Manifold Learning for Image Processing

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

MASTER OF SCIENCE
IN
ADVANCED COMBINATORICS

by

PANKAJ KUMAR

Moscow Institute of Physics and Technology

Moscow, Russian Federation

June, 2017

Moscow Institute of Physics and Technology

Moscow, Russian Federation

CERTIFICATE

This is to certify that the present work entitled, "**Manifold Learning for Image Processing**" submitted by **Pankaj Kumar** to the Moscow Insitute of Physics and Technology, Russian Federation in partial fulfilment of the requirements for the award of the degree of Master of Science is approved.

.....
Prof. Roman Karasev
Professor
Department of Mathematics
Moscow Institute of Physics and
Technology

Moscow Institute of Physics and Technology

Moscow, Russian Federation

DECLARATION

I hereby declare that the thesis titled, "**Manifold Learning for Image Processing**" being submitted to the Moscow Institute of Physics and Technology, Russian Federation in partial fulfillment of the requirements for the award of the degree of Master of Sceience is a record of bonafide work carried out by me.

The matter embodied in the thesis has not been submitted for the award of any degree in any university or institution.

June, 2017
Moscow, Russian Federation

Pankaj Kumar

Acknowledgments

There are no proper words to convey my deep gratitude and respect for my thesis and research advisor, Prof. Roman Karasev, Moscow Insitute of Physics and Technology, Russian Federation. With his course in discrete geometry, he not only inculcated my interest in manifold learning, but has inspired me to become an independent researcher. He helped me realize the power of critical reasoning, simple, but concrete mathematical representation of idea and guidance to recover when my steps faltered. He taught me how to critically question thoughts and express ideas in simple but scientific way. His insightful thought-provoking comments and constructive criticisms at different stages of my research allowed me to stay focused to core of research ideas. I am grateful to him for holding me to a high research standard and enforcing strict validations for each research result, and thus teaching me how to do research.

There is no way to express how much it meant to me to have been a member and faculty of discrete mathematics department for all meaningful discussion on different plethora of research topics. The brilliant friends, colleagues and and all the other current and former Lab graduates students and visitors that I know inspired me over my stay in Moscow. Also, thanks to active communities of Python and Matlab.

I would like to thank all the people associated with Department of Discrete Mathematics, fellow students and teachers who have been source of thoughts, inspiration and smiles. Acknowledgement will be incomplete without expressing sincere gratitude to Prof. Andrei Raigorodskii for providing excellent infrastructure, guidance and scholastic path to realize our dreams into reality.

Though only some name appears on the cover of this thesis, a great many people have contributed to its production. I owe my deep gratitude to all those people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever.

In conclusion, I recognize that this research would not have been possible without the financial assistance from Russian Government and MIPT, and express my gratitude to those agencies.

Pankaj Kumar

Abstract

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

The rapidly progressing digital revolution have infused enormous amount of images and video, which is growing constantly. Dealing with this large scale data calls for software application for enhancing image quality is the topic of Image Processing. Although image processing algorithms are getting accurate day by day, there is constant pressure on algorithms to deal with noise. It is inherent to the acquisition tools. State of the art preprocessing and postprocessing algorithms are being developed to suppress noise with altering the important content. The noise in an image is reduced by convolving the image with a Gaussian function. The above common approach blurs significant features and destroys some of the geometric information in the image [1]. Image processing task are often tackled with geometric methods, which targets to understand the geometric configuration and relation between the observed objects. Manifold learning algorithms are one such geometric methods.

1.2 Manifold Learning

Manifold learning is form of unsupervised machine learning algorithms, which extract low-dimensional structure from high dimensional data. These algorithms typically try to unfold the underlying manifold so that Euclidean distance in the new space is a meaningful measure of distance metrics between any pair of points. Manifold learning assumes that the data lies approximately on a low dimensional surface embedded in a high dimensional space [2]. It can be further illustrated using figure 1.1, where manifold learning algorithms for non-linear data builds an embedding function f mapping $\mathcal{M}$ to $\mathbb{R}^2$.

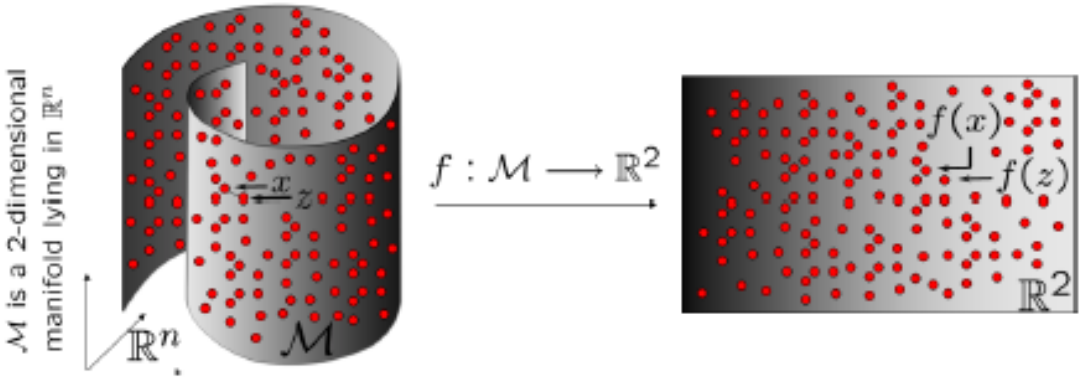


FIGURE 1.1: Manifold Learning [3]

1.3 Motivation

Consider the example of an image sequence. In the absence of features such as contour points or wavelet coefficients, each image is a point in a space of dimension equal to the number of image pixels. When facing an observation space of possibly tens or hundreds of thousands of dimensions, it is often reasonable to assume that the data is not dense in such a space and that many of the measured variables must be dependent with only a few free parameters that are embedded in the observed variables, frequently in a nonlinear way. Assuming that the number of free parameters remains the same throughout the observations, and also assuming spatially smooth variation of the parameters, we have geometric restrictions which can be well modeled as a manifold. Learning this manifold is a natural approach to the problem of modeling the data, with the advantage of allowing nonlinear dimensionality reduction

1.4 Contribution

This paper presents a novel manifold learning approach for high dimensional data, with emphasis on the problem of anomaly detection in image.

1.5 Organization

Chapter 2

Mathematical Background

Overview

This chapter is dedicated to provide mathematical notions necessary for under standing the intuition behind manifold learning. Further, we present the most representative manifold learning algorithms which are used in this thesis to solve specific problems in image processing and anomaly detection. We follow the structure from Nicolas Thorstensen [1] and Boothby [5].

2.1 Metric Space

Use of near and far is intuitive yet rigorous at the same time, which is rare in mathematics. Classifying the relationship between two 'primitives', whether they are close or far apart is some time not useful for the computation. A metric space is the mathematical construct of redefining the near and far primitive idea, which is useful in computation.

Definition 2.1. A metric on an arbitrary abstract set $\mathbb{X}$ is a function $d : \mathbb{X} \times \mathbb{X} \to [0, \infty)$ such that for all $a, b, c \in \mathbb{X}$, the following conditions are satisfied:

1. Non-negativity: $d(a, b) \geq 0 and d(a, b) = 0 \Leftrightarrow a = b$
2. Symmetry: $d(a, b) = d(b, a)$
3. Triangle inequality: $d(a, c) \leq d(a, b) + d(b, c)$

Then we say that d is a *metric* on $\mathbb{X}$ and that $(\mathbb{X}, d)$ is a *metric space*.

A quite known instance of a metric space is the three dimensional Euclidean space $\mathbb{R}^3$ with the Euclidean metric.

Definition 2.2. Let V be a vector space over $\mathbb{R}$ and $N : V \to \mathbb{R}$ a map such that, $N(\mathbf{v}) = \|\mathbf{v}\|$, the following condition holds:

(i) Non-negativity: $\|\mathbf{v}\| \geq 0$ for all $\mathbf{v} \in V$. If $\|\mathbf{v}\| = 0$, then $\mathbf{v} = \mathbf{0}$.

(ii) Linearity: If $\lambda \in \mathbb{R}$ and $\mathbf{v} \in V$, then $\|\lambda\mathbf{v}\| = |\lambda|\|\mathbf{v}\|$.

(iii)Triangle inequality: If $\mathbf{v}_1$, $\mathbf{v}_2 \in V$, then $\|\mathbf{v}_1\| + \|\mathbf{v}_2\| \geq \|\mathbf{v}_1 + \mathbf{v}_2\|$.

Then we call $\| \ \|$ a *norm* and say that $(V, \| \ \|)$ is a *normed vector space*.

Any normed vector space can be made into a metric space by the condition $d(\mathbf{v}_1, \mathbf{v}_2) = \|\mathbf{v}_1 - \mathbf{v}_2\|$.

Definition 2.3. Let V be a vector space over $\mathbb{R}$ and $M : V \times V \to \mathbb{R}$ a map such that, writing $M(\mathbf{v}_1, \mathbf{v}_2) = \langle \mathbf{v}_1, \mathbf{v}_2 \rangle$, the following results hold for $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3 \in V$, $\lambda \in \mathbb{R}$.

(i) $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle \geq 0$.

(ii) If $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle = 0$, then $\mathbf{v}_1 = \mathbf{0}$.

(iii) $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle = \langle \mathbf{v}_1, \mathbf{v}_2 \rangle$.

(iv) $\langle \mathbf{v}_1 + \mathbf{v}_3, \mathbf{v}_2 \rangle = \langle \mathbf{v}_1, \mathbf{v}_2 \rangle + \langle \mathbf{v}_3, \mathbf{v}_2 \rangle$.

(v) $\langle \lambda\mathbf{v}_1, \mathbf{v}_2 \rangle = \lambda\langle \mathbf{v}_1, \mathbf{v}_2 \rangle$.

Then we call $\langle \ , \ \rangle$ an *inner product* and say that $(V, \langle \ , \ \rangle)$ is an *inner product space*.

Working on $\mathbb{R}^n$ to make it vector space, then $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{j=1}^{n} x_j y_j$ is an inner product. We notice that the norm derived from inner product is called Euclidean norm.

2.2 Topology

With the metric structure in hand, the idea of 'near' and 'far' between elements of metric space can be quantified by characterizing the connectivity of sets through small neighborhoods of elements which gives rise to a topology of the sets.

Definition 2.4. Let $(\mathbb{X}, d)$ be a metric space and element $x \in \mathbb{X}$. If $r > 0$, then $B_{open}(x, r) = \{y \ : \ d(x, y) < r\}$ is called as *open ball* with centre $\mathbf{x}$ and radius r. Similarly, $B_{closed}(x, r) = \{y \ : \ d(x, y) \leq r\}$ is called closed ball.

The topology of $\mathbb{X}$ induce by the metric is easy to study.

Definition 2.5. Let $\mathbb{X}$ be a set and τ a collection of subsets of $\mathbb{X}$ with the following properties.

(i) The empty set $\emptyset \in \tau$ and the space $\mathbb{X} \in \tau$.

(ii) If $U_\alpha \in \tau$ for all $\alpha \in A$, then $\bigcup_{\alpha \in A} U_\alpha \in \tau$.

(iii) If $U_j \in \tau$ for all $1 \leq j \leq n$, then $\bigcap_{j=1}^{n} U_j \in \tau$.

Then we say that τ is a *topology* on $\mathbb{X}$ and that $(\mathbb{X}, \tau)$ is a *topological space*.

If $(\mathbb{X}, d)$ is a metric space, then the collection of open sets forms a topology. The topology of a set allows one to study properties such as connectivity.

2.2.1 Connectedness

The metric space $(\mathbb{X}, d)$ is not connected if it is the union of two disjoint open nonempty sets. Similarly, the converse of earlier implies that $(\mathbb{X}, d)$ is connected. Mathematically, It can be defined as:

Definition 2.6. A topological space $(\mathbb{X}, \tau)$ is said to be *disconnected* if we can find non-empty open sets V_1 and V_2 such that $V_1 \cup V_2 = \mathbb{X}$ and $V_1 \cap V_2 = \emptyset$. A space which is not disconnected is called *connected*.

2.2.2 Neighborhoods

Hausdorff was the first one to define topologies in terms of neighborhoods. It always appears to be technically easier to define topologies in terms of open sets as we have been doing it so far. Generally, It can be defined as:

Definition 2.7. Let $(\mathbb{X}, \tau)$ be a topological space. If $x \in \mathbb{X}$, we say that N is a *neighbourhood* of x if we can find $U \in \tau$ with $x \in U \subseteq N$.

2.3 Manifolds

A manifold is a topological space that locally resembles Euclidean space near each point. More precisely, they are spaces that locally look like the much more familiar Euclidean spaces. This allows the well-understood notions on Euclidean spaces to be generalised to manifolds rather directly. Using all the above definition, we now define manifold.

Definition 2.8. A d-dimensional manifold is a topological space 2.5 in which points can be separated by neighborhoods 2.7 and where every point has a neighborhood that is homeomorphically mapped onto an open Euclidean ddimensional ball [1].

As the general definition is not apt to perform vector calculus on a manifold, The the notion of a differentiable manifold comes hand in hand with the concept of coordinate chart 2.1. It can be thought of as assigning a set of coordinates to the points in the neighborhood U.

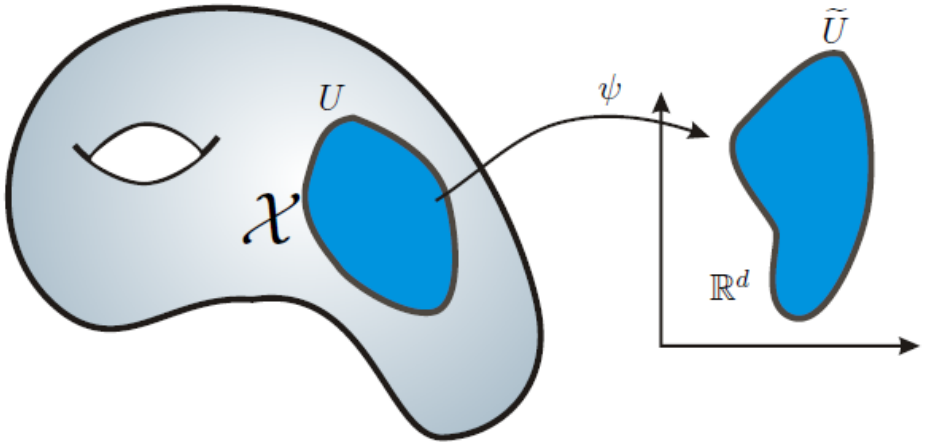


FIGURE 2.1: Coordinate Chart[3]

To provide an entire description of the manifold, several chart over manifold is used rather single. Transition map with two chart, corresponds to a change of coordinates is depicted in the figure 2.2.

2.4 Tangent Spaces

The notion of linear approximation of a surface in vector calculus translates to tangent space in differential geometry. The tangent space to a manifold $\mathcal{X}$ at a point x is the closest flat approximation to $\mathcal{X}$ at that point. If the dimension of $\mathcal{X}$ is n, then the tangent space is an $\mathbb{R}^n$ grazing $\mathcal{X}$ at x, as shown in figure 2.3. For detail discussion please refer work by Nicolas Thorstensen [1] and Boothby [5].

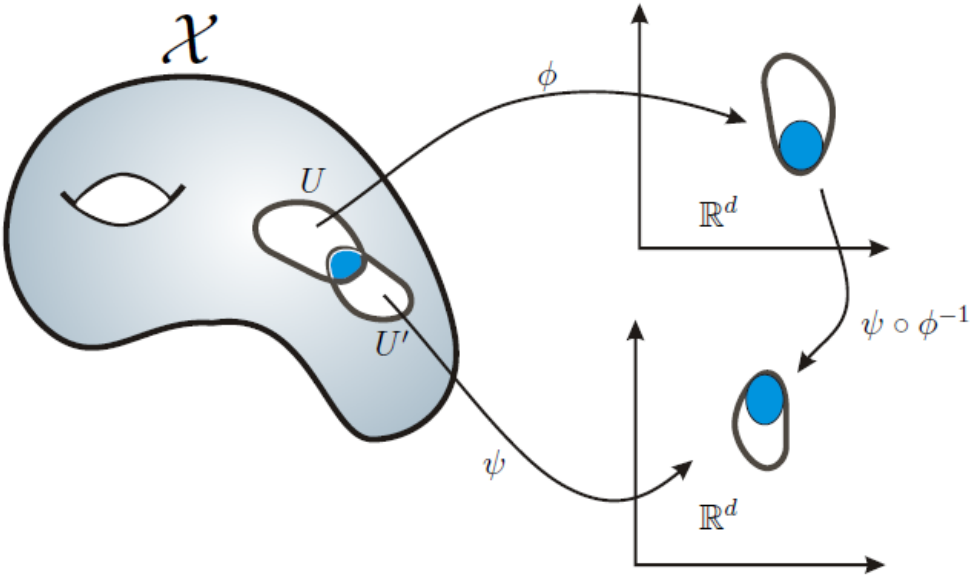


FIGURE 2.2: Transition map.[3]

2.4.1 Embedding

When talking about the relationship between two topological objects, it is interesting to imagine a mapping from one object to the other that somehow preserves its original properties.

Let $\mathbb{X}$ and $\mathbb{Y}$ be two topological objects of the same type, a function $\phi\colon \mathbb{X} \to \mathbb{Y}$ is an embedding of $\mathbb{X}$ into $\mathbb{Y}$ if ϕ is an isomorphism which preserves the original properties of $\mathbb{X}$, where such properties are relative to the type of the objects at hand. Shortly, $\mathbb{X}$ is said to be *embedded* in $\mathbb{Y}$.

Example 2.1 (Embedding of Spaces)**.** *Consider the vector spaces* $\mathbb{R}^p$ *and* $\mathbb{R}^q, p \leq q, p > 0$ *and the isomorphism* $t\colon \mathbb{R}^q \to \mathbb{R}^t \mid t(x) = [x|\bar{0}] = y$*, where* y *is the vector* x *concatenated with* $p - q$ *zeros.* $\mathbb{R}^p$ *is embedded on* $\mathbb{R}^q$*, as the operations sum and scalar multiplication are preserved.*

The same definition applies to manifolds too.

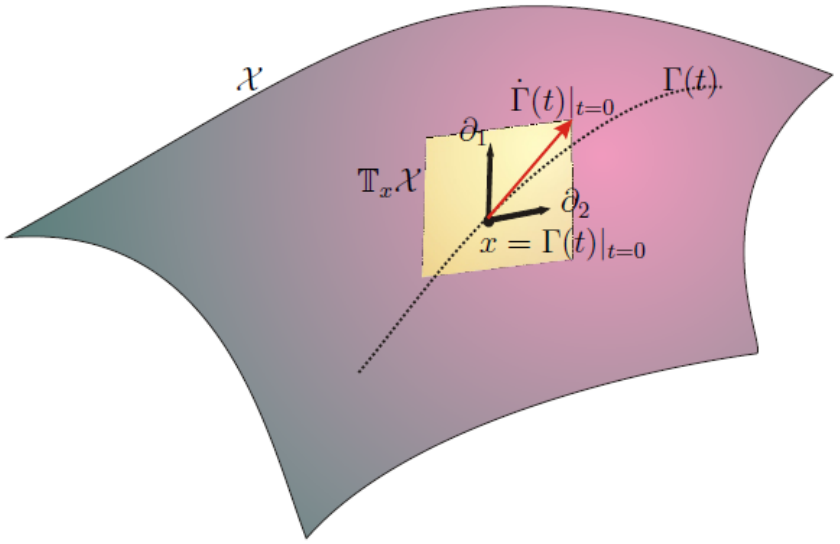


FIGURE 2.3: Tangent Space.[3]

Chapter 3

Manifold Learning

Overview

Manifold learning is an emerging and promising approach in nonparametric dimension reduction widely used in image processing, data mining, signals processing and computer vision. The core theme of manifold leaning is to find the most succinct low dimensional structure that is embedded in a higher dimensional space. Using earlier, the algorithms can work directly in the lower dimensional latent space of the manifold rather high dimensional data space and thus get computationally efficient. In this chapter, we review the representative sample of manifold learning, mathematical developments, as well as some interesting applications.

3.1 Introduction

Most of the datasets encountered in image processing often consist of a large number of samples each of which are themselves of high dimension. Further processing of the datum is done either treating it directly or extracting complex features before processing. Nevertheless, it wither its entire datum or a set extracted features, the dimensionality of problems usually remain high. This in turn slows down the processing considerably or sometimes making the treatment even intractable. The task to reducing the computational burden is to reduce the dimensionality of the data. Dimensionality reduction can be thought of as mathematical mapping of high-dimensional data into a meaningful representation of the intrinsic dimensionality. The intrinsic dimensionality of a data set can be defined as the lowest number of variables that can preserve the geometrical structure of the data.

3.2 Problem Definition

Given a set of of high-dimensional training instances $\mathbb{X} = \{x_1, x_2, ..., x_N\}$, where $x_i \in \mathbb{R}^D$. We assume that $\mathbb{X}$ approximately lie on a smooth manifold $\mathcal{X}$. The idea of manifold learning algorithms is to find an embedding set $\mathbb{Y} = \{y_1, y_2, ..., y_N\}$ of $\mathbb{X}$ in low dimension space $\mathbb{R}^d$, where $d < D$. The local manifold structure formed by $\mathbb{X}$ in the original space $\mathbb{R}^D$ is preserved in the embedded space $\mathbb{R}^d$.

Manifold Learning can be mainly divided into linear and non linear methods. Linear methods, which have long been used for analyzing multivariate data, include principal component analysis (PCA) and multidimensional scaling (MDS). A further distinction within non-linear methods is to divide the methods into two classes: one is to preserve the global geometric structure of manifold, such as Isomap [6] and the other one is to preserve the local neighborhood geometric structure, such as LLE [7].

3.3 Linear Methods

3.3.1 Principle Component Analysis

Principal component analysis (PCA) is a standard algorithm to explain the dispersion of a point cloud by projecting data onto a carefully chosen linear subspace. It reduces the dimensionality of the point cloud while keeping the maximum of variance information. The dispersion in linear subspaces of higher dimension is captured by by building an orthonormal basis such that the projection along each axis has a decreasing maximum variance.

We consider n points $\mathbf{X} = \{x_1, x_2, ..., x_n\}$ of dimension D from $\mathbb{R}^D$. Also, let us assume V^* be a d-dimensional subspace of $\mathbb{R}^D$ and let $u = \{u_1, u_2, ..., u_D\}$ be an orthonormal basis of $\mathbb{R}^D$ such that $\{u_1, u_2, ..., u_d\}$ is a basis of V. Then each data point is approximated by:

$$y_n = \sum_{i=1}^{d} a_{n,i} u_i + \sum_{i=1}^{D+1} a_{n,i} u_i \tag{3.1}$$

The below energy function is used to recover $u_1, u_3, ..., u_d$.

$$\mathbb{E} = \frac{1}{N} \sum_{n=1}^{N} \|x_n - y_n\|^2 \tag{3.2}$$

This is equivalent to minimizing the approximation error.

$$x_n - y_n = \sum_{i=d+1}^{D} \{(x_n - \bar{x})^T u_i\} u_i \tag{3.3}$$

The above equation 3.3 can be best visualized in the figure 3.1.

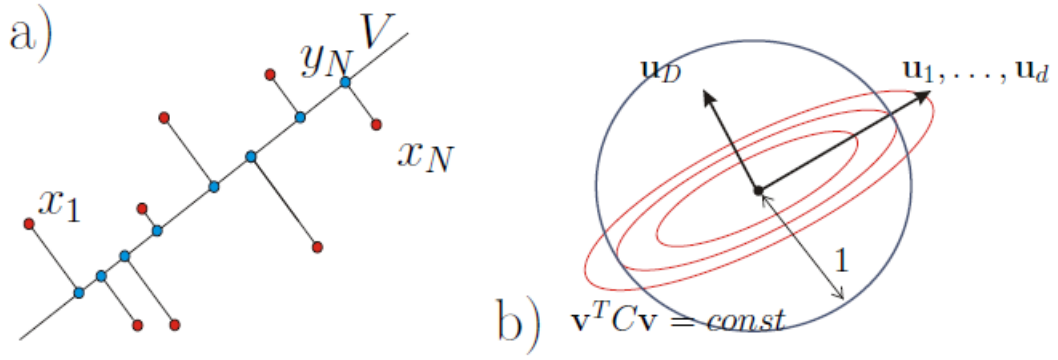


FIGURE 3.1: a) Manifold with original (red) and projected (blue) points. b) Geometry and constraint visualization of optimization function. Pictures taken from Nicolas Thorstensen works [3]

Now, we can write the energy function as

$$\mathbb{E} = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=d+1}^{D} (x_n^T u_i - \bar{x}^T u_i)^2 = \sum_{i=d+1}^{D} u_i^T \mathbf{C} u_i \tag{3.4}$$

$\mathbf{C}$ is defined as covariance matrix, $\mathbf{C} = \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^2$. Our optimization problem look like

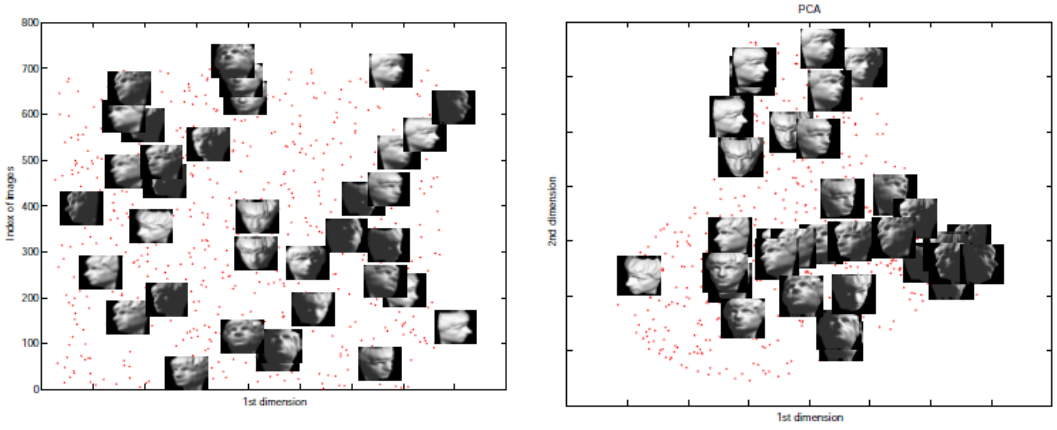
$$\begin{aligned} \underset{D}{\text{minimize}} \quad & \mathbb{E} = u_D^T \mathbf{C} u_D \\ \text{subject to} \quad & u_D^T u_D = 1 \end{aligned} \tag{3.5}$$

Now using Lagrange multipliers, we solve this constraint optimization problem as

$$u_D^T \mathbf{C} u_D + \lambda(1 - u_D^T u_D) = 0 \tag{3.6}$$

The solution off the above equation gives standard form eigenvalue problem $\mathbf{C} u_D = \lambda u_D$. The geometry of the minimization problem is illustrated in figure 3.1 b).

We take an input image 3.2 (A), which is sequence of vector in $\mathbb{R}^{4096}$ representing the brightness values of 64×64 pixel image of the face. The 1st dimension representing embedding correlates with intrinsic-dimension of one, which is left-right pose. The two-dimensional projection of original input image found by PCA is shown in figure 3.2 (B).



(A) Input

(B) Output

FIGURE 3.2: Application of PCA.

3.3.2 Multi-Dimensional Scaling

The Multi-Dimensional Scaling (MDS), is a PCA-like technique that maps the original high dimensional space to a lower dimensional space by preserving pairwise distances. Figure 3.3 shows an example of MDS, where pairwise distance is preserved. MDS [8] is applied when it difficult to calculate covariance matrix from the dataset, when data are, for instance, qualitative or infinite dimensional, when only a point-wise distance is known or also when the size $\mathbb{D}$ of the sample set is lower than the dimension of data [3].

MDS addresses the problem of finding d-dimensional Euclidean coordinates for each data sample ($x_i \in \mathbf{X}$) so that the pairwise distance (d) of their Euclidean coordinates match the original pairwise distance as closely as possible.

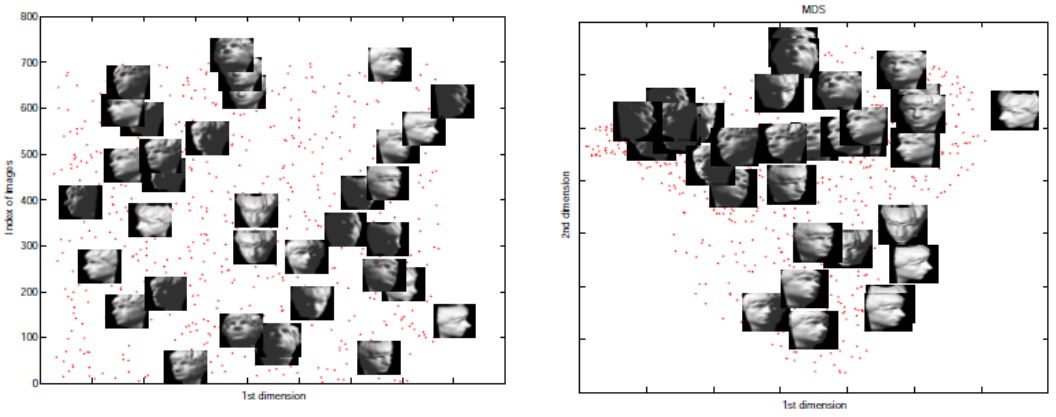
We define $N \times N$ euclidean distance or affinity matrices $\mathbf{D}$ as symmetric, $d_{ii} = 0$ and $d_{ij} > 0$, if $i \neq j$. With $\mathbf{D}$ matrix in hand, the MDS algorithm attempts to find N data points $\mathbf{Y} = y_1, y_2, y_3, ..., y_N$ in $\mathbb{R}^d$ such that distance matrix is preserved as depicted in the figure 3.4.

In particular, we try to optimize following function:

$$\underset{\mathbf{Y}}{\text{Min}} \quad \sum_{i=1}^{N}\sum_{i=1}^{N}(d_{ij}^{\mathbf{X}} - d_{ij}^{\mathbf{Y}})^2 \tag{3.7}$$

where, $d_{ij}^{\mathbf{X}}$ and $d_{ij}^{\mathbf{Y}}$ is normed distance from sample point x_i and x_i respectively. Converting distance $\mathbf{D}^{\mathbb{X}}$ matrix into kernel matrix of inner product yields:

$$\mathbf{X}^T\mathbf{X} = -\frac{1}{2}\mathbf{J}\mathbf{D}^{\mathbb{X}} \tag{3.8}$$



(A) Input (B) Output

FIGURE 3.3: Application of MDS.

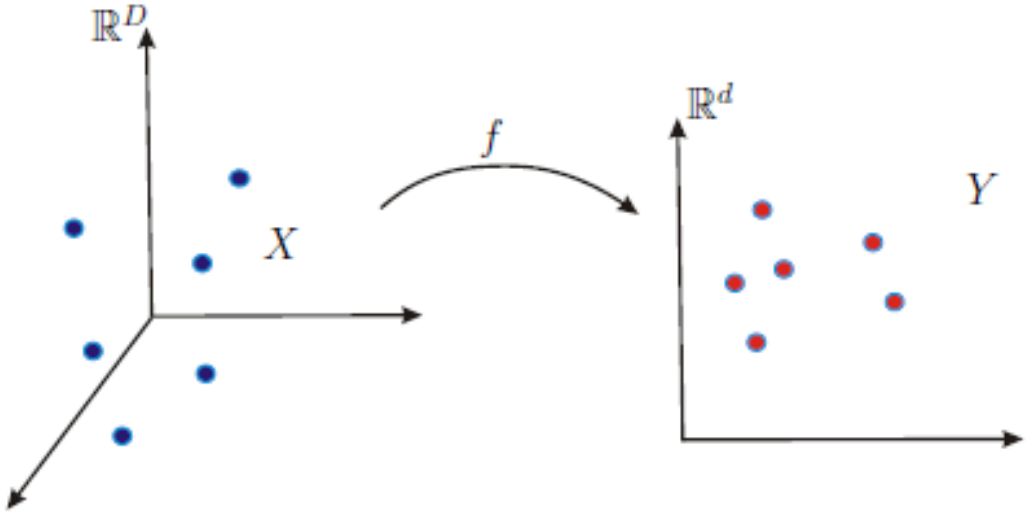


FIGURE 3.4: Multi-Dimensional Scaling

where $\mathbf{J} = \mathcal{I}d_N - \frac{1}{N}\mathbf{1}\mathbf{1}^T$. Now the equation 3.7 can be reduced into

$$\underset{\mathbf{Y}}{\text{Min}} \quad \sum_{i=1}^{N}\sum_{i=1}^{N}(x_i^T x_j - y_i^T y_j)^2 \tag{3.9}$$

Solving equation 3.9 as shown in Cox and Cox paper[8] gives $\mathbf{Y} = \lambda^{\frac{1}{2}}\mathbf{U}^T$. Here, $\mathbf{U}$ is eigenvector of $\mathbf{X}^T\mathbf{X}$ corresponding to top d eigenvalues, and λ is top d eigenvalues of $\mathbf{X}^T\mathbf{X}$.

MDS can be generalized into to geodesic distances between point samples from a non-linear manifold. But a non-convex energy function coming out makes it

tedious to optimize. In the next section we will provide an overview of non-linear manifold algorithms.

3.4 Non-Linear Methods

Linear methods such as PCA and MDS fails to produce desired results when the data is sampled from non linear manifolds [1]. We now review representative non-linear methods in manifold learning in this section with assumption that the data is distributed along a d-dimensional submanifold $\mathcal{X}$ embedded in $\mathbb{R}^D$.

3.4.1 Graph-based Algorithms

Manifold learning based on graph based algorithms often relies on preserving the geometric structure in the neighborhood of some points. The important step is to construct a graph using K-nearest neighbor graph or ε-neighborhood and apply MDS as previously detailed.

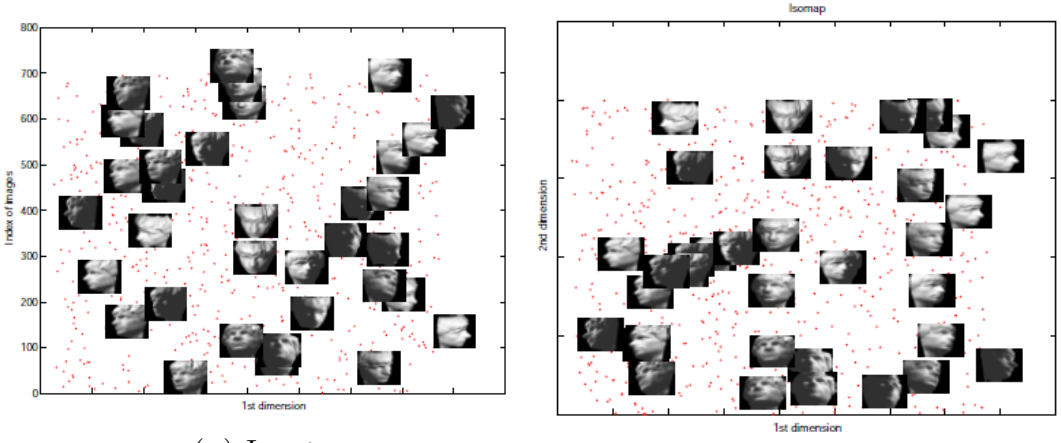
3.4.1.1 Isomap

Isomap was proposed by Joshua Tenenbaum, Vin de Silva and John Langford in Science [6]. It is an extended version of MDS for data lying on smooth non-linear manifold. Unlike MDS, the pairwise distance matrix in Isomap is replaced by the matrix of pairwise geodesic distances approximated by distances in graphs. The algorithm proceeds in three steps. First a distance $d(x; y)$ is considered in the data space. Then, a neighborhood graph based on a ε-neighborhood or k-nearest neighbor points is build with following weights $D_{ij} = d(xi; xj)$ if an edge link exists between points x_i and x_j , otherwise $D_{ij} = 1$. The pairwise distance matrix $\mathbf{D}$ is then calculated using a Dijkstra-like algorithm in the graph. Finally, the data via MDS is embedded so as to preserve these distances. A two-dimensional projection using isomap is shown in the figure 3.5 with k=6.

Precautionary measure should be taken while calculating distances between far apart points on the manifold, which can be very close in the data space as discussed in the figure 3.6.

3.4.1.2 Locally linear Embedding

Locally linear Embedding (LLE) is a manifold mapping dimension reduction algorithm introduced by Roweis & Saul [7]. It aims at recovering the low dimensional geometry of the data by looking at the local interaction between data points. The construction of a nearest neighbor graph recovers the local interactions. Then



(A) Input

(B) Output

FIGURE 3.5: Application of Isomap

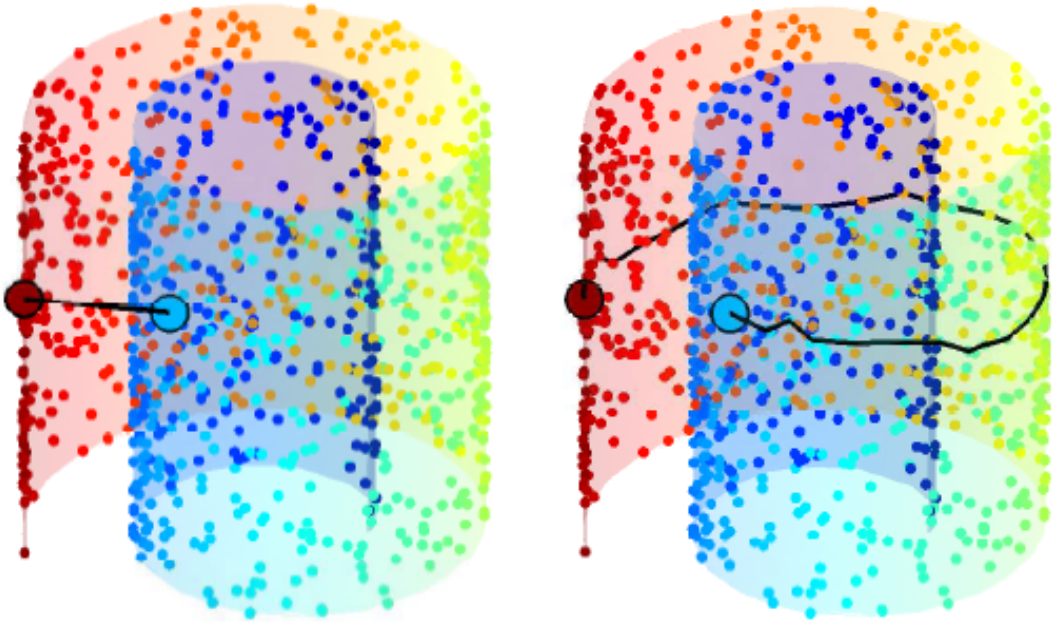


FIGURE 3.6: Multi-Dimensional Scaling. Left: Distance between two points using a distance in the data space. Right: Distance between two points using an approximated geodesic path. Image taken from Patrick Etyngier work [3].

further processing is done to reduce the dimensionality of the data and provide a parametrization in a d- dimensional hyperplane. LLE algorithm can be summarized in the following steps:

1. Neighborhood graph $\mathcal{G}$ is build on the dataset $\mathbb{X}$ using K-nearest neighbour method.

2. Weights W_{ij} are calculated to each of the nearest neighbours pair (x_i, y_i) by minimising the cost function:

$$\varepsilon(W) = \sum_i |x_i - \sum_{j=1}^{k} W_{ij} x_j|^2 \tag{3.10}$$

3. The inner products between each point x_i and each of its nearest neighbours are computed to produce gram positive-semidefinite matrix, $G_{jk} = (x - x_j)^T(x - x_i)$.

4. The reconstruction weights are then computed using:

$$W_{ij} = \sum_k \mathbb{G} - jk^{-1}(G_{jk} + \lambda) \tag{3.11}$$

such that, $W_{ij} = 0$ if x_j is not one of the nearest neighbours of x_i and $\sum_j W_{ij} = 1$)

5. Finally the embedded $\mathbb{Y}$ which will make up the final output data are calculated by minimising a second cost function:

$$\Phi(Y) = \sum_i |y_i - \sum_j W_{ij} y_j|^2 \tag{3.12}$$

We also constraint $\sum_i y_i = 0$ to centre the projection on the origin. The other constraint is imposed in order to avoid degenerate solutions;

$$\frac{1}{n} \sum_i y_i y_i^T = I \tag{3.13}$$

where I is the d×d identity matrix.

6. This cost function now defines a quadratic form containing the N×N symmetric matrix M_{ij}:

$$\Phi(I) = \sum_{ij} M_{ij} y_i y_i^T \tag{3.14}$$

where $M_{ij} = \delta_{ij} - W_{ij} - W_{ij} + \sum_k W_{ki} W_{kj}$ and $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise.

7. The embedding $\mathbb{Y}$ is then given by the the lowest d+1 eigenvectors of the matrix M_{ij}. The bottom eigenvector representing a free translation mode of eigenvalue 0 is left out in order to find the optimum embedding.

The geometry the optimization problem is depicted in figure 3.7.

A two-dimensional projection of the original input by using with k=5 is shown in the figure 3.8.

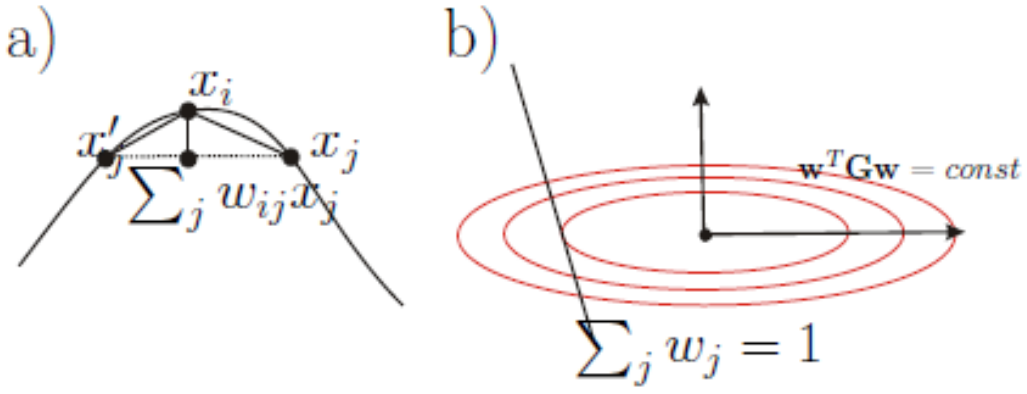
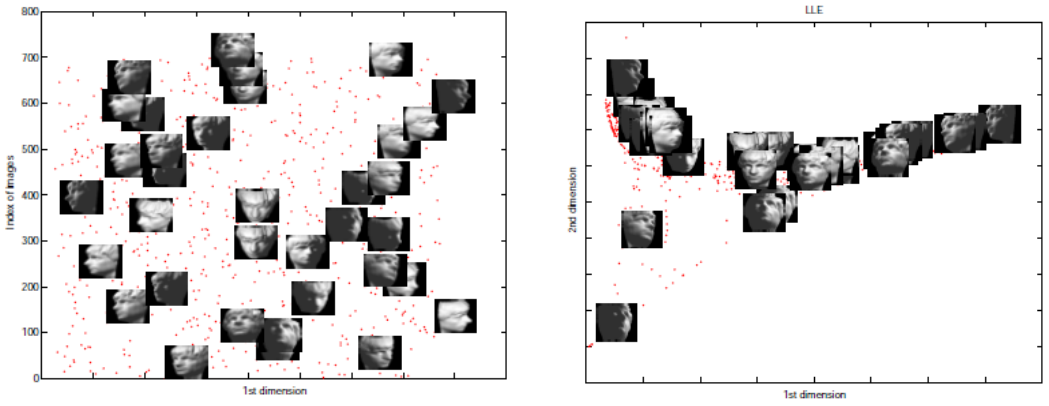


FIGURE 3.7: LLE. a) x_i is appropriated by its neighborhood (x'_j, x_j). b) The black line touching the level set at a single point defines the constraints



(A) Input

(B) Output

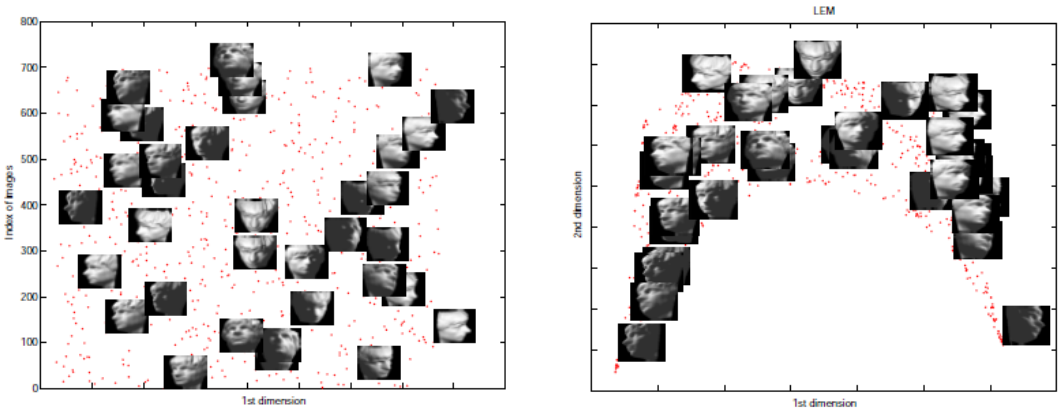
FIGURE 3.8: Application of LLE

3.4.2 Laplacian-based Algorithms

Laplacian based algorithms assume that data have a structure of low dimensional manifold embedded in a higher dimensional space. The maps are constructed into a low dimensional space preserving the local neighborhood topology. In this part, we give outlines of representative laplacian based algorithms.

3.4.2.1 Laplacian Eigenmaps

Laplacian Eigenmaps were introduced by Belkin [4]. Its intent is to embed the observed data $\mathbf{X}$ into $\mathbb{R}^d$ by first constructing the k-nearest-neighbor or ε-graph $\mathcal{G}$ from $\mathbf{X}$. In the k-nearest-neighbor graph, an edge is present between x_i and x_j if x_i is among the k nearest neighbors of x_j or vice versa. In the ε-graph, x_i and



(A) Input

(B) Output

FIGURE 3.9: Application of Laplacian Eigenmaps

x_j are adjacent if $\|x_i - x_j\|^2 < \varepsilon$ for a given threshold parameter ε. The weighted adjacency matrix of $\mathcal{G}$ is defined as $\mathbf{W}$,

$$\mathbf{W}_{ij} = \begin{cases} \mathbf{K}_{ij} & \text{if } \{i,j\} \in E \\ 0 & \text{else,} \end{cases}$$

and let $\mathbf{D} \in \mathbb{R}^{N \times N}$ be the diagonal matrix defined by $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$ for $i \in [N]$.

Then the normalized weighted graph Laplacian of $\mathbf{G}$ [4] is given by $\mathbf{W} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$. If we represent eigendecomposition of $\mathbf{W}$ by $\mathbf{W} = \mathbf{U}\Lambda\mathbf{U}^\top$ with the diagonal entries of Λ non-increasing, then Laplacian eigenmaps embeds $\mathbf{X}$ via $\mathbf{U}[:, 2 : d+1]$—the first d nontrivial eigenvectors of $\mathbf{W}$. The local geometry of $\mathbf{X}$ is optimally preserved in a least squares sense by the former.

A two-dimensional projection of the original input by using with k=9 is shown in the figure 3.9.

3.4.2.2 Diffusion Maps

The objective of Diffusion Maps (DM) is to define a metric, named the diffusion distance, that measures the connectivity between points in an arbitrary set. We will follow the construction of DM as described in the paper by Lindenbaum and others [9].

Representing the high dimensional input dataset $\boldsymbol{X}$, the DM framework contains the following steps:

1. A kernel function $\mathcal{K} : \boldsymbol{X} \times \boldsymbol{X} \longrightarrow \mathbb{R}$ is chosen to define an anisotropic transition matrix, represented by $\boldsymbol{K} \in \mathbb{R}^{D \times D}$. It satisfies the following properties for all $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \boldsymbol{X}$.

(a) Symmetry: $K_{i,j} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathcal{K}(\boldsymbol{x}_j, \boldsymbol{x}_i)$,
(b) Positive semi-definiteness: $\boldsymbol{v}_i^T \boldsymbol{K} \boldsymbol{v}_i \geq 0$ for all $\boldsymbol{v}_i \in \mathbb{R}^D$ and
(c) Non-Negativity $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$.

2. When we normalize the kernel using $\boldsymbol{M}$; where $M_{i,i} = \sum_j K_{i,j}$, the following matrix elements are computed:

$$P^x_{i,j} = \mathcal{P}(\boldsymbol{x}_i, \boldsymbol{x}_j) = [\boldsymbol{M}^{-1}\boldsymbol{K}]_{i,j}. \tag{3.15}$$

The resulting matrix $\boldsymbol{P}^x \in \mathbb{R}^{D\times D}$ is actually transition kernel of a Markov chain on $\boldsymbol{X}$ such that the expression $[(\boldsymbol{P}^x)^t]_{i,j} = p_t(\boldsymbol{x}_i,\boldsymbol{x}_j)$ describes the transition probability from point $\boldsymbol{x}_i$ to point $\boldsymbol{x}_j$ in t steps.

3. Next, the spectral decomposition is applied to matrix $\boldsymbol{P}^x$ to obtain a sequence of eigenvalues $\{\lambda_d\}$ and normalized eigenvectors $\{\boldsymbol{\psi}_d\}$ that satisfies $\boldsymbol{P}^x\boldsymbol{\psi}_d = \lambda_m\boldsymbol{\psi}_d, d = 0, ..., D-1$;

4. Defining a new representation for the dataset $\boldsymbol{X}$

$$\boldsymbol{\Psi}_t(\boldsymbol{x}_i) : \boldsymbol{x}_i \longmapsto \left[\lambda_1^t\psi_1[i], \lambda_2^t\psi_2[i], \lambda_3^t\psi_3[i], ..., \lambda_{D-1}^t\psi_{D-1}[i]\right]^T \in \mathbb{R}^{D-1}, \tag{3.16}$$

where t is the selected number of steps and $\psi_d[i]$ denotes the i^{th} element of $\boldsymbol{\psi}_d$.
Now the Euclidian distance between two data points is equal to the weighted L_2 distance between the conditional probabilities $p_t(\boldsymbol{x}_i, :)$, and $p_t(\boldsymbol{x}_j, :)$, $i, j = 1, ..., D$ (the i-th and j-th rows of $\boldsymbol{P}^t$), which is referred as the Diffusion Distance

$$\mathcal{D}_t^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = ||\boldsymbol{\Psi}_t(\boldsymbol{x}_i) - \boldsymbol{\Psi}_t(\boldsymbol{x}_j)||^2 = \sum_{d\geq 1} \lambda_d^{2t}(\psi_d[i] - \psi_d[j])^2 = ||p_t(\boldsymbol{x}_i, :) - p_t(\boldsymbol{x}_j, :)||^2_{\boldsymbol{W}^{-1}}, \tag{3.17}$$

where $\boldsymbol{W}$ is a diagonal matrix with elements $W_{i,i} = \frac{D_{i,i}}{\sum_{i=1}^M D_{i,i}}$.

5. The desired accuracy $\delta \geq 0$ is chosen for the diffusion distance defined by Eq. (3.17) such that $s(\delta, t) = \max\{\ell \in \mathbb{N} \text{ such that } |\lambda_\ell|^t > \delta|\lambda_1|^t\}$. By using δ, a new mapping of $s(\delta, t)$ dimensions is defined as
$\Psi_t^{(\delta)} : X \to \left[\lambda_1^t\psi_1[i], \lambda_2^t\psi_2(i), \lambda_3^t\psi_3[i], ..., \lambda_s^t\psi_s[i]\right]^T \in \mathbb{R}^{s(\delta,t)}$.

As discussed above, diffusion distance reflects the intrinsic geometry of the data set defined via the adjacency graph in a diffusion process.

3.5 Other manifold learning algorithms

Many other Laplacian based techniques exist in literature to reduce the dimensionality of a point cloud which were not dicussed in this chapter. For example,

Maximum Variance Unfolding (MVU)[3], which links most of Laplacian-based methods such as LE, LLE and Isomap. Another important approach which is derived from the LLE framework is called Hessian eigenmaps[3]. The methods is widely used when the underlying embedding is not convex. In figure 3.10 taken from Thorstensen work [1] lists the general properties for manifold learning algorithms.

3.6 Intrinsic Dimension

Intrinsic Dimension (ID) is defined as the minimal number of parameters necessary to represent the variability of a data set. It is a a key priori knowledge in computer vision and image processing to improve their performance. Mathematically, a formal definition of the intrinsic dimension is the following, due to [10].

Definition 3.1. A data set $\mathbf{X} \subseteq \mathbb{R}^{\mathbf{D}}$ said to have intrinsic dimension equal to $\mathbf{d}$ if its elements lie entirely, without information loss, within a $\mathbf{d}$-dimensional manifold of $\mathbb{R}^{\mathbf{D}}$, where $\mathbf{d} < \mathbf{D}$.

Most of the existing approach to estimate the intrinsic dimension can be roughly divided into two groups: eigenvalue or projection methods, and geometric methods. Projection or eigenvalues methods, estimate the ID by thresholding the observed eigenspectrum i.e. the spectrum of the eigenvalues output by the global or local PCA. It may be good method for exploratory data analysis, where one might plot the eigenvalues and look for a clear-cut boundary, but not for providing reliable estimates of intrinsic dimension.

The other, geometric methods exploit the intrinsic geometry of the dataset and are most often based on fractal dimensions or nearest neighbor (NN) distances [11].

Finding out the ID of the dimension estimation might become a very complex problem, when data lie on a smooth manifold. In this thesis, we used method proposed by Levina and Bickel [11] to verify the ID computed by projection and geometric methods. They derive ID by applying the principle of maximum likelihood to the distances between close neighbors.

Method	Locality	Linearity
PCA	global structure preserved	linear
MDS	global structure preserved	linear
ISOMAP	global structure preserved	nonlinear
LLE	local structure preserved	nonlinear
HLLE	local structure preserved	nonlinear
Laplacian Eigenmaps	local structure preserved	nonlinear
Diffusion maps	local structure preserved	nonlinear

Method	Strategy	special feature
PCA	preserves variance of data	based on inner product
MDS	preserve inter-point distance	based on distance function
ISOMAP	preserve the geodesics	graph based
LLE	preserve local neighbor	linear neighborhood approximation
HLLE	preserve local neighbor	theoretical guarantees
Laplacian Eigenmaps	preserve local neighbor	laplacian regularization
Diffusion maps	preserve local neighbor	wavelet approach and density independence

FIGURE 3.10: Overview of Manifold Learning Algorithms

Chapter 4

Dimension Reduction and Image Process

Chapter 5

Anomaly Detection

Chapter 6

Conclusion

6.1 Conclusions

1. The enormous blockchain (60 GB as of now) and the newer bitcoin clients indexing the full blockchain using LevelDB had made earlier public available software obsolete. The thesis develops an open source blockchain parsing tools to extract agent resolved data, which can be used to extend the stucked research in bitcoin transaction dynamics.

2. The validation of the data parsed from our tool is then checked by reproducing the "Mathew Effect" phenomenon from prominent paper using their original matlab code, but with our own data. The transaction dynamics in the bitcoin as money flow on the transaction network, support popular hypothesis in economics having roots in preferential attachment, called Matthew effect or the "rich get richer phenomenon".

3. By using the agent resolved data, the case study of silk road arrest is illustrated with the transaction graph with details. It paves the way to understand the important events in bitcoin based on transaction.

4. By defining quantitative measure of transformation over time, in terms of similarity index using different kernels, we infer that there is correspondence between network structure and exchange price in bitcoin. It also concludes that choice of graph kernels as per the graph structure plays important role on problem related to graph isomorphism.

5. We extend deep graph kernels [12], involving propagation kernels, unseen in literature, to solve graph isomorphism problem, which gives extremely agreeable results.

6. By defining quantitative measure of transformation over time, in terms of similarity index using different kernels, we infer that there is correspondence between network structure and exchange price in bitcoin, which is first step in price forecasting.

6.2 Future Work

1. On transactions data front, the future work would be to transform the data set into a simplified one indexed by user entity rather than address to do some other meaningful studies, like identifying cluster, market player and linking events to predict bubbles.

2. The possible extension of our work would be to leverage on blockchain network features, as a basis to conduct deep learning learning prediction on the price change of Bitcoin.

Appendix A

Blockchain SQL Schema

Bibliography

[1] Thorstensen, Nicolas. Manifold learning and applications to shape and image processing. *PhD Thesis, Ecole des Ponts ParisTech*. https://pastel.archives-ouvertes.fr/pastel-00005860/file/Thesis.pdf, 2009.

[2] Talwalkar, A and Kumar, S and Rowley, H. Large-scale manifold learning. *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 1-8, June, 2008.

[3] Etyngier, Patrick . Statistical learning, Shape Manifolds and Applications to Image Segmentation. *PhD thesis, Ecole Nationale des Ponts et Chausees*. http://imagine.enpc.fr/publications/papers/PhD08.pdf, 9, 41, 135, 2008.

[4] Belkin, Mikhail and Partha Niyogi. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. *Advances in Neural Information Processing Systems 14*. http://papers.nips.cc/paper MIT Press, 585–591, 2002.

[5] Boothby, W.M. : An Introduction to Differential Manifolds and Riemannian Geometry. . *Academic Press*, London, 2003.

[6] Tenenbaum, J, De Silva, V and Langford, J. A global geometric framework for nonlinear dimension reduction. *Science*, vol. 290, 2000.

[7] Roweis ST and Saul LK. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[8] Cox, Trevor and Cox, Michael. Multidimensional Scaling. *Second Edition. Chapman & Hall/CRC*, September 2000.

[9] Lindenbaum, Ofir and Yeredor, Arie and Salhov, Moshe and Averbuch, Amir. MultiView Diffusion Maps. *CoRR*. https://arxiv.org/pdf/1508.05550.pdf, 2015.

[10] Camastra, Francesco and Staiano, Antonino. Intrinsic dimension estimation: Advances and open problems. *Inf. Sci.*,328,C, 26-41, 2016.

[11] Levina, Elizaveta and Bickel, Peter. Maximum Likelihood Estimation of Intrinsic Dimension. *Advances in Neural Information Processing Systems 17*,777–784, MIT Press, 2005.

[12] Yanardag, P and Vishwanathan, S.V.N. Deep graph kernels. *In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1365–1374, New York, USA, 2015.