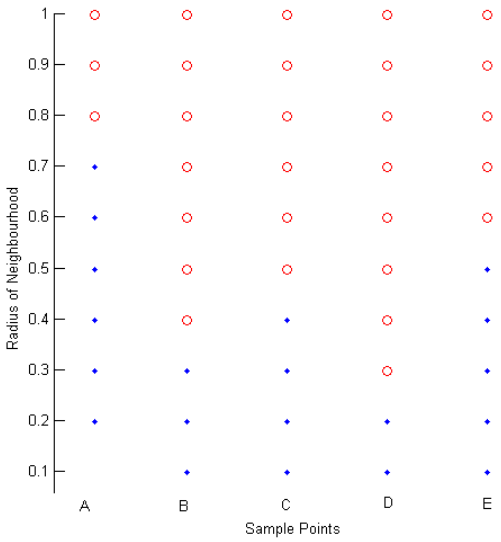


Figure 18. Cluster of local PCA structure of the data shown in figure 17 for the target points in table 7.

Note: scales with fewer than 5 data points have been exempted. Also clusters of different target points with similar colour do not imply that the PCA structures are the same. The

colours are specific to each target point and only show the natural clusters as we increase the range for the specific target point.

Looking at cluster analysis at target points A and B, we can observe the change in the PCA structure as we navigate from one end of the dataspace to the other end. Also the cluster analysis at the mean point of the dataset reveals how the PCA structure changes as the radius of neighbourhood increases (i.e. as we move from the center of the dataset towards the end of the dataset).

4.5 Localization in Scale and Space

We have introduced PCA localization in scale in chapter 3 and PCA localization in the dataspace in this chapter. These two approaches to localization can be combined for a robust exploration and modelling of the data. Combining these two approaches will result in PCA structure(s) that can be represented and analysed to further explore the dataset. We recall that the MPCA structure of a dataset *X* depends on the scale (*l, u), (*see section 3.4), whereas local PCA structure at a target point $\mathbf{x}_o$ depends on $(\mathbf{x}_o, r)$ where *r* is the radius of neighbourhood

In this section we shall combine the two localizations to analyse a dataset (i.e. we analyse the dataset both locally in space and in scale simultaneously) and we will demonstrate with an example how we can make localized PCA in space more robust by combining with localization in scale. The problem can be stated as find the local PCA of a dataset *X* at a given target point $\mathbf{x}_o$ and at a given scale (*l,u).*

Proposition 4.3**:** *The local PCA of data point* $\mathbf{x}_i$ *within r neighbourhood of a target point* $\mathbf{x}_o$ *and at a given scale* (*l,u) is obtained by finding for each dimension* $1 \le k \le m$ *a k-dimensional subspace L of the dataspace, which maximizes:*

$$D_X = \sum_{i<j} w_{ij} \Psi_{ij} \| P_L(\mathbf{x}_i - \mathbf{x}_j) \|_2^2 \tag{4.8}$$

Where w_{ij} *is defined as:*

$$\begin{cases} w_{ij} = 1 & l \le \| \mathbf{x}_i - \mathbf{x}_j \|^2 \le u \\ w_{ij} = 0 & otherwise. \end{cases} \tag{4.9}$$

(l,u) is as defined in section 3.4 and Ψ_{ij} *is defined as*

$$\Psi_{ij} = \Phi(\mathbf{x}_i, \mathbf{x}_o)\Phi(\mathbf{x}_j, \mathbf{x}_o)$$

and $\Phi(\mathbf{x}_i, \mathbf{x}_o)$ *is given by (4.1) or (4.6) .*

Proof

From *proposition 4.2* we have that local PCA about a target point $\mathbf{x}_o$ maximizes $\sum_{i<j} \Psi_{ij} \| P_L(\mathbf{x}_i - \mathbf{x}_j) \|_2^2$. In addition we seek a subspace that maximizes pairwise distance for datapoints $\mathbf{x}_i, \mathbf{x}_j$ such that $\| \mathbf{x}_i - \mathbf{x}_j) \|_2^2 \in [l, u]$ as introduced in section 3.3. *Proposition 4.3* follows from introducing weight w_{ij} as given in (4.9) in order to enforce the constraint introduced by scale (*l,u)* on local PCA about target point.

Let $\Lambda_{ij} = w_{ij}\Psi_{ij}$, then

$$\begin{cases} \Lambda_{ij} = 1 & \text{if } w_{ij} = 1,\ \Psi_{ij} = 1 \\ \Lambda_{ij} = 0 & \text{otherwise.} \end{cases} \tag{4.10}$$

then we can rewrite (4.8) as maximize

$$D_X = \sum_{i<j} \Lambda_{ij} \| P_L(\mathbf{x}_i - \mathbf{x}_j) \|_2^2 \tag{4.11}$$

This is a weighted PCA problem as discussed in section 3.2.1. From The problem can be stated as maximize

$$\max_{v_1,\ldots,v_k} \sum_{\alpha=1}^{k} \mathbf{v}_\alpha^T \tilde{M} \mathbf{v}_\alpha \tag{4.12}$$

$$\text{Subject to } (\mathbf{v}_\alpha, \mathbf{v}_\beta) = \delta_{\alpha\beta} \quad \alpha, \beta = 1,2,..,k$$

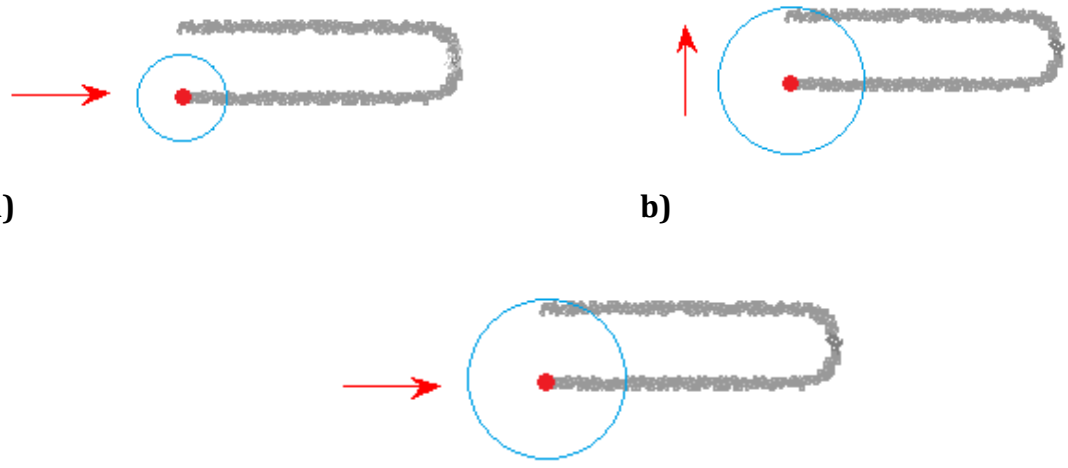
Where $\tilde{M} = \sum_{i<j} \Lambda_{ij} \left[(\mathbf{x}_i - \mathbf{x}_j) \otimes (\mathbf{x}_i - \mathbf{x}_j) \right]$ and can be written as $X^T L^\Lambda X$ (proposition 3.2). Finally from *theorem 1.1 w*e have that the *p*-dimension projection that maximizes (4.12) is given by the eigenvectors corresponding to the sorted eigenvalues of the matrix $\tilde{M} = X^T L^\Lambda X$.

From (4.10) we remark that the weight of 1 is assigned to a pair of data projection provided the assigned $w_{ij} = 1$ and $\Psi_{ij} = 1$; 0 otherwise. The weight $w_{ij} = 1$ ensures that only pairwise distances of datapoints within a given scale (*l,u)* are included in the analysis of principal component and that the weight $\Psi_{ij} = 1$ constraint the pairwise distances to only pairwise distances of datapoints within the *r*-neighborhood of a given target point $\mathbf{x}_o$. For ease of computation, we translate the data points within

an r-radius neighborhood of $\mathbf{x}_o$ to have mean $\mathbf{0}$ (where $\mathbf{0}$ is the zero vector) by subtracting μ_{x_o} as earlier given in equation (4.3).

Therefore the PCA structure depends on the point in the 4-space defined by $(\mathbf{x}_o, r, l, u)$ where $\mathbf{x}_o$ is the target point, r is the radius of neighbourhood, l is the lower limit of scale and u is the upper limit of scale.

To demonstrate how dataset approximation can benefit from the combination of localization in scale and space, let us consider the horseshoe data shown in figure 19 and a target point shown by the red point.The radius of neighbourhood have been depicted by the blue circle with the target point at the center. In figure (a), we see that local PCA (with the radius of neighbourhood) captures the structure of the data quite well (as shown by the red arrow). In figure (b) we observe that when the radius of neighbourhood is increased, due to the presence of some influential datapoints, the first principal component given by local PCA seems to have distorted the data badly. However by combining PCA localization in space and scale, we have been able to recover the structure of the data local to the target point as shown in figure (c).



a) **b)**

c)

Figure 19. (a) and (b) are PCA localization in space with the radius of neighbourhood depicted by the blue circle. The red arrow shows the direction of the first principal component. (c) is the PCA localization in both data space and scale using standard scale of $(0,0.5)$.

4.6 Conclusion

In this chapter we looked at localization of PCA in the dataspace. The local PCA about a target point is obtained by computing PCA on data within a neighbourhood of the target point. The neighbourhood is specified using a kernel function. We represented the resulting PCA structure using the orthogonal projection matrix associated with the subspace of the principal components. The local PCA structures of selected target points were further analysed to reveal intrinsic structure(s) in the dataset. Local PCA can be seen as a tangent approximation to nonlinear PCA methods and it provides easy computation, exploration and analysis alternative to nonlinear PCA methods especially in dealing with complex data. We also combine localization in dataspace and scale. Using an example, we demonstrate how data exploration and modelling could benefit from the combination of these two localization approaches. We should remark that computing 'local PCA' in step 2 of the recursive local PCA algorithm discussed in this chapter can be replaced with localized PCA in *scale* or the combination of localized PCA in *scale* and *space*.

We can extend the idea of clustering of scales and localization in space and scale to other projection methods. By projection methods we mean approximation methods that project data to a linear manifold (see section 2.5) such as PCA, weighted PCA, reduced-rank linear discrimant analysis, partial least square e.t.c. For a given dataset X, the dataspace can be partitioned and such methods applied locally on a region of the dataspace. We can then derive a set of orthogonal basis for the subspaces resulting from applying these methods on the dataset and these set of basis can be represented using the orthogonal projection matrix which can be further analysed to provide insight to the structure of the dataset.

Chapter 5

Data Exploration Using Localized PCA

5.1 Introduction

In the previous two chapters, we considered PCA and its localization in two different directions; localization in scale and localization in dataspace. PCA being a non-parametric analysis leads to a solution which favours large distances in the data which could obfuscate other interesting structures in the dataset that may exist at smaller distances. In chapter 3, we proposed studying MPCA structures of a dataset as a tool to explore the intrinsic structure(s) of the data. The MPCA structure(s) are the subspaces of the dataspace which maximize the sum of the projection of weighted pairwise distances of the dataset, where the weights have been chosen to impose the constraint that only pairwise distances of datapoints within a given scale are maximized. Also, in chapter 4 we considered localization of the PCA in the dataspace and also the combination of both localization approaches. Using the representation discussed in chapter 3 and with further analysis, we demonstrated that analysis of the result of local PCA can reveal some hidden structures of datasets (especially for datasets with clear multiscale structures) and to identify regions with similar PCA structures. In this chapter we will use these methods as a tool for data exploration and also for data approximation on some simulated and real data.

5.2 Datasets used

Dataset I: *Artificial dataset* as given in example 3 of chapter 3. The data points are distributed on a plane with some outlying points and embedded into 3-space. See figure 16a.

Dataset II: *Vertebral column dataset*. This dataset, available online on the UCI machine learning repository [73] contains 310 samples, and each sample is represented by 6 biochemical features derived from the shape and orientation of the pelvis and lumbar spine. Therefore each sample can be viewed as a point in $\mathbb{R}^m$. This dataset was used in two related classification tasks; the first one was to classify the orthopaedic patients

into three classes (*Hernia, Spondylolisthesis, Normal*) and the second task was to classify the patients into two classes (*normal or abnormal*).

Dataset III. *Breast tissue dataset.* This dataset also available online on the UCI machine learning repository [73] contains 106 samples, with each sample represented by nine features computed from the impedance spectrum of freshly excised breast tissue. This data was used in two related classification tasks. The samples contain six classes namely: carcinoma, fibro-adenoma, mastopathy, glandular, connective, and adipose.

Dataset IV. *Iris Dataset.* This dataset contains 50 samples each of 3 species of the iris plant. Each sample is represented by 4 features and the dataset was used to train a classifier to predict the specie of iris plant. The data is available online on the UCI machine learning repository [73].

Dataset V. *Energy Efficiency Dataset* available online at the UCI machine Learning Repository [73]. This dataset contains 768 samples and 8 features and used to predict 2 different outputs (Heating Load and Cooling Load). The features are from energy analysis using 12 building shapes simulated in Ecotect.

Dataset I-III were analysed for localization in scale while dataset IV and V were analysed for localization in space.

5.3 Pre-Processing Data for MPCA

Since the features are measured in different units, it becomes desirable to make the features dimensionless for the purpose of comparison and analysis; therefore each variable in the dataset have been normalized. Different normalizations of the dataset are possible such as: normalization to unit variance or to unit mean; normalization with respect to the range or normalization to unit length. For the dataset analysed in this chapter (except for dataset I), the data have been normalized to unit variance. As earlier mentioned in section 3.6, MPCA is not invariant to normalization. Normalization changes the topology of the data and this in turn changes the PCA and MPCA structures of the data. The choice of normalization should be problem dependent and carefully chosen.

As discussed in section 3.4, there is a continuum of scales (l,u) in which MPCA can be computed, hence in application there is a need to sample the space of scales. We have standardized the set of scale such that maximum distance equals one (as described in section 3.3) and sampled the space of scale uniformly from 0 to 1.

5.4 Multiscale Principal Component Analysis of Datasets

Here, we used the MPCA algorithm as given in section 3.3.1 with the following additions:

Step 2: In addition to centralizing the dataset by subtracting the mean, the variables are also normalized to unit variance as previously mentioned. Hence we use the z-score of the variables.

Step 4**:** Finite points on some regular grid of the interval of scale was selected (see figure 20) such that adjacent points are of distance 0.1**.** These points correspond to various scales (localization) at which MPCA will be computed.

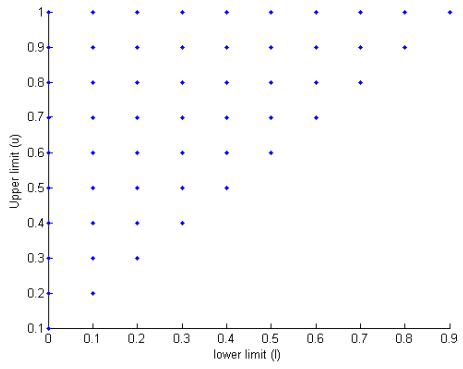
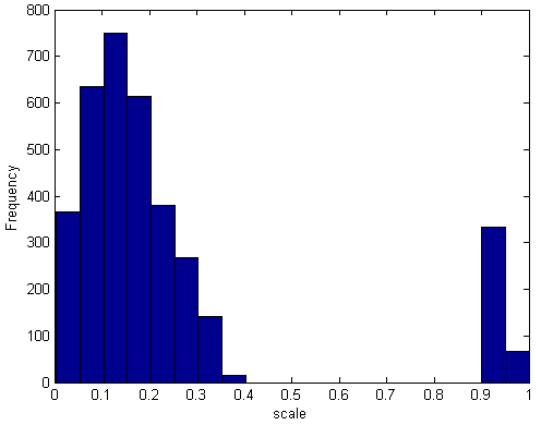
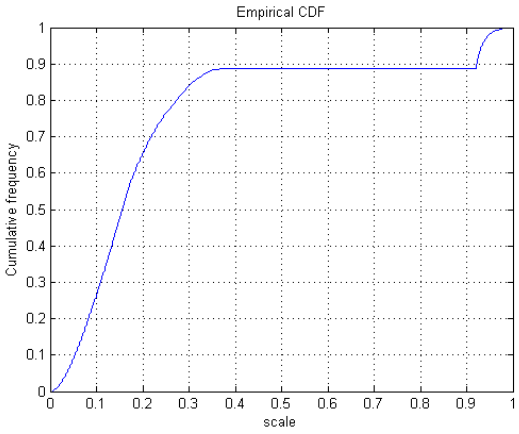


Figure 20: The sample of the standard scale selected for analysis.

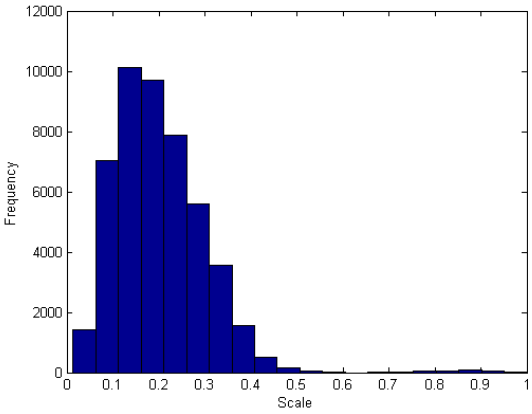
Find below the distribution of the pairwise distance of the datapoints for dataset I-III



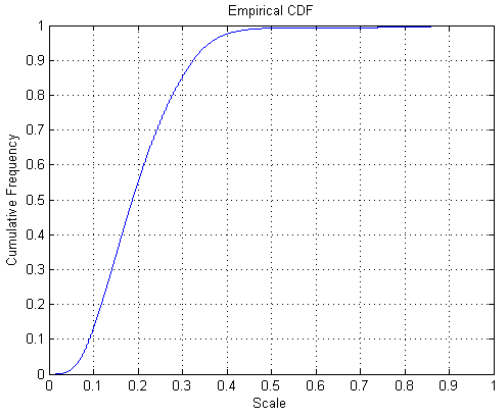
Dataset 1 :



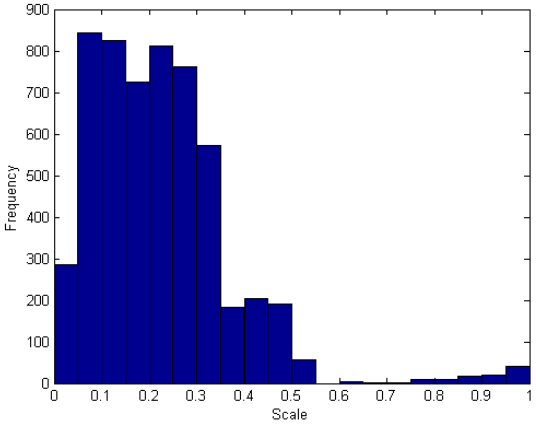
Dataset 1



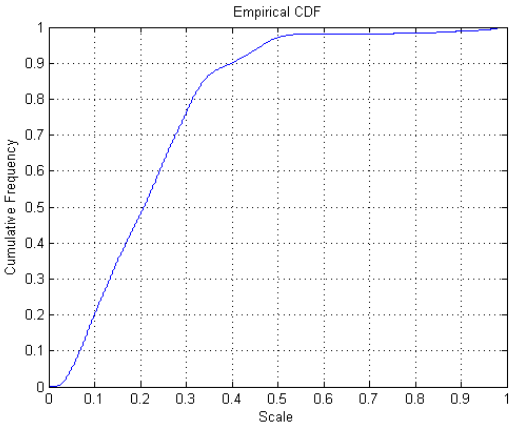
Dataset 2



Dataset 2



Dataset 3



Dataset 3

Figure 21. Histogram showing the distribution of pairwise distances and cumulative frequency of the datasets

For each point (l,u) in the space of scale, the PCA structure is scale dependent and represented by the projector matrix $\rho_k = \sum_{i=1}^{k} \mathbf{e}_i \otimes \mathbf{e}_i$ corresponding to the subspace of the principal component where $\mathbf{e}_i$ is the eigenvector which spans the i-th principal component.

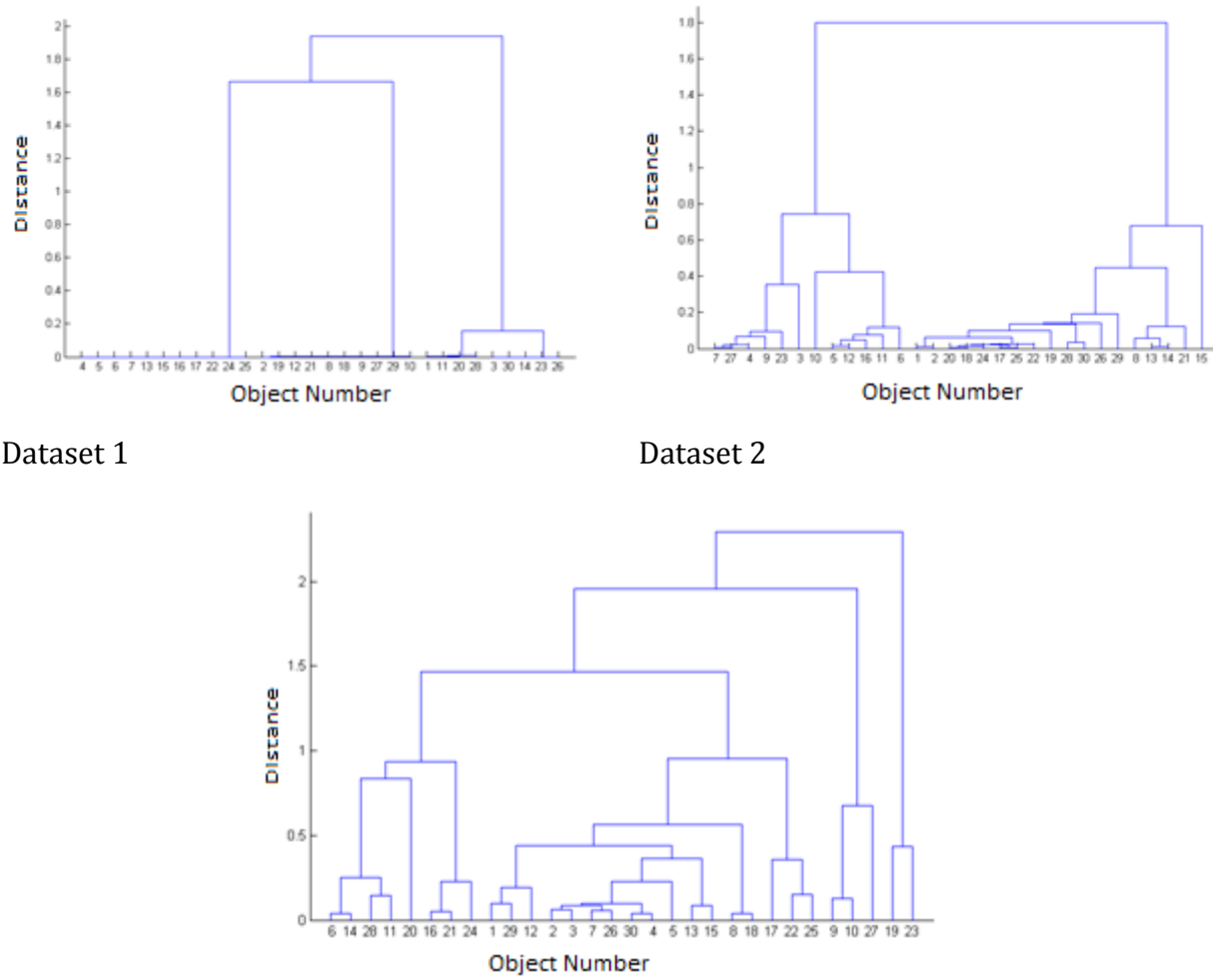
For the artificial data, the PCA structure for each scale of the MPCA has been represented by ρ_2 , for vertebral column dataset, ρ_4 and for the breast tissue dataset ρ_3. The choice of k in ρ_k is based on the result of the inconsistency coefficient, cophenetic correlation coefficient and visual inspection of the dendrogram when the dimension of subspace chosen to approximate the data is k. These criteria used will be discussed next.

To further explore the data, we clustered the interval of scale as described in section 3.5, using agglomerative hierarchical clustering. Since in hierarchical clustering, eventually all links are joined together at some level, there is a need to decide the natural cluster division of the scales. Deciding the natural cluster division of a data is a classical problem in clustering and many methods have been proposed to solve this problem. However we will be using the inconsistency coefficient. The inconsistency coefficient seeks to separate natural cluster in a given multi-level agglomerative hierarchical clustering by comparing the distance between two objects which is to be joined together with the distance of existing objects in the cluster. This can be viewed on a dendrogram as comparing the height of a link in a cluster tree with the heights of neighbouring links below it in the tree. A link is said to be consistent if the distance between the objects being joined is approximately the same as the distances between the objects they contain. In such case, the inconsistency coefficient will be close to zero. Whereas a high value of inconsistency coefficient implies that the objects being joined together is farther apart from each other than their components were when they were joined, and this suggests that the object probably belongs to a different cluster and hence indicate a border of natural division of the dataset.

It is important that the cluster tree generated by the hierarchical clustering method reflect the dissimilarity in the dataset. One way to measure this is using the cophenetic correlation coefficient. First we define the cophenetic distance between two objects as the distance between the two clusters that contain the objects. The distance between clusters is measured using linkage function. If the clustering method reflects the data well, then there should be a strong correlation between the cophenetic distance

and the pairwise distances of the original data. The cophenetic correlation coefficient is the correlation between the pairwise and the cophenetic distances of the data. We remark that the linkage function used can affect the quality of cluster and hence the cophenetic correlation coefficient.

To decide the number of clusters, we will be using the inconsistency coefficient, taking into consideration the cophenetic correlation coefficient (to ensure that the cluster distances represent the data distance efficiently) and also visual inspection of the dendrogram. See figure 22 for the dendrogram and table 1 for the inconsistency coefficient and cophenetic correlation coefficients of the datasets I-III.



Dataset 1 Dataset 2

Dataset 3

Figure 22. Dendrogram of hierarchical clustering of scales.

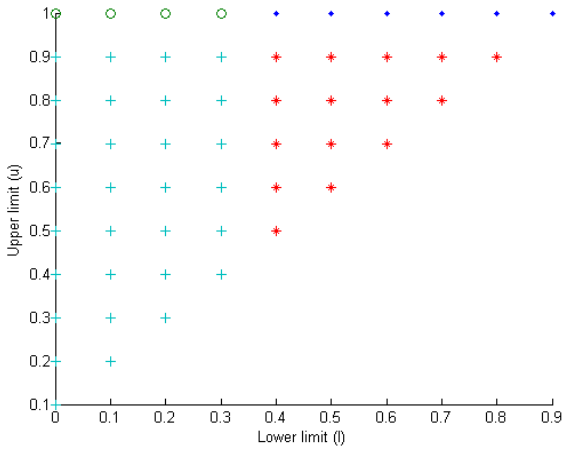
Based on the inconsistency coefficients of the datasets supported by cophenetic correlation coefficient and visual inspection of the dendrograms, we have selected 4

clusters for the artificial dataset, 3 Clusters for the Vertebral column dataset and 4 clusters for the Breast Tissue dataset.

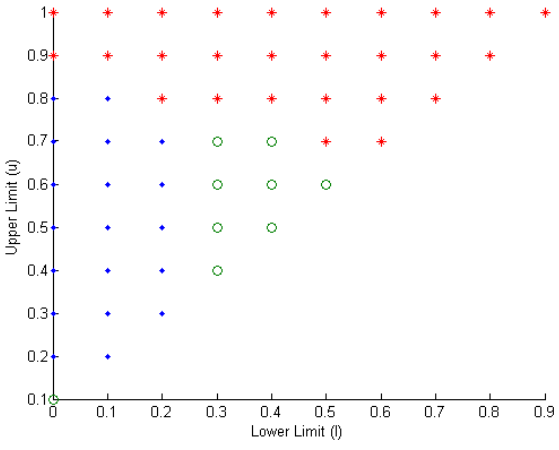
Table 8

Dataset	***Cophenetic***	***Inconsistency***
Artificial data (dataset I)	0.9979	1.1540 (4 clusters) 1.1547 (3 clusters)
Vertebral Column Dataset (dataset II)	0.9694	1.1384 (3 clusters) 1.1531 (2 clusters)
Breast Tissue (dataset III)	0.9120	1.0848 (5 clusters) 1.1543 (4 clusters)

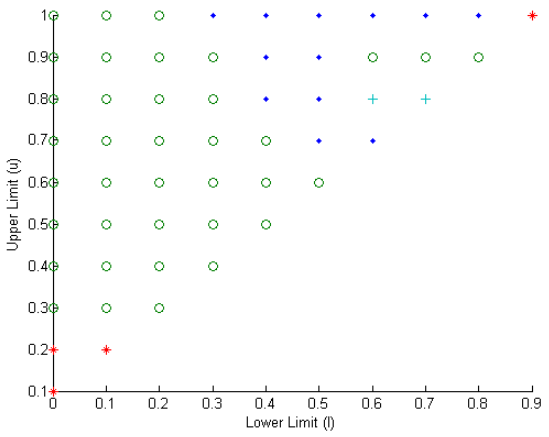
For each cluster, some scales were selected to represent the cluster and these representative scales were further analysed. The scales selected for each cluster include the medoid point, scale with maximum ratio of distortion (see section 3.8), scale with minimum ratio of distortion and some other random scales from each cluster. The clusters of scales for all the example datasets is visualized in figure 23.



Dataset 1



Dataset 2



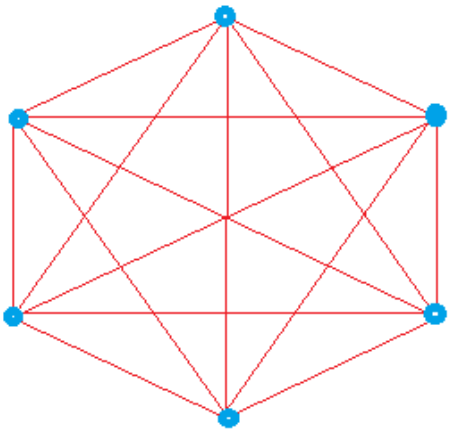
Dataset 3

Figure 23. Visualization of the cluster analysis of the interval of scales for the three datasets

5.5 Overfitting in MPCA

The number of pairwise distances of a given dataset with n samples is $n(n+1)/2$. Performing MPCA leads to the exclusion of some pairwise distances of datapoints which do not fall within the scale and for some scales, quite a large fraction of the number of pairwise distances may be excluded. In the situation where the result of MPCA is used to decide the best approximation of a dataset, selecting such scale could lead to overfitting in the sense that even though MPCA identifies the structure at such scales. However the structure represented at those scales may not be the best to represent the data. In this thesis, scales for which the excluded pairwise distances exceed 90% (i.e. scales for which the pairwise distances which fall within the scale is less than 10% of the total pairwise distances) have been exempted. Let *n* be the number of sample points, the total number of pairwise distances is $n(n+1)/2$ and for $n \geq 19$, 10% of $n(n+1)/2 \geq n$.

However, it should be noted that a given scale (l,u) does not correspond to a spatial location in the dataspace; it is pairwise distance across the dataspace with length within the interval (l,u). I demonstrate this with the example below. Consider a dataset located on the vertices of an hexagon (blue dots) with equal sides such that the Euclidean distance between any adjacent pairs of point is a (see figure 16a) then the red lines represent the pairwise distances between the datapoints.



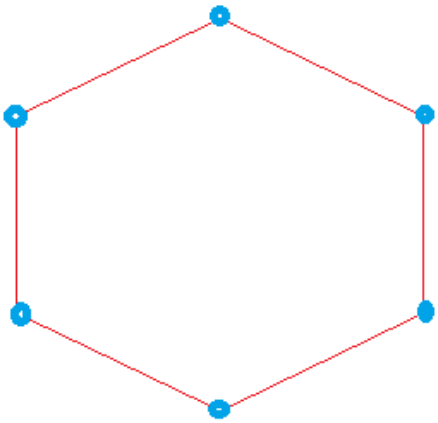


Figure 24a **Figure 24b**

However given a scale $(0, a)$ then figure (b) shows the pairwise distances that are included in the analysis of MPCA. For this dataset, we can observe that the pairwise distances within the given scale is distributed across the data space but not restricted to a spatial location in the dataset.

5.6 Data distortion

Given a data point $\mathbf{x} \in \mathbb{R}^m$, and let $P_L\mathbf{x}$ be its projection onto a linear manifold of dimension $k < m$, then for any pair of points $\mathbf{x}_i$ and $\mathbf{x}_j$,

$$\| P_L\mathbf{x}_i - P_L\mathbf{x}_j \|_2 \leq \| \mathbf{x}_i - \mathbf{x}_j \|_2 .$$

i.e. there is a contraction of distances when a data is projected to a subspace, with equality holding if the original data lies in a subspace of the dataspace.

In addition to the contraction, there is also a distortion of data during dimension reduction. For example, distant points in the data space can be closely projected and close points can be projected far from each other. For labelled data (i.e. dataset for which each datapoint has been allocated into a class), this could distort the picture of inter and intra-class distances that exist in the original datset. Distortion of data during dimension reduction implies the distortion of the distance structure of the data.

In dimension reduction, PCA relatively preserves large pairwise distances, which sometimes lead to the distance structure for smaller distances being badly distorted. We will like to investigate the distortion of PCA at various selected scales with MPCA at the

same scale. To investigate this we use the ratio of distortion as introduced in section 3.8.

$$\frac{\sum_{i,j=1}^{n} w_{ij} \| (P_L \mathbf{x}_i) - (P_L \mathbf{x}_j) \|^2}{\sum_{i,j=1}^{n} w_{ij} \| (\mathbf{x}_i - \mathbf{x}_j) \|^2}$$

Where w_{ij} is a binary weight based on the scale (l, u) as given in (3.19). This weight is introduced to restrict the pairwise distances to be consistent with the scale. The result is shown in tables 9, 10 and 11.

Table 9. Ratio of Distortion for dataset 1.

Scale	***Cluster***	***Ratio of Distortion PCA***	***Ratio of Distortion MPCA***
(0.4 , 1)	1	1	1
(0 , 1)	2	0.90	0.90
(0.3 , 1)	2	0.99	0.99
(0.1 , 1)	2	0.92	0.92
(0 , 0.1	4	0.67	1
(0 , 0.2)	4	0.74	1
(0 , 0.3)	4	0.82	1
(0 , 0.9)	4	0.83	1
(0.1 , 0.9)	4	0.85	1

Note that cluster 3 is missing. It has been omitted because it was classified as overfitting.

Table 10. Ratio of Distortion for dataset 2.

Scale	*Cluster*	*Ratio of Distortion PCA*	*Ratio of Distortion MPCA*
(0,0.2)	1	0.94	0.96
(0.2,0.6)	1	0.97	0.98
(0.1,0.7)	1	0.97	0.98
(0,0.1)	2	0.92	0.95
(0.3,0.6)	2	0.98	0.99
(0.2,1)	3	0.97	0.97
(0,0.9)	3	0.96	0.96
(0,1)	3	0.96	0.96

Table 11. Ratio of Distortion for dataset 3.

Scale	*Cluster*	*Ratio of Distortion PCA*	*Ratio of Distortion MPCA*
(0.3 ,1)	1	0.96	0.95
(0.4,1)	1	0.97	0.99
(0,1)	2	0.93	0.93
(0,0.5)	2	0.92	0.94
(0.2,1)	2	0.95	0.94
(0.1,0.8)	2	0.93	0.94
(0.1,0.2)	3	0.88	0.93
(0,0.2)	3	0.87	0.93
(0,0.1)	3	0.83	0.94

Note that cluster 4 is missing. It has been omitted because it was classified as overfitting.

For the artificial dataset, we observe that the PCA distorts the data for scales in cluster 4 (which represent distant structure for distances less than 0.9.) compared to MPCA at the same scale. We also observe that the distortion for clusters 1 and 2 (representing

large distances) is quite large. This supports the fact that PCA preserves large distances more efficiently sometimes to the detriment of small distances. See table 9.

For the vertebral column dataset (dataset 2), there is no much difference observed in the distortion due to PCA compared to MPCA. See table 10. For the Breast Tissue dataset (dataset 3), we observe an improvement in distortion for cluster 3 which correspond to scales with upper limit $u \leq 0.2$.

5.7 Preservation of local structures

For further analysis we will like to measure how well the subspaces resulting from MPCA and PCA preserves the small distances across the various scales and perhaps find the scale at which small distance are best preserved. To measure this, we will use a test proposed in [41], which calculate the *k* nearest neighbours in the projected space for every point and count how many of them are also point neighbour in the original data space. This test returns the average ratio of the intersection size of these two sets (*k* nearest neighbours in the projected space and the original dataspace) over k. A value close to 1 indicates a good neighbourhood preservation of the data in the projected space and a value close to 0 indicate a serious distortion in the projected space or a situation where many distant datapoints are projected close to each other in the projected space. See table 12, 13 and 14 for result.

Table 12. Average intersection of *k*-nn in the original space and projected space for dataset 1.

	PCA	***Cluster 1***	***Cluster 2***		***Cluster 4***				
Scale / *knn*	***(0,1)***	***(0.4,1)***	***(0.3,1)***	***(0.1,1)***	***(0,0.1)***	***(0,0.2)***	***(0,0.3)***	***(0,0.9)***	***(0.1,0.9)***
k = 3	0.56	0.55	0.56	0.56	0.93	0.93	0.93	0.93	0.93
k = 5	0.54	0.50	0.55	0.54	0.92	0.92	0.92	0.92	0.92
k = 10	0.58	0.55	0.58	0.58	0.94	0.94	0.94	0.94	0.94

Table 13. Average intersection of *k*-nn in the original space and projected space for dataset 2.

	PCA	***Cluster 1***			***Cluster 2***		***Cluster 3***	
Scale / *knn*	***(0,1)***	***(0,0.2)***	***(0.2,0.6))***	***(0.1,0.7)***	***(0,0.1)***	***(0.3,0.6)***	***(0.2,1)***	***(0,0.9)***
$k = 3$	0.74	0.80	0.82	0.80	0.83	0.82	0.75	0.74
$k = 5$	0.73	0.81	0.81	0.80	0.82	0.80	0.74	0.73
$k = 10$	0.77	0.84	0.83	0.83	0.84	0.82	0.77	0.78

Table 14. Average intersection of *k*-nn in the original space and projected space for dataset 3.

	PCA	***Cluster 1***		***Cluster 2***			***Cluster 3***		
Scale / *knn*	***(0,1)***	***(0.3,1)***	***(0.4,1)***	***(0,0.5)***	***(0.2,1)***	***(0.1,0.8)***	***(0.1,0.2)***	***(0,0.2)***	***(0,0.1)***
k = 3	0.78	0.77	0.74	0.78	0.76	0.77	0.79	0.80	0.80
k = 5	0.78	0.77	0.75	0.75	0.77	0.75	0.81	0.81	0.82
k = 10	0.85	0.83	0.81	0.84	0.84	0.84	0.87	0.87	0.86

For dataset 1, we observe that cluster 4 performs better in preserving local structure than cluster 1 and 2. For the dataset 2 (vertebral column dataset), cluster 1 and 2 performs better than cluster 3 (PCA belong to cluster 3). However there is not much difference for all clusters in dataset 3.

5.8 Class compactness

In the case where data is labelled and thereby partitioned into classes, we will like to also investigate how the various subspaces generated by various scales of MPCA preserve class distance structure in comparison to PCA using the class compactness test proposed in [41].

For a class C, Let 'class compactness' be defined as the average of a proportion of the points of class C among k nearest neighbours of the data point, calculated over the points from class C. We will expect that minimizing the distortion of local distance structure of the datasets in the projected space will help improve the cluster structure of the data approximation which can improve the class compactness of the data

especially for classes that can be identified by clustering the dataspace. See table 15 and 16. See also Appendix H to O for the remaining results.

Table 15. Class compactment result for dataset 2 (Class "*Abnormal*")

	PCA	***Cluster 1***			***Cluster 2***		***Cluster 3***	
Scale / *knn*	***(0,1)***	***(0,0.2)***	***(0.2,0.6))***	***(0.1,0.7)***	***(0,0.1)***	***(0.3,0.6)***	***(0.2,1)***	***(0,0.9)***
k = 3	0.74	0.79	0.80	0.79	0.84	0.83	0.74	0.75
k = 5	0.77	0.81	0.82	0.81	0.84	0.81	0.78	0.77
k = 10	0.81	0.84	0.84	0.84	0.85	0.84	0.81	0.81

Table 16. Class compactment result for dataset 2 (Class "*Normal*")

	PCA	***Cluster 1***			***Cluster 2***		***Cluster 3***	
Scale / *knn*	***(0,1)***	***(0,0.2)***	***(0.2,0.6))***	***(0.1,0.7)***	***(0,0.1)***	***(0.3,0.6)***	***(0.2,1)***	***(0,0.9)***
k = 3	0.77	0.91	0.92	0.89	0.97	0.96	0.78	0.76
k = 5	0.77	0.89	0.89	0.88	0.95	0.90	0.77	0.75
k = 10	0.83	0.92	0.92	0.90	0.96	0.94	0.85	0.83

The result of class compactment for dataset 2 is shown in table 15 and 16. The result shown is for the 2 class ("*Normal*" and "*abnormal*") classification problem. We remark that cluster 1 and 2 consistently outperform cluster 3 for the classification problems for both classes, in particular the best class compactment is achieved at MPCA with scale *(0, 0.1)*. The same applies to the result of the three class (*"Hernia"," Spondylolisthesis", "normal"*) classification problem. See table H, I and J in the appendix for the result.

For the breast tissue dataset (dataset 3), we observe slightly better class compactment for cluster 3 compared to cluster 1 and 2 for some classes (e.g. carcinoma, fibro-adenoma, mastopathy and glandular) and no difference in class compactment for some classes (e.g. connective and Adipose). See tables J to O in the appendix for the result.

5.9 Preservation of Global Structure

To measure how well the subspaces generated by MPCA and PCA approximate the global structure of the dataset, a possible indicator is the correlation coefficient between the pairwise distances of the data in the original dataspace and the projected data space. Let d_{ij} and $\hat{d}_{ij}$ represent the distance between data points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the original dataspace and projected data space respectively. Due to the dependencies which exist in the d_{ij}, the estimation of the correlation coefficient is biased and thus a method which select a representative independent pairwise distances from the set of pairwise distances was proposed in [41] to reduce this bias. This method was called Natural PCA (NatPCA) and briefly described below.

Let M be a finite set of points and $S \in M$ be a subset. The distance between a point $i \in M$ to the set S is defined as

$$dist(i,S) = \min\{d_{ij}, j \in S\}..$$

If there are several closest points in S to a point i then one is selected randomly. The NatPCA is the $m-1$ (where m is the sample size) pairs of points $\{i, j\} \in M \times M$ selected by the following algorithm:

1) Let S be an empty set.
2) The first component is a pair of the most distant points $\{i_m, j_m\} = \arg\sup_{ij} d_{ij}$. We put i_m and j_m in S.
3) Among all the points which are not in S we select a point k_m which is the most distant to S:

$$k_m = \arg\sup_j \{dist(j,S)\}. \tag{18}$$

4) We define next the 'natural' component as a pair $\{k_m, p_m\}$ where $p_m \in S$ is the point in S closest to k_m. We add k_m to S.
5) We repeat steps 3-4 until all points are in S.

The distances selected by NatPCA algorithm are independent and represent all scales in the distribution of data. This can be sensitive to the presence of outliers in the data and an attempt to resolve this problem was proposed in [41]. To measure

the adequacy of both PCA and MPCA in representing the global structure, we compute the correlation coefficient

$$r = corr(d_{ij}, \hat{d}_{ij}), \{i, j\} \in NatPCA. \quad (18)$$

See table 17 - 19 for the result.

Table 17. Global structure preservation using NatPCA for dataset 1.

	PCA	***Cluster 1***	***Cluster 2***		***Cluster 3***				
Scale	***(0,1)***	***(0.4,1)***	***(0.3,1)***	***(0.1,1)***	***(0,0.1)***	***(0,0.2)***	***(0,0.3)***	***(0,0.9)***	***(0.1,0.9)***
2 PC	0.99	0.98	0.98	0.99	0.89	0.89	0.89	0.89	0.89

Table 18. Global structure preservation using NatPCA for dataset 2.

	PCA	***Cluster 1***			***Cluster 2***		***Cluster 3***	
Scale	***(0,1)***	***(0,0.2)***	***(0.2,0.6))***	***(0.1,0.7)***	***(0,0.1)***	***(0.3,0.6)***	***(0.2,1)***	***(0,0.9)***
4 PC	0.99	0.99	0.99	0.99	0.98	0.99	0.99	0.99

Table 19. Global structure preservation using NatPCA for dataset 3.

	PCA	***Cluster 1***		***Cluster 2***			***Cluster 3***		
Scale	***(0,1)***	***(0.3,1)***	***(0.4,1)***	***(0,0.5)***	***(0.2,1)***	***(0.1,0.8)***	***(0.1,0.2)***	***(0,0.2)***	***(0,0.1)***
4 PC	1.00	0.99	0.99	1.00	0.99	1.00	1.00	1.00	0.99

For the artificial dataset, PCA and in general cluster 1 and 2 perform better in terms of global structure preservation. The result is not distinguishable for the vertebral column dataset (dataset 2) and the breast tissue dataset (dataset 3).

5.10 Data exploration using Local PCA in Dataspace

Approximating a dataset locally using PCA is done by partitioning the dataset such that points in each partition are more homogenous (provided the dataset is well partitioned). Representing the PCA structures of each partition and analysing them give insight into the structure of the dataset. This knowledge can be incorporated into for further analysis of the dataset.

Next, we will analyse the structure of the Iris dataset (see figure 1 for data visualization via PCA) using local PCA. In particular we will look at the local PCA of few data points (including the mean point) of the dataset at various radii of neighbourhood to see how the PCA structure changes. For each target point, we consider the local PCA at various scales of regular interval from 0 to 1. The scale 0 to 1 has been chosen such that the maximum pairwise distance $r_{x_0}^{\max}$ between target point x_0 and other points in the dataset has been scaled to 1. Therefore a given scale s corresponds to $s \times r_{x_0}^{\max}$ radius of neighbourhood. The points selected are given in table 20.

Table 20

Target points	***Variable 1***	***Variable 2***	***Variable 3***	***Variable 4***
A	-1.6223	-1.739	-1.3935	-1.1776
B	2.2422	1.7205	1.667	1.3121
C (mean Point)	0	0	0	0
D	-0.8977	1.7205	-1.2801	-1.1776
E	0.3100	-0.5858	0.1368	0.1328

The target point A and B have been selected because they represent the pair of points with largest distance which can be viewed as the end points of the dataset. The target point C represent the mean point (data have been normalized to have mean 0 and unit variance). Points D and E have been chosen randomly. For all target points and corresponding scales $(\mathbf{x}_i, s)$ the PCA structures were clustered and the result is shown in figure 18. For each target point, the inconsistency coefficients indicate 2 natural clusters of the PCA structures and therefore we show the result of 2 clusters for each target point.

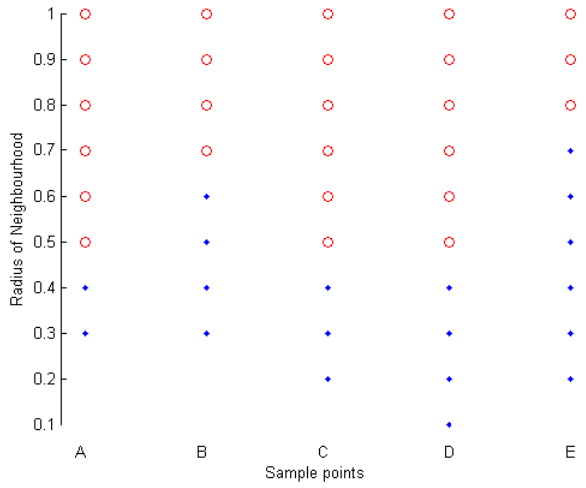


Figure 25. Cluster of local PCA structure of the Iris dataset for the target points in table 20

Note: scales with fewer than 5 data points have been exempted. Also clusters of different target points with similar colour do not imply that the PCA structures are the same. The colours are specific to each target point and only show the natural clusters as we increase the range for the specific target point.

Looking at cluster analysis at target point A and B, we can observe the change in the PCA structure as we navigate from one end of the dataspace to the other end. Also the cluster analysis of at the mean point of the dataset reveals how the PCA structure changes as the radius of neighbourhood increases (i.e. as we move from the center of the dataset towards the end of the dataset).

Finally, we will like to analyse the local PCA structures of the Energy efficiency dataset. We partition the dataset using the recursive local PCA algorithm as discussed in section 4.3. Recall that the algorithm partition a region if it is better approximated by two hyperplanes instead of one based on the criterion given in 4.10. The result of recursive local PCA on the dataset using hyperplane of dimension 3 suggest 6 partitions of the dataspace. The PCA of each partition was computed and represented by ρ_3. We cluster the partition by clustering the PCA structure of each of the regions and the result is given in table

Table 21

Dataset	***Cophenetic***	***Inconsistency***
		0.7071 (4 clusters)
Energy Efficient Dataset	1.0000	0.7071 (3 clusters)
		1.1527 (2 clusters)

The cophenetic coefficient is 1, which indicates that the cluster tree generated by the hierarchical clustering reflects the dissimilarity in the data. The inconsistent coefficient clearly suggests 2 clusters which is supported by the dendrogram. See figure 26.

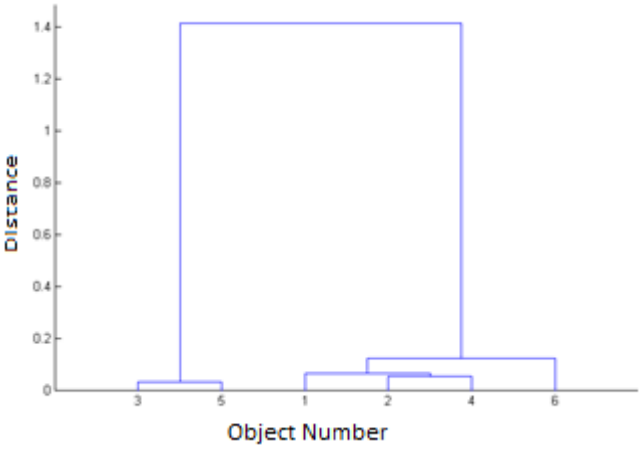


Figure 26. Cluster of local PCA structure of the Energy Efficiency Dataset partitioned using recursive local PCA.

We therefore see that even though the dataset is better approximated by six hyperplanes rather than one, two of the regions share similar PCA structure and the remaining four also share similar PCA structure and the two PCA are quite dissimilar.

5.11 Discussion and Conclusion

In this chapter we explore the datasets using various localizations. The first three datasets were analysed using MPCA, and the remaining two datasets were analysed using local PCA in space. We further clustered the local PCA structures using agglomerative hierarchical clustering. Cluster selection was made based on the Inconsistency coefficient taking into consideration the cophenetic coefficient (which indicates how well the cluster tree generated by the hierarchical clustering reflects the dissimilarity in the data) and the visual inspection of the dendrogram. For each cluster analysed using MPCA we selected representative scales based on the ratio of distortion and also the medoid point for further analysis. For each representative scale we further analysed the data to understand the distortion in the dataset, how well the approximating subspace preserves both local and global structures and how efficient the approximating subspace preserves class structures of the dataset.

For dataset with clear multiscale structure like dataset 1, MPCA analysis revealed the structures of the data, in addition we found a PCA structure (cluster 4) with better approximating subspace than PCA in terms of ratio of distortion and preservation of local structure. We particularly remark that for the vertebrae column dataset (dataset 2) the approximating suspace given by MPCA at scale (0,0.1) have a better result than PCA and any other scale in terms of preservation of local structure and class

compactment. Though the result of hierarchical clustering of the MPCA structures of the breast tissue dataset (dataset 3) does not indicate clear clusters in the PCA (see figure 22), cluster 3 has minimal ratio of distortion and better class compactment for some classes (see appendix J-O).

Finally we analysed 2 dataset using localization in space. For the Iris dataset (dataset 4), we analysed the local PCA structures at varying radius of neighbourhood for a given target point. We studied the PCA structures of the dataset as we move from the mean point to the end of the dataset in the dataspace; this particularly revealed that there are two distinct PCA structures in the dataset. Also we analysed the energy efficiency dataset using the recursive local PCA algorithm, this revealed that the dataset is better approximated by six hyperplanes rather than a single hyperplane (PCA). We also clustered the local PCA structures of the six partitions which revealed two distinct clusters of the local PCA structures. This implies that certain partitions of the dataspace have similar PCA.

Chapter 6

Conclusion

The first part of this thesis is an empirical study to investigate the effect of initialization on manifold modelling methods for two popular initialization methods using SOM as a case study. The two initialization methods studied were random initialization (in which the initial weights are selected randomly from the dataspace) and principal component initialization (in which the initial weights are selected from the space of the principal components of the data). To ensure an exhaustive study, synthethic data sets distributed along various shapes of only 2-dimensions were considered and the map was 1-dimensional. 1-dimensional SOM is important, for example, for approximation of principal curves. To marginalise other factors that could influence the result, SOM learnings were subject to the same initial conditions such as neighbourhood function and learning rate. To marginalize the effect which the sequence of training vectors could have on the study, the batch algorithm was adopted.

To further understand the dynamics of manifold learning, we propose a classification of the dataset into linear, quasilinear and nonlinear classes. Quasilinearity was defined using the principal curve. The fraction of variance unexplained (FVU) was used as the criterion to evaluate the quality of learning of the final map for the two initialization methods. For random initialization (RI) methods, a sample of 100 initial configurations from the space of possible initial configurations was selected for each dataset. The probability distribution of the FVU were drawn and analysed for each dataset studied.

The results of our case study show that RI tends to perform quite well for nonlinear datasets. In general, we can conclude from the study that PCI performs not better or even worse than the median of RI for the datasets that we studied. The performance of RI remains inconclusive for quasilinear datasets. Furthermore, the result shows that the presence of noise has significant influence on the performance of PCI SOM, for example, the good performance of PCI for spiral dataset at node 50 was destroyed by noise.

The result from this case study demonstrates that the widely accepted presumption about advantages of PCI SOM is not universal. We showed that statistically three random initializations are sufficient to obtain SOM with FVU less or equal to PCI with probability 0.95, and for quasilinear dataset, five RI SOM is sufficient to obtain SOM with FVU less than or equal to PCI with probability 0.90 for the datasets we studied.

In addition, the classification of data into quasilinear and nonlinear classes has been important for understanding the dynamics of manifold learning and selection of initial approximation. The results based on our classification of datasets show that the optimal choice of initial weights for SOM depends on the geometry of the dataset and in particular, randomisation of initial weights can help in manifold learning.

Further areas of research will be to extend this study to data with higher dimension than two (this has been done but not included in this thesis) and also SOM with dimension higher than one. Since SOM is often considered as an approximation of the principal manifold, it is desirable to consider the quasilinearity of datasets without the concept of principal manifold. Developing such will provide apriori knowledge that can guide in the choice of initial approximation for manifold modelling methods.

The second part of this thesis sets out to develop a method to investigate and reveal intrinsic structure(s) in data and to identify regions within a given data space that have similar intrinsic structure thereby providing additional tool for data exploration and understanding.

To achieve this, we developed the multiscale principal component analysis (MPCA) algorithm. This algorithm is based on weighted PCA and a generalization of the classical PCA as stated in definition four. Though similar to [71], however, the weights in MPCA are chosen from the distribution of the pairwise distances of datapoints. This choice of weights restricts the analysis to include only pairwise distances of interest, allowing us to study the changes in PCA structures of the data for the various distributions of pairwise distances of the dataset and therefore help us to understand the pairwise distance structure of the dataset at various scales. In order words we can study the changes in the subspaces that approximate the data for various restricted pairwise distance structures of the dataset. To reveal the intrinsic structure(s) that may

be present in a dataset, we analyse the resulting PCA structures resulting from the MPCA of the dataset.

MPCA structures are scale dependent and to analyse the MPCA structures of the data, we studied the principal components as points in the real projective space or in general, the Grassmanian space. To fully understand this, we embedded points in the Grassmanian space (our principal component in this case) in a suitable vector space with similar topological properties. We chose the space of orthogonal projection matrix because it is homeomorphic to the Grassmannian space and also meaningful to principal components. The projection matrix corresponding to a point (principal component) can be seen as the projector matrix which maps the data from the dataspace to the subspace of the principal components. The properties of principal components were compared with the properties of orthogonal projection matrix for consistency and to ensure that desired properties are preserved. Therefore for a given scale, we represented the PCA structure by a sequence of projector matrices which maps the data to the corresponding sequence of subspaces. Cluster analysis of the MPCA structures (represented by the corresponding orthogonal projection matrices) group together scales in the data with similar structures and separate scales with dissimilar structures thereby revealing the various classes of structures in the dataset. Each cluster is taken to represent a structure in the dataset.

We evaluated the quality of approximation of MPCA and PCA in terms of data distortion using the ratio of distortion which we defined in chapter three. Other measures include: preservation of local distance structure, preservation of global distance structure and preservation of class compactness.

MPCA reveals some intrinsic structures of the data which the classical PCA might not reveal because classical PCA gives the global structure of the data. In particular for data with clear multiscale structures, MPCA was able to reveal such structures. We remark here that even for dataset without clear multiscale structure(s), we still found approximations from MPCA which have lower data distortion compared to PCA for all the datasets we analysed. Being able to identify the intrinsic structures of a data is useful for taking decision about the choice of subspace to approximate the data in order to preserve certain properties of the data which can help improve the performance of

subsequent analyses. For example, an intrinsic structure which preserves the local pairwise distance of the data projection may be preferable if we intend to run a nearest neighbour algorithm on the data approximation or if we want to mitigate against the effect of outliers.

The results of application of MPCA to various artificial and real datasets were also presented. This revealed various structures of the datasets. We discover that for some datasets, approximation using PCA distort the structure of data that exist at smaller scales (pairwise distances) while MPCA preserves the local structure of the data better than PCA.

In addition to the above, this thesis also extended the representation of local PCA structures in scale (discussed in chapter three) to local PCA in dataspace (based on local PCA definition as used in [14, 29]). In order to use the MPCA algorithm developed in chapter three for local PCA, we formulated the local PCA problem based on definition four and solved as a weighted PCA as shown in chapter four, and we were able to introduce localization using kernel function.

One of the reasons for the development of nonlinear PCA and other nonlinear data approximation techniques is the fact that PCA does not "efficiently" approximate complex datasets (for example, data that can be termed as curved, disconnected, or branched) as such complex datasets are often characterised by different structures at various region of the dataspace. Partitioning the dataspace into regions and performing PCA on each region can be seen as approximation of nonlinear technique using hyperplanes. The local PCA around a given target point in a dataset is the PCA of the set of datapoints within a given radius of the target point.

We study the intrinsic structures of a dataset in the dataspace by selecting various target points from the data and analysing the local PCA structures for each of these target points. The resulting local PCA structures depend on the target points and radius of neighbourhood. We represented these local PCA structures using orthogonal projection matrix as discussed in chapter three and cluster analysis of the local PCA structure separate dissimilar structures and groups together similar structures thereby revealing the various classes of structures in the dataset. Each cluster represents a structure in the dataset.

Our proposed analysis of the local PCA structure of data in the dataspace provides useful insight to the geometry of the data as we can study the changes in the local PCA structures over the data space and relate this with the global PCA structure. Also we can identify regions in the dataspace with similar PCA structure providing useful insight for data exploration. Examples of applications of local PCA (localization in the dataspace) were given in chapter five.

Sometimes it is desirable to approximate data locally using a linear method such as PCA. However the result of PCA localization in dataspace is distorted by a few influential datapoints (see figure 19). We proposed and demonstrated that the two localization approaches (see section 4.5) can be combined to improve the robustness of the linear method for approximating data locally.

We must however state that the result of MPCA and local PCA analysis of a dataset is not invariant to data scaling; different results will be obtained for various normalizations of the datset. This applies generally to all methods that model the topology of dataset. We should also state that the quality of local PCA depends on the quality of cluster analysis or partitioning of the dataset.

PCA localization either in scale or space involves the exclusion of either some datapoints (as in the case of local PCA) or exclusion of some pairwise distances of data projection (as in the case of MPCA). There is a risk of overfitting the data; especially in the case where we seek an approximation to a dataset that preserves certain internal structure(s) using MPCA. The exclusion of large datapoints (or pairwise distances) may lead to the loss of "information" similar to the idea of "uncertainty principle". Therefore an area for further research is that of developing criteria to limit the percentage of datapoints (or pairwise distances) that can be exempted in the analysis of principal component without losing relevant information.

It should be mentioned that in predictive modelling, it is possible that structures whose approximation is considered to result in overfitting of the dataset (or which we classify as noise) sometimes can have better predictive performance. This research can be further extended by developing methods which seeks intrinsic structure(s) (which could be present in a dataset) with better predictive performance than PCA. One suggestion to developing such method will be to find intrinsic structure, taken into

consideration the feature we want to predict. This will be a “partial least square” like method.

Most statistical methods which involve the use of distances of datapoints suffer terribly in high dimension; therefore we propose that MPCA should only be applied to data for which PCA can be applied without suffering from the “curse of dimensionality”.

Also, as functional principal component analysis is becoming popular, another area for further research will be to extend the idea of multiscale PCA and clustering of scale to functional principal component analysis.

In conclusion, the first part of this thesis addressed the problem of initialization of manifold modelling methods for two initialization methods. Using various examples, we demonstrated that the widely accepted assumption of the advantages of PCI SOM over RI is not universal and for SOM this assumption is essentially wrong for nonlinear datasets. We also showed that data classification is important for understanding the dynamics of manifold learning.

In the second part of this thesis, we demonstrated that analysing local PCA structures of a dataset can be used to reveal some intrinsic structure(s) that PCA might not reveal. This provides useful insight into the structure(s) of the dataset and also provides robust linear approximations of the data. We analysed the local PCA structures of data for various distributions of weight which we called Multiscale PCA. Cluster analysis of these local PCA structures reveals some intrinsic structures of the data. From the result of various examples, we found that we can sometimes find subspaces which approximate data better than PCA in terms of minimal data distortion, local structure preservation, class structure preservation and minimizing the effect of some influential data points.

Finally, we also analysed the local PCA structure of various partitions of data in the dataspace. Analysing the local PCA structures identifies the various regions with similar structures and also reveals the changes in the structure of the data as we move along the dataspace. Local PCA in dataspace provides a linear approximation for non linear methods and in particular it is able to efficiently approximate data with disconnected regions. We demonstrated that the combination of the two localization

methods discussed in this thesis provides a more robust linear method for approximation of data.

Appendix

Table A. The distribution of FVU for RI. (The last column is FVU for PCI)

Dataset	*k*	*Mean*	*Standard Deviation*	*Minimum*	*Maximum*	*PCI*
Spiral	10	13.81	2.117	11.38	20.37	11.43
	20	3.31	0.93	1.98	6.25	3.26
	50	0.44	0.19	0.12	6.25	0.13
Spiral With Noise	10	19.24	1.84	7.47	24.74	18.52
	20	6.92	0.93	5.95	11.01	8.65
	50	2.4	0.22	1.81	2.87	2.61
Horse shoeshoe	10	16.51	2.86	12.03	26.43	17.12
	20	3.55	1.33	1.81	7.23	5.99
	50	0.35	0.16	0.11	0.993	1.32
Horse shoe with	10	16.52	2.45	12.76	23.23	18.27
	20	4.6	1.09	2.93	7.32	6.15
	50	1.31	0.15	0.59	1.73	1.91
	100	0.72	0.06	0.57	0.88	0.86
S Shape	10	12.89	0.54	12.73	15.38	12.76
	20	3.96	0.84	2.34	6.42	2.37
	50	0.73	0.25	0.19	1.41	0.35
S Shape with Noi	10	13.04	0.01	13.03	13.05	13.03
	20	3.91	0.87	2.51	5.99	2.52
	50	0.78	0.24	0.41	1.35	0.46
C Shape	10	4.28	0.07	4.22	4.35	4.35
	20	1.19	0.48	0.75	2.9	0.88
	30	0.53	0.19	0.21	1.24	0.31
C shape with	10	11.41	3.05	9.7	21.94	9.78
	20	4.04	0.67	3.08	6.56	3.13
	30	2.02	0.15	1.66	2.4	2.07

Table B. The proportion estimate of RI which performs better than PCI for various datasets and number of nodes

Dataset	*k*	*% better than PCI*	*Confidence Interval (%) at confidence level of 95%*	*Classification*
Spiral	10	41%	31.86 - 50.80	Nonlinear
	20	55%	45.24 - 64.39	Nonlinear
	50	1%	0 - 1.96	Nonlinear
Spiral With Noise	10	49%	39.42 - 58.65	Nonlinear
	20	95%	88.54 - 98.13	Nonlinear
	50	84%	75.47 - 90.01	Nonlinear
Horse shoe	10	73%	65.53 - 80.77	Nonlinear
	20	95%	88.54 - 98.13	Nonlinear
	50	100%	99.02 - 100	Nonlinear
Horse shoe with	10	74%	64.58 - 81.64	Nonlinear
	20	89%	81.21 - 93.91	Nonlinear
	50	100%	99.02 - 100	Nonlinear
	100	99%		Nonlinear
S Shape	10	36%	27.26 - 45.78	Quasi-linear
	20	7%	3.20 - 13.98	Quasi-linear
	50	7%	3.20 - 13.98	Quasi-linear
S Shape with	10	48%	38.46 – 57.68	Quasilinear
	20	9%	4.62 – 16.42	Quasilinear
	50	11%	6.09 - 18.79	Quasilinear
C Shape	10	100%	99.02-100	Quasilinear
	20	33%	24.54 - 42.72	Quasilinear
	30	13%	7.62 - 21.12	Quasilinear
C shape with	10	73%	65.53 - 80.77	Quasilinear
	20	8%	3.90 – 15.21	Quasilinear
	30	72%	62.48 – 79.90	Quasilinear

Table C. Number of unique final configurations in the datasets and the relative performance of RI versus PCI for quasi-linear datasets.

% of PCI greater than	*Unique final*
100.00%	2
33.00%	34
13.00%	47
73.00%	11
8.00%	50
72.00%	47
36.00%	11
7.00%	67
7.00%	65
48.00%	2
9.00%	49
11.00%	58

Table D. The angles between original vector and 1st principal axis at different scales for example 3.

Scale		Upper Limit									
		1.0	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**	**0.4**	**0.3**	**0.2**	**0.1**
Lower Limit	**0.0**	85.25	6.65	6.65	6.65	6.65	6.65	6.65	6.45	8.77	14.92
	0.1	85.29	6.59	6.59	6.59	6.59	6.59	6.59	6.37	8.54	0.00
	0.2	85.62	6.12	6.12	6.12	6.12	6.12	6.12	5.61	0.00	0.00
	0.3	86.09	7.20	7.20	7.20	7.20	7.20	7.20	0.00	0.00	0.00
	0.4	86.27	90.00	90.00	90.00	90.00	90.00	0.00	0.00	0.00	0.00
	0.5	86.27	90.00	90.00	90.00	90.00	0.00	0.00	0.00	0.00	0.00
	0.6	86.27	90.00	90.00	90.00	0.00	0.00	0.00	0.00	0.00	0.00
	0.7	86.27	90.00	90.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	0.8	86.27	90.00	0.00	0.0	0.00	0.00	0.00	0.00	0.00	0.00
	0.9	86.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Note: The MPCA at scale 0-1 is the same as PCA. The cell for PCA as being marked with a grey-scale background

Table E. The ratio of distortion on example 3 data at different scales for k = 2.

Scale		Upper Limit									
		1.0	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**	**0.4**	**0.3**	**0.2**	**0.1**
Lower Limit	**0.0**	0.903	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	0.1	0.920	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.000
	0.2	0.973	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.000	0.000
	0.3	0.991	1.000	1.000	1.000	1.000	1.000	1.000	0.000	0.000	0.000
	0.4	0.997	NaN	NaN	NaN	NaN	NaN	0.000	0.000	0.000	0.000
	0.5	0.997	NaN	NaN	NaN	NaN	0.000	0.000	0.000	0.000	0.000
	0.6	0.997	NaN	NaN	NaN	0.000	0.000	0.000	0.000	0.000	0.000
	0.7	0.997	NaN	NaN	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.8	0.997	NaN	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.9	0.997	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Note: The MPCA at scale 0-1 is the same as PCA. The cell for PCA as being marked with a grey-scale background

Table F. The ratio of distortion on example 3 data at different scale for k = 1

SCALE		Upper Limit									
		1.0	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**	**0.4**	**0.3**	**0.2**	**0.1**
Lower Limit	**0.0**	0.470	0.830	0.830	0.830	0.830	0.830	0.830	0.814	0.743	0.675
	0.1	0.498	0.851	0.851	0.851	0.851	0.851	0.851	0.836	0.763	0.000
	0.2	0.646	0.934	0.934	0.934	0.934	0.934	0.934	0.928	0.000	0.000
	0.3	0.867	0.953	0.953	0.953	0.953	0.953	0.953	0.000	0.000	0.000
	0.4	0.985	NaN	NaN	NaN	NaN	NaN	0.000	0.000	0.000	0.000
	0.5	0.985	NaN	NaN	NaN	NaN	0.000	0.000	0.000	0.000	0.000
	0.6	0.985	NaN	NaN	NaN	0.000	0.000	0.000	0.000	0.000	0.000
	0.7	0.985	NaN	NaN	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.8	0.985	NaN	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.9	0.985	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Note: The MPCA at scale 0-1 is the same as PCA. The cell for PCA as being marked with a grey-scale background

Table G. The percentage of pairwise distances exempted in computing MPCA at various scales for example 3 data.

SCALE		Upper Limit									
		1	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**	**0.4**	**0.3**	**0.2**	**0.1**
Lower Limit	**0**	0%	11%	11%	11%	11%	11%	11%	16%	35%	74%
	0.1	26%	38%	38%	38%	38%	38%	38%	42%	61%	0%
	0.2	65%	77%	77%	77%	77%	77%	77%	81%	0%	0%
	0.3	84%	95%	95%	95%	95%	95%	95%	0%	0%	0%
	0.4	89%	100%	100%	100%	100%	100%	0%	0%	0%	0%
	0.5	89%	100%	100%	100%	100%	0%	0%	0%	0%	0%
	0.6	89%	100%	100%	100%	0%	0%	0%	0%	0%	0%
	0.7	89%	100%	100%	0%	0%	0%	0%	0%	0%	0%
	0.8	89%	100%	0%	0%	0%	0%	0%	0%	0%	0%
	0.9	89%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Note: The MPCA at scale 0-1 is the same as PCA.

Table H. *Result for Vertebral Column with data labelled to 3 Classes. Class* compactment result for class *"Hernia"*

	PCA	***Cluster 1***			***Cluster 2***		***Cluster 3***	
Scale / *knn*	***(0,1)***	***(0,0.2)***	***(0.2,0.6))***	***(0.1,0.7)***	***(0,0.1)***	***(0.3,0.6)***	***(0.2,1)***	***(0,0.9)***
$k = 3$	0.79	0.89	0.90	0.88	0.94	0.91	0.80	0.79
$k = 5$	0.84	0.90	0.92	0.90	0.93	0.90	0.85	0.84
$k = 10$	0.90	0.94	0.93	0.93	0.97	0.93	0.89	0.90

Table I. *Result for Vertebral Column with data labelled to 3 Classes. Class* compactment result for class *"Spondylolisthesis"*

	PCA	***Cluster 1***			***Cluster 2***		***Cluster 3***	
Scale / *knn*	***(0,1)***	***(0,0.2)***	***(0.2,0.6)***	***(0.1,0.7)***	***(0,0.1)***	***(0.3,0.6)***	***(0.2,1)***	***(0,0.9)***
k = 3	0.74	0.77	0.79	0.77	0.82	0.81	0.73	0.73
k = 5	0.76	0.80	0.82	0.80	0.83	0.82	0.77	0.75
k = 10	0.78	0.81	0.83	0.82	0.84	0.85	0.79	0.78

Table J. *Result for Breast tissue dataset. Class* compactment result for class *"Carcinoma"*

	PCA	***Cluster 1***		***Cluster 2***			***Cluster 3***		
Scale / *knn*	***(0,1)***	***(0.3,1)***	***(0.4,1)***	***(0,0.5)***	***(0.2,1)***	***(0.1,0.8)***	***(0.1,0.2)***	***(0,0.2)***	***(0,0.1)***
k = 3	0.79	0.79	0.75	0.83	0.78	0.81	0.81	0.83	0.79
k = 5	0.81	0.81	0.78	0.77	0.80	0.77	0.90	0.90	0.90
k = 10	0.90	0.89	0.87	0.90	0.90	0.90	0.98	0.97	0.97

Table K. *Result for Breast tissue dataset. Class* compactment result for class *"Fibro-adenoma".*

	PCA	***Cluster 1***		***Cluster 2***			***Cluster 3***		
Scale / *knn*	***(0,1)***	***(0.3,1)***	***(0.4,1)***	***(0,0.5)***	***(0.2,1)***	***(0.1,0.8)***	***(0.1,0.2)***	***(0,0.2)***	***(0,0.1)***
k = 3	0.89	0.87	0.84	0.82	0.84	0.80	0.98	0.98	0.98
k = 5	0.88	0.83	0.83	0.81	0.83	0.80	1.00	1.00	1.00
k = 10	0.91	0.89	0.88	0.91	0.91	0.90	0.99	0.99	0.99

Table L. *Result for Breast tissue dataset. Class* compactment result for class "*Mastopathy*".

	PCA	***Cluster 1***		***Cluster 2***			***Cluster 3***		
Scale / *knn*	***(0,1)***	***(0.3,1)***	***(0.4,1)***	***(0,0.5)***	***(0.2,1)***	***(0.1,0.8)***	***(0.1,0.2)***	***(0,0.2)***	***(0,0.1)***
k = 3	0.83	0.89	0.89	0.81	0.83	0.81	0.91	0.91	0.89
k = 5	0.90	0.87	0.84	0.84	0.87	0.83	0.96	0.96	0.98
k = 10	0.95	0.94	0.94	0.95	0.94	0.95	1.00	1.00	0.99

Table M. *Result for Breast tissue dataset. Class* compactment result for class "*Glandular*".

	PCA	***Cluster 1***		***Cluster 2***			***Cluster 3***		
Scale / *knn*	***(0,1)***	***(0.3,1)***	***(0.4,1)***	***(0,0.5)***	***(0.2,1)***	***(0.1,0.8)***	***(0.1,0.2)***	***(0,0.2)***	***(0,0.1)***
k = 3	0.89	0.87	0.77	0.89	0.89	0.89	0.94	0.94	0.96
k = 5	0.88	0.90	0.91	0.88	0.88	0.88	0.94	0.94	0.93
k = 10	0.99	0.98	0.96	0.99	0.99	0.99	0.99	0.99	1.00

Table N. *Result for Breast tissue dataset. Class* compactment result for class "*Connective*".

	PCA	***Cluster 1***		***Cluster 2***			***Cluster 3***		
Scale / *knn*	***(0,1)***	***(0.3,1)***	***(0.4,1)***	***(0,0.5)***	***(0.2,1)***	***(0.1,0.8)***	***(0.1,0.2)***	***(0,0.2)***	***(0,0.1)***
k = 3	0.81	0.79	0.74	0.86	0.81	0.86	0.83	0.86	0.86
k = 5	0.91	0.90	0.87	0.96	0.81	0.86	0.83	0.86	0.86
k = 10	0.93	0.91	0.91	0.94	0.93	0.96	0.86	0.87	0.89

Table O. *Result for Breast tissue dataset. Class* compactment result for class "*Adipose".*

	PCA	***Cluster 1***		***Cluster 2***			***Cluster 3***		
Scale / *knn*	***(0,1)***	***(0.3,1)***	***(0.4,1)***	***(0,0.5)***	***(0.2,1)***	***(0.1,0.8)***	***(0.1,0.2)***	***(0,0.2)***	***(0,0.1)***
k = 3	0.80	0.80	0.82	0.80	0.79	0.80	0.86	0.86	0.83
k = 5	0.86	0.86	0.88	0.85	0.85	0.85	0.85	0.85	0.81
k = 10	0.92	0.93	0.92	0.93	0.92	0.93	0.88	0.88	0.87

Bibliography

[1] Abdi, H. (2007). Singular Value Decomposition (SVD) and Generalized Singular Value Decomposition. *Encyclopedia of measurement and statistics*, 907-912.

[2] Akinduko, A. A., & Gorban, A. N. (2014, March). Multiscale principal component analysis. *In Journal of Physics: Conference Series* (Vol. 490, No. 1, p. 012081). IOP Publishing.

[3] Akinduko, A. A., & Mirkes, E. M. (2012). Initialization of Self-Organizing Maps: Principal Components versus Random Initialization. A Case Study. arXiv preprint arXiv:1210.5873.

[4] Anderson, E. (1936). The species problem in Iris. *Annals of the Missouri Botanical Garden,* 457-509.

[5] Anderson, T. W. (1963). Asymptotic theory for principal component analysis. *Annals of Mathematical Statistics,* 122-148.

[6] Arnold, R., & Jupp, P. E. (2013). Statistics of orthogonal axial frames. *Biometrika*, 100(3), 571-586.

[7] Attik, M., Bougrain, L., & Alexandre, F. (2005). Self-organizing map initialization. *In Artificial Neural Networks: Biological Inspirations–ICANN* 2005 (pp. 357-362). Springer Berlin Heidelberg.

[8] Baldi, P. F., & Hornik, K. (1995). Learning in linear neural networks: *A survey. Neural Networks, IEEE Transactions on,* 6(4), 837-858.

[9] Banfield, J. D., & Raftery, A. E. (1992). Ice floe identification in satellite images using mathematical morphology and clustering about principal curves. *Journal of the American Statistical Association,* 87(417), 7-16.

[10] Bellman, R.E. (1961). Adaptive control processes: a guided tour (Vol. 4). Princeton: Princeton University Press. 94–100

[11] Beyer, M. A., & Laney, D. (2012). The importance of 'big data': a definition. Stamford, CT: Gartner.

[12] Bishop, C. M., Svensén, M., & Williams, C. K. (1998). GTM: The generative topographic mapping. *Neural computation*, 10(1), 215-234.

[13] Bohn, R. E., & Short, J. E. (2009). How much information. 2009 report on American consumers. University of California, San Diego, Global Information Industry Centre.

[14] Braverman, E.M. (1970). Methods of extremal grouping of parameters and problem of apportionment of essential factors. *Automation and Remote Control*, 31 (1), 108–116

[15] Bullinaria, J. A. (2004). Self Organizing Maps: Fundamentals. Introduction to Neural networks: Lecture note, University of Birmingham

[16] Burges C. J. C. (2010) Geometric Methods for Feature Extraction and Dimensional Reduction - A Guided Tour Data Mining and Knowledge Discovery Handbook (New York: Springer) ed O Maimon and L Rokach . 2nd Edition ISBN 978-0-387-09822-7 pp 53-82.

[17] Cochran, R. N., & Horne, F. H. (1977). Statistically weighted principal component analysis of rapid scanning wavelength kinetics experiments. *Analytical Chemistry*, 49(6), 846-853.

[18] da Costa, J. F.P, Alonso, H., & Roque, L. (2011). A weighted principal component analysis and its application to gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB),* 8(1), 246-252.

[19] De Lathauwer, L., De Moor, B., & Vandewalle, J. (2000). A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4), 1253-1278.

[20] Delicado, P. (2001). Another look at principal curves and surfaces. *Journal of Multivariate Analysis*, 77(1), 84-116.

[21] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1-38.

[22] Diamantaras, K. I., & Kung, S. Y. (1996). Principal component neural networks: theory and applications. John Wiley & Sons, Inc.

[23] Efron, B. (1967). The two sample problem with censored data. *Proc. 5th Berkeley Sympos. Math. Statist. Prob.*", Prentice-Hall: New York.

[24] Einbeck, J., Evers, L., & Bailer-Jones, C. (2008). Representing complex data using localized principal components with application to astronomical data. In Principal Manifolds for Data Visualization and Dimension Reduction (pp. 178-201). Springer Berlin Heidelberg.

[25] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2), 179-188.

[26] Fort, J. C., Letremy, P., & Cottrell, M. (2002). Advantages and drawbacks of the Batch Kohonen algorithm. *In ESANN* (Vol. 2, pp. 223-230).

[27] Fort, J. C., Cottrell, M., & Letremy, P. (2001). Stochastic on-line algorithm versus batch algorithm for quantization and self organizing maps. *In Neural Networks for Signal Processing XI, 2001. Proceedings of the 2001 IEEE Signal Processing Society Workshop* (pp. 43-52). IEEE.

[28] Fréchet M. (1948). Les element aléqatoires de nature quelconque dans un espace distancié. *Ann. Inst. H. Poincaré 10, 215-310*

[29] Fukunaga, K., & Olsen, D. R. (1971). An algorithm for finding intrinsic dimensionality of data. *Computers, IEEE Transactions on,* 100(2), 176-183.

[30] Gabriel, K. R., & Zamir, S. (1979). Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics*, 21(4), 489-498.

[31] Gantz, J., & Reinsel, D. (2009). As the economy contracts, the digital universe expands. IDC Multimedia White Paper.

[32] Gantz, John F., David Reinsel, Christopher Chute, Wolfgang Schlichting, John McArthur, Stephen Minton, Irida Xheneti, Anna Toncheva, and Alex Manfrediz, (2007). "The expanding digital universe," IDC white paper, sponsored by EMC.

[33] Gantz, John F., Christopher Chute, Alex Manfrediz, Stephen Minton, David Reinsel, Wolfgang Schlichting, and Anna Toncheva, (2008). "The diverse and expanding digital universe," IDC white paper, sponsored by EMC.

[34] Gantz, J., & Reinsel, D. (2010). The digital universe decade-are you ready. External publication of IDC (Analyse the Future) information and data, 1-16.

[35] Gerbrands, J. J. (1981). On the relationships between SVD, KLT and PCA. *Pattern recognition,* 14(1), 375-381.

[36] Gifi, A. (1990). Nonlinear multivariate analysis. John Wiley and Sons. Chichester, England.

[37] Girshick, M. A. (1936). Principal components*. Journal of the American Statistical Association,* 31(195), 519-528.

[38] Girshick, M. A. (1939). On the sampling theory of roots of determinantal equations. *The Annals of Mathematical Statistics*, 10(3), 203-224.

[39]Gnanadesikan, R. (1977) . Methods for statistical Data Analysis of Multivariate Observations. New York: Wiley

[40] Gorban, A. N., & Zinovyev, A, Y, (2009). Principal Graphs and Manifolds Handbook of Research on Machine Learning, Applications and Trends: Algorithms, Methods and Techniques. *Information Science Reference*.(Preprint arXiv:0809.0490v2)

[41] Gorban, A. N., & Zinovyev, A. Y. (2008). Elastic maps and nets for approximating principal manifolds and their application to microarray data visualization. In Principal

Manifolds for Data Visualization and Dimension Reduction (pp. 96-130). Springer Berlin Heidelberg. arXiv:0801.0168 [physics.data-an]

[42] Gorban, A. N., Pitenko, A. A., Zinov'ev, A. Y., & Wunsch, D. C. (2001). Vizualization of any data using elastic map method. *Smart Engineering System Design*, 11, 363-368.

[43] Gorban, A. N., Rossiev, A. A., & Wunsch II, D. C. (1999). Neural network modeling of data with gaps: method of principal curves, Carleman's formula, and other, Lecture given at the USA–NIS Neurocomputing opportunities workshop, Washington DC, (Associated with IJCNN'99). Preprint online: http://arXiv.org/abs/cond-mat/0305508.

[44] Gorban, A.N. & Zinovyev, A.Y. (2001). Visualization of data by method of elastic maps and its applications in genomics, economics and sociology. Preprint of Institut des Hautes Etudes Scientiques, M/01/36,.
http://www.ihes.fr/PREPRINTS/M01/Resu/resu-M01-36.html

[45] Gorban, A. N., Zinovyev, A. Y., & Wunsch, D. C. (2003, July). Application of the method of elastic maps in analysis of genetic texts. *In Neural Networks, 2003. Proceedings of the International Joint Conference on* (Vol. 3, pp. 1826-1831). IEEE.

[46] Gorban, A. N., Sumner, N. R., & Zinovyev, A. Y. (2007). Topological grammars for data approximation. *Applied Mathematics Letters,* 20(4), 382-386.

[47] Gorban A., Kégl B., Wunsch D., Zinovyev A. (Ed.) (2008). Principal Manifolds for Data Visualization and Dimension Reduction. *Lecture Notes in Computational Science and Engineering,* Vol. 58: Berlin-Heidelberg, Springer.

[48] Gorban, A.N., & Yablonsky, G.S. (2013) Grasping Complexity, Computers & Mathematics with Applications, Volume 65, Issue 10, May 2013, Pages 1421-1426, ISSN 0898-1221, http://dx.doi.org/10.1016/j.camwa.2013.04.023.

[49] Greenacre, M.J. (1984). Theory and Applications of Correspondence Analysis. London: Academic Press

[50] Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1). Springer, Berlin: Springer series in statistics.

[51] Hastie, T., & Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association,* 84(406), 502-516.

[52] He, X., Cai, D., & Niyogi, P. (2005). Tensor subspace analysis. *In Advances in neural information processing systems* (pp. 499-506).

[53] Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), 55-67.

[54] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6), 417.

[55] Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 321-377.

[56] Huber, P. J. Robust statistics. 1981. (Republished in paperback, 2004).

[57] Hubert, M., & Engelen, S. (2004). Robust PCA and classification in biosciences. *Bioinformatics*, 20(11), 1728-1736.

[58] Jolliffe, I. (2002). Principal component analysis. John Wiley & Sons, Ltd.

[59] Jolliffe, I. T. (1982). A note on the Use of Principal Components in Regression. *Journal of the Royal Statistical Society, Series C* 31 (3): 300–303. doi:10.2307/2348005.

[60] Kambhatla, N., & Leen, T. K. (1997). Dimension reduction by local principal component analysis. *Neural Computation,* 9(7), 1493-1516.

[61] Karhunen K. (1946). Zur Spektraltheorie Stochastischer Prozesse. *Ann.Acad. Sci. Fennicae*, 37.

[62] Kazmierczak, J.B. (1985). Analyse logarithmique deux exemples d'application. *Rev. Statistique Appliquée*, 33 , 13-24.

[63] Kégl, B., Krzyzak, A., Linder, T., & Zeger, K. (2000). Learning and design of principal curves. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(3), 281-297.

[64] Kégl, B., & Krzyzak, A. (2002). Piecewise linear skeletonization using principal curves. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(1), 59-74.

[65] King, I., & Xu, L. (1997). Localized principal component analysis learning for face feature extraction and recognition. *In Proceedings to the Workshop on 3D Computer Vision* (pp. 124-128).

[66] Kohonen, T., & Honkela, T. (2007). Kohonen network. *Scholarpedia*, 2(1), 1568.

[67] Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological cybernetics,* 43(1), 59-69.

[68] Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., & Saarela, A. (2000). Self organization of a massive document collection. *Neural Networks, IEEE Transactions on*, 11(3), 574-585.

[69] Kohonen, T. (1998). The self-organizing map. *Neurocomputing,* 21(1), 1-6.

[70] Kohonen, T. (2012). Self-organization and associative memory (Vol. 8). Springer.

[71] Koren, Y., & Carmel, L. (2004). Robust linear dimensionality reduction. *Visualization and Computer Graphics, IEEE Transactions on*, 10(4), 459-470.

[72] Kruger, U., Zhang, J., Xie, L. (2008). Development and Applications of Nonlinear Principal Component Analysis – a Review. In: Principal Manifolds for Data Visualization and Dimension Reduction: LNCSE, Gorban, A.N., Kegl, B., Wunsch, D.C., Zinovyev, A.Y. (eds.), vol. 58, pp. 1-43. Springer, Heidelberg

[73] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science

[74] Linting, M., Meulman, J. J., Groenen, P. J., & van der Koojj, A. J. (2007). Nonlinear principal components analysis: introduction and application. *Psychological methods*, 12(3), 336.

[75] Loeve, M. (1970). Probability Theory. 1955. Princeton NJ: Von Nostrand.

[76] Lu, H., Plataniotis, K. N., & Venetsanopoulos, A. N. (2008). MPCA: Multilinear principal component analysis of tensor objects. *Neural Networks, IEEE Transactions on*, 19(1), 18-39.

[77] Lumley J.L. (1967). The Structure of Inhomogeneous Turbulent Flows. In Yaglom A.M., Tatarski V.I (Eds.) *Atmospheric turbulence and radio propagation* (pp. 166-178). Moscow: Nauka.

[78] Lyman, P., & Varian, H. (2004). How much information 2003? School of Information Management and Systems, University of California at Berkeley.

[79] Maitra, R., Peterson, A. D., & Ghosh, A. P. (2010). A systematic evaluation of different methods for initializing the K-means clustering algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 522-537.

[80] Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. (2011). Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute.

[81] Mardia, K. V., & Jupp, P. E. (2009). Directional statistics (Vol. 494). John Wiley & Sons.

[82] Matsushita, H., & Nishio, Y. (2008, June). Batch-Learning Self-Organizing Map with false-neighbor degree between neurons. *In Neural Networks, 2008*. *IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on (pp. 2259-2266).* IEEE.

[83] Mirkes, E.M. Principal Component Analysis and Self-Organizing Maps: applet. University of Leicester, 2011. http://www.math.le.ac.uk/people/ag153/homepage/PCA_SOM/PCA_SOM.html

[84] Paluš, M., & Dvořák, I. (1992). Singular-value decomposition in attractor reconstruction: pitfalls and precautions*. Physica D: Nonlinear Phenomena,* 55(1), 221-234.

[85] Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science,* 2(11), 559-572.

[86] Pena, J. M., Lozano, J. A., & Larranaga, P. (1999). An empirical comparison of four initialization methods for the K-Means algorithm. *Pattern recognition letters*, 20(10), 1027-1040.

[87] Reinsel, D., Chute, C., Schlichting, W., McArthur, J., Minton, S., Xheneti, I., Toncheva, A., & Manfrediz, A. (2007). The Expanding Digital Universe. White paper, IDC.

[88] Richardson, M. (2009). Principal component analysis. Mathematical Modelling and Scientific Computing, University of Oxford, Oxford, UK.

[89] Shlens, J. (2014). A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100.*

[90] Rao, C. R. (1964). The use and interpretation of principal component analysis in applied research. Sankhyā: *The Indian Journal of Statistics, Series A*, 329-358.

[91] Roweis, S. (1998). EM algorithms for PCA and SPCA. *Advances in neural information processing systems*, 626-632.

[92] Schölkopf, B., Smola, A., & Müller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5), 1299-1319.

[93] Short, J. E., Bohn, R. E., & Baru, C. (2011). How much information? 2010 report on enterprise server information. , University of California, San Diego, Global Information Industry Centre.

[94] Su, M. C., Liu, T. K., & Chang, H. T. (2002). Improving the self-organizing feature map algorithm using an efficient initialization scheme. *Tamkang Journal of Science and Engineering*, 5(1), 35-48.

[95] Sylvester, J. J. (1889). On the reduction of a bilinear quantic of the nth order to the form of a sum of n products by a double orthogonal substitution. *Messenger of Mathematics*, 19, 42-46.

[96] Tarpey, T., & Flury, B. (1996). Self-consistency: A fundamental concept in statistics. *Statistical Science*, 229-243. Doi:10.1214/ss/1032280215

[97] Tarpey, T. (1999). Self-consistency and principal component analysis. *Journal of the American Statistical Association*, 94(446), 456-467.

[98] Tarpey, T., Li, L., & Flury, B. D. (1995). Principal points and self-consistent points of elliptical distributions. *The Annals of Statistics*, 23(1), 103-112.

[99] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*. Series B (Methodological), 267-288.

[100] Tibshirani, R. (1992). Principal curves revisited. *Statistics and Computing*, 2(4), 183-190.

[101] Tipping, M., & Bishop, C. (1999). Mixtures of probabilistic principal component analyzers. *Neural computation,* 11(2), 443-482.

[102] Tipping, M. E., & Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society*: Series B (Statistical Methodology), 61(3), 611-622.

[103] Venables, W. N., & Ripley, B. D. (2002). Modern applied statistics with S. Springer. ISBN 0- 387-95457-0, ISBN 978-0-387-95457-8, page 336 (from wiki)

[104] Verbeek, J. J., Vlassis, N., & Kröse, B. (2002). A k-segments algorithm for finding principal curves. *Pattern Recognition Letters*, 23(8), 1009-1017.

[105] Yang, J., Zhang, D., Frangi, A. F., & Yang, J. Y. (2004). Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(1), 131-137.

[106] Yin, H. (2008). The self-organizing maps: Background, theories, extensions and applications. In Computational intelligence: *A compendium* (pp. 715-762). Springer Berlin Heidelberg.

[107] Yin, H. (2008). Learning nonlinear principal manifolds by self-organising maps. In Principal manifolds for data visualization and dimension reduction (pp. 68-95). Springer Berlin Heidelberg.

[108] Farrar, D. E., & Glauber, R. R. (1967). Multicollinearity in regression analysis: the problem revisited. *The Review of Economic and Statistics*, 92-107.

[109] Tu, Y. K., Kellett, M., Clerehugh, V., & Gilthorpe, M. S. (2005). Problems of correlations between explanatory variables in multiple regression analyses in the dental literature. *British dental journal*, *199*(7), 457-461.

[110] Tan, J. (2012). Principal Component Analysis and Portfolio Optimization. *Available at SSRN 2213687*

[111] Dong, D., & McAvoy, T. J. (1996). Nonlinear principal component analysis—based on principal curves and neural networks. *Computers & Chemical Engineering,* 20(1), 65-78.

[112] Gorban, A. N., Tyukin, I. Y., Prokhorov, D. V., & Sofeikov, K. I. (2015). Approximation with random bases: Pro et Contra. *Information Sciences.*

[113] Flanagan, J. A. (1996). Self-organisation in Kohonen's SOM. Neural networks, 9(7), 1185-1197.

[114] Baralić, D. (2011). How to understand Grassmannians?. *The Teaching of Mathematics*, (27), 147-157.

[115] Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. Psychometrika, 1(3), 211-218.

[116] Johnson, R. M. (1963). On a theorem stated by Eckart and Young. Psychometrika, 28(3), 259-263.

[117] Zucchini, W., Berzel, A., & Nenadic, O. (2003). Applied smoothing techniques. Part I: Kernel Density Estimation, 15.