Clustering on an Aggregation of Sub-manifolds

by

Aurelie Jodelle Kemme

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Mathematics with Specialization in Data Science at Quantum Leap Africa (QLA), AIMS Rwanda Centre.

QUANTUM LEAP AFRICA

Quantum Leap Africa (QLA),
African Institute for Mathematical Sciences, Rwanda Center (AIMS-RW),
Remera Sector KN 3, Kigali, Rwanda.

Supervisor: **Dr. Yaé U. Gaba**

December 2022

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Quantum Leap Africa (QLA), AIMS Rwanda Centre will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Signature: ....
Aurelie Jodelle Kemme

April 6, 2023
Date: ....

Abstract

Clustering on an Aggregation of Sub-manifolds

Aurelie Jodelle Kemme
Quantum Leap Africa (QLA),
African Institute for Mathematical Sciences, Rwanda Center (AIMS-RW),
Remera Sector KN 3, Kigali, Rwanda.

Thesis: MSc. (Mathematics)
December 2022

Non-linear high dimensionality data is a real challenge when dealing with data because most algorithms are looking for linear relationships by considering the Euclidean distance as a metric. Unfortunately, more often data come as a distribution along a non-linear sub-manifold and in this case, one should consider non-linear algorithm to address such issues.
This project aims to develop an extension of the classical algorithms PCA, SVD, and MDS on a Riemannian Manifold to improve the geometry representation of the structure of the data.

Keywords: Clustering, Dimensionality Reduction, Manifold, Riemannian Manifold,PCA, SVD, MDS, Geodesic .

Acknowledgements

I would like to express my sincere gratitude to the following people and organisations ...

Dedications

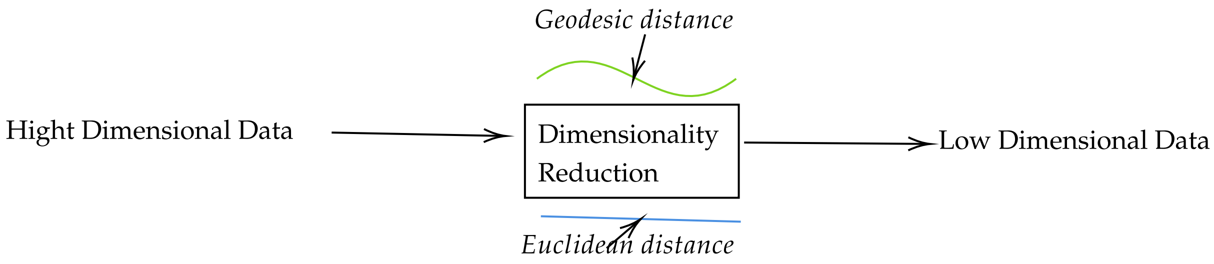
T..

Contents

List of Figures

Chapter 0

Introduction



Summary of Dimensionality Reduction

Dimensionality reduction is a powerful technique in machine learning and data analysis that aims to transform high-dimensional data into a lower-dimensional representation, while preserving as much relevant information as possible. This is particularly useful for data visualization, feature extraction, and compression, where reducing the number of dimensions can lead to faster and more efficient computation, easier interpretation, and better generalization.

One common approach to dimensionality reduction is through the aggregation of submanifolds, which involves representing high-dimensional data as a union of lower-dimensional submanifolds or clusters. This method is particularly useful when the high-dimensional data has a complex structure that can be decomposed into simpler substructures, or when the data lies on or near a low-dimensional manifold.

By aggregating submanifolds, we can reduce the dimensionality of the data by focusing on the essential features of each submanifold, rather than trying to capture all the details of the high-dimensional space. This can be achieved through various methods, such as principal component analysis (PCA), MultiDimensional Scaling, and manifold learning techniques, which aim to find a low-dimensional embedding that preserves the local or global structure of the data.

In this thesis, we will focus on three popular NLDR algorithms, namely Locally Linear Embedding (LLE), Isomap, and Hessian Locally Linear Embedding (HLLE). We will discuss the principles, advantages, and limitations of each algorithm, and compare their performance on different datasets.

In Chapter 1, we will introduce the Locally Linear Embedding algorithm, which is a popular NLDR algorithm that was proposed by Roweis and Saul in 2000. We will discuss the basic principles of LLE, including how it constructs a graph of interconnected points based on the local geometry of the data, and how it uses linear regression to estimate the low-dimensional representation of the data. We will also discuss the advantages and limitations of LLE, and compare its performance to other NLDR algorithms.

In Chapter 2, we will introduce the Isomap algorithm, which is another popular NLDR algorithm that was proposed by Tenenbaum et al. in 2000. We will discuss the basic principles of Isomap, including how it constructs a graph of interconnected points based on the geodesic distance between the points, and how it uses multidimensional scaling to estimate the low-dimensional representation of the data. We will also discuss the advantages and limitations of Isomap, and compare its performance to other NLDR algorithms.

In Chapter 3, we will introduce the Hessian Locally Linear Embedding algorithm, which is a recent extension of LLE that was proposed by Coifman and Lafon in 2006. We will discuss the basic principles of HLLE, including how it uses the Hessian matrix of the distance function to estimate the local geometry of the data, and how it uses nonlinear optimization to estimate the low-dimensional representation of the data. We will also discuss the advantages and limitations of HLLE, and compare its performance to LLE and other NLDR algorithms.

Chapter 1

Linear Methods for Dimensionality Reduction

This chapter explains the mathematics Concepts behind some linear machine learning algorithms for dimensionality reduction. Some application using python and some of limitations are provided. The chapter focus on four principal algorithms namely Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Multi-Dimensional Scaling, and Non-Negative Matrix Factorization (NMF). A brief description of Independent Component Analysis (ICA), and Linear Discriminant Analysis (LDA) is also provided.

1.0 Principal Component Analysis (PCA)

The Principal Component Analysis is a features extraction linear dimensionality reduction machine learning algorithm. We use PCA to reduce the dimension of a large dataset into a lower one without losing meaningful information about the large dataset. Reducing variables in a dataset while preserving as much as possible meaningful insight makes the data exploration and analysis more simple, improves the performance, and make the processing faster. When it comes to data analysis, we have what we generally call data pre-processing and data processing. The data pre-processing and processing are the fact of cleaning data (dealing with extremes and missing values) and also highlighting relevant features. In mathematics, we can generalize those methods into four steps: the standardization of the data, the computation of the covariance matrix, the computation of eigenvectors and eigenvalues, and the construction of principal components.

1.0.0 Standardization

Standardization is a way of putting data on the same scale. The goal of the standardization of the data is to make features within the data equally contribute during the data analysis. Note that the standardization will take only into account continuous variables because they change over time. Standardizing the data will re-scale it to avoid bias.

Coming to mathematics, re-scaling a data or standardizing a data means bringing the standard deviation to 1 if we use the Z-score method and between [-3 3] if we use the row-min, row-max methods. When the different between the minimum and the maximum within the data is too large, it's not recommended to use the row-min, row-max methods because the maximum will scale down all the data values. In data analysis, the standard deviation will provide us the information on how dispersed the data are in relation to the mean. A low standard deviation will mean the data are clustered around the mean and a high standard deviation will mean the data are spread out.

Let **X** be our sample data. We can standardize our data using the following formula:

$$\mathbf{z} = \frac{value - mean}{standard\ deviation} = \frac{\mathbf{x}_i - \bar{\mathbf{X}}}{\sigma} \tag{1.0.1}$$

where,

$$\bar{\mathbf{X}} = \frac{1}{n}\sum \mathbf{x}_i$$

is the mean of the data and it used to measure the performance in general, $\mathbf{x_i}$ is our data points, n is the sample size and,

$$\sigma = \frac{1}{\sqrt{n-1}}\sqrt{\sum(\mathbf{x}_i - \bar{\mathbf{X}})^2}$$

is the standard deviation.
After re-scaling the data, we want to look at the correlation between features and to do that, we will build the similarity measure matrix using the co-variance or the correlation measure.

1.0.1 Computation of the co-variance matrix

The co-variance values between each couple of variables within the data are inputs of the covariance matrix.

We use the co-variance matrix to measure the degree of relationship between features within the data. Because depending on the relationship between the features, we can have strongly positive correlated, strongly negative correlated, or noncorrelated features. To avoid redundancy information, when two features are strongly positively correlated, it's better to consider only one of them for the analysis.

1.0.1.1 Formula of the co-variance matrix for 3-dimensionality data

Let's take the case of 3-demensionality data: $\mathbf{X} = (\mathbf{x_1}, \mathbf{y_1}, \mathbf{z_1})$

$$\textbf{Covariance Matrix} = \begin{bmatrix} COV(\mathbf{x}_1,\mathbf{x}_1) & COV(\mathbf{x}_1,\mathbf{y}_1) & COV(\mathbf{x}_1,\mathbf{z}_1) \\ COV(\mathbf{y}_1,\mathbf{x}_1) & COV(\mathbf{y}_1,\mathbf{y}_1) & COV(\mathbf{y}_1,\mathbf{z}_1) \\ COV(\mathbf{z}_1,\mathbf{x}_1) & COV(\mathbf{z}_1,\mathbf{y}_1) & COV(\mathbf{z}_1,\mathbf{z}_1) \end{bmatrix} \tag{1.0.2}$$

Where,

$$COV(\mathbf{x},\mathbf{y}) = \frac{1}{n-1}\sum(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{y} - \bar{\mathbf{y}})$$

Using the correlation matrix, we will compute eigenvalues and eigenvectors.

1.0.2 Computation of Eigenvalues and Eigenvectors

Let **X** be an *n*- dimensionality data and let **A** be it covariance matrix.

Definition 1.0.1. ***(Eigenvalues)***
An eigenvalue of **A** *is a real number* $\lambda \in \mathbf{R}$ *such that* λ *is a solution of the equation* **det** *(*λ $\mathbf{I}_n$*-* **A***)=0.*

The set of all eigenvalues of **A** is called spectrum of **A** and noted $\sigma(\mathbf{A})$

$$\sigma(\mathbf{A}) = \left\{ \lambda \in \mathbf{R} \ \mid \ \mathbf{det}(\lambda \mathbf{I}_n - \mathbf{A}) = 0 \right\} \tag{1.0.3}$$

Definition 1.0.2. ***(Eigenvectors)***
An eigenvector of **A** *associated to the eigenvalue* $\lambda \in \mathbf{R}$ *is a real vector* $\mathbf{x} \in \mathbf{R}^{*n}$ *such that* **x** *is a solution of the equation* $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$*.*

Let V be a set of all eigenvectors. V can be defined as

$$\mathrm{V} = \left\{ \mathbf{x} \in \mathbf{R}^n \ \mid \ \mathbf{A}\mathbf{x} = \lambda \mathbf{x} \right\} \tag{1.0.4}$$

Examples 1.0.3. ***(Eigenvalues and eigenvectors in 3-dimensionality data.)***
Let **X** *be an* 3*- dimensionality data and*

$$\mathbf{A} = \begin{pmatrix} 5 & -10 & -5 \\ 2 & 14 & 2 \\ -4 & -8 & 6 \end{pmatrix} \tag{1.0.5}$$

be it covariance matrix.

1.0.2.1 Eigeinvalues

$$\sigma(\mathbf{A}) = \{\lambda \in \mathbf{R} \ \mid \ \boldsymbol{det}(\lambda \mathbf{I}_3 - \mathbf{A}) = 0\} \tag{1.0.6}$$

$$\boldsymbol{det}(\lambda \mathbf{I}_3 - \mathbf{A}) = \boldsymbol{det}\left(\lambda \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 5 & -10 & -5 \\ 2 & 14 & 2 \\ -4 & -8 & 6 \end{pmatrix}\right)$$

$$= \begin{vmatrix} \lambda - 5 & 10 & 5 \\ -2 & \lambda - 14 & -2 \\ 4 & 8 & \lambda - 6 \end{vmatrix}$$

$$= \Big(\lambda - 5\Big)\Big[\Big(\lambda - 14\Big)\Big(\lambda - 6\Big) + 16\Big] - 10\Big[-2\Big(\lambda - 6\Big) + 8\Big] + 5\Big[-16 - 4\Big(\lambda - 14\Big)\Big]$$

$$= \Big(\lambda - 5\Big)\Big[\lambda^2 - 20\lambda + 100\Big] + 20\Big[\lambda - 10\Big] - 20\Big[\lambda - 10\Big]$$

$$\boldsymbol{det}(\lambda \mathbf{I}_3 - \mathbf{A}) = \Big(\lambda - 5\Big)\Big(\lambda - 10\Big)^2$$

$$\begin{aligned} \boldsymbol{det}(\lambda \mathbf{I}_3 - \mathbf{A}) = 0 &\implies \Big(\lambda - 5\Big)\Big(\lambda - 10\Big)^2 = 0 \\ &\implies \lambda = 5 \ \ or \ \ \lambda = 10 \end{aligned}$$

So,

$$\sigma(\mathbf{A}) = \{5, 10\} \tag{1.0.7}$$

1.0.2.2 Eigenvectors

We will compute eigenvectors associated to each eigenvalue

$$\mathrm{V} = \Big\{\mathbf{x} \in \mathbf{R}^3 \ \mid \ \mathbf{A}\mathbf{x} = \lambda \mathbf{x} \ \ with \ \ \lambda \in \{5, 10\}\Big\}$$

Let $\mathbf{x} = (x_1, x_2, x_3)$

$$(\mathbf{A} - \lambda I_3)\mathbf{x} = 0 \implies \begin{pmatrix} \lambda - 5 & 10 & 5 \\ -2 & \lambda - 14 & -2 \\ 4 & 8 & \lambda - 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

$$\implies \begin{cases} (\lambda - 5)x_1 + 10x_2 + 5x_3 = 0 \\ -2x_1 + (\lambda - 14)x_2 - 2x_3 = 0 \\ 4x_1 + 8x_2 + (\lambda - 6)x_3 = 0 \end{cases}$$

$$\lambda = 5 \implies \begin{cases} +10x_2 + 5x_3 = 0 \\ 2x_1 + 9x_2 + 2x_3 = 0 \\ 4x_1 + 8x_2 - x_3 = 0 \end{cases}$$

$$\implies \begin{cases} x_3 = -2x_2 \\ x_1 = -\frac{5}{2}x_2 \end{cases}$$

$$\implies (x_1, x_2, x_3) = x_2(-\frac{5}{2}, 1, -2)$$

$$\lambda = 10 \implies x_1 + 2x_2 + x_3 = 0$$

$$\implies (x_1, x_2, x_3) = x_1(1, 0, -1) + x_2(0, 1, -2)$$

$$\text{V} = \left\{ (-\frac{5}{2}, 1, -2); (1, 0, -1); (0, 1, -2) \right\} \tag{1.0.8}$$

1.0.2.3 What is the usefulness of eigenvalues and eigenvectors in PCA?

We use eigenvalues and eigenvectors to identify principal components.
The eigenvectors give us the direction of the axes where we have the most information and eigenvalues will give the amount of information in each principal component.

1.0.3 Principal Components

Definition 1.0.4. *Principal Components are new features constructed as a linear combination of initial features such that they are uncorrelated and most of the information is stored in the first component.*

1.0.3.1 How to choose the optimal number of Principal Components?

To choose the optimal number of principal components, it is recommended to look at the plot showing the **cumulative explained variance ratio in function of number of components**.

Examples 1.0.5. *content...*

1.0.4 Reconstructing the new data-frame

Let consider a m-dimensionality data $\mathbf{X}$ and let $\mathbf{V}$ be its vectors features. $\mathbf{V}$ and it transpose $\mathbf{V}^t$ are square matrix of dimension m, $\mathbf{X}$ is an $n \times m$ matrix and it transpose $\mathbf{X}^t$ is an $m \times n$ matrix where n is the sample size of the data.
The reconstruction of the new data consists in multiplying the transpose of the features vectors with the transpose of the initial data and we then obtain:

$$New\ dataframe \ = \ \mathrm{V}^{\mathbf{t}} \times \mathbf{X}^{\mathbf{t}}.$$

From the example 1.0.3, the initial data is given by the equation 1.0.5 and the features vectors are given by the equation 1.0.8.
The transpose of the initial data is as follow:

$$\mathbf{A}^t = \begin{pmatrix} 5 & 2 & -4 \\ -10 & 14 & -8 \\ 5 & 2 & 6 \end{pmatrix}.$$

The features vectors are

$$\mathrm{V} = \begin{pmatrix} -\frac{5}{2} & 0 & 1 \\ 1 & 1 & 0 \\ -2 & -2 & -1 \end{pmatrix} \quad and\ its\ transpose\ are, \quad \mathrm{V}^t = \begin{pmatrix} -\frac{5}{2} & 1 & -2 \\ 0 & 1 & -2 \\ 1 & 0 & -1 \end{pmatrix}.$$

The reconstructed data is

$$\mathrm{V}^{\mathbf{t}} \times \mathbf{A}^{\mathbf{t}} = \begin{pmatrix} -\frac{5}{2} & 1 & -2 \\ 0 & 1 & -2 \\ 1 & 0 & -1 \end{pmatrix} \times \begin{pmatrix} 5 & 2 & -4 \\ -10 & 14 & -8 \\ 5 & 2 & 6 \end{pmatrix} = \begin{pmatrix} \frac{5}{2} & -35 & 20 \\ -5 & 16 & -12 \\ -5 & -2 & 14 \end{pmatrix}. \tag{1.0.9}$$

1.0.5 Recap of usefulness of Principal Components Analysis

- PCA helps to remove correlated features. When working with high dimensionality data, the best way to improve the algorithm performance is to highlight only the variables useful for the analysis. One way to do that is to remove correlated features but it almost impossible to do it hand by hand. So, PCA helps solve this problem;

- PCA improves the algorithm performance. Data with many variables generally leads to overitting. Reducing the size of variables will considerably improve the performance of the algorithm and will also reduce overfitting;

- High dimensionality data is very hard to visualize. Scaling down the size of data will help improve data visualization and that's where PCA stands.

1.0.6 Weaknesses of Principal Components Analysis

- PCA is very sensitive to the sparsity of the data so, before applying the PCA, the standardization of the data must be done and for doing that, all categorical variables must be converted into numerical variables;
- PCA can perform well only if there is a linear dependence between the variables;
- PCA transforms correlated features into a set of linear uncorrelated principal components that are linear combination of initial features. Uncorrelated principal components or new features are less interpretable than the original ones;
- PCA may lead to some loss of information because of the choice of the number of principal components to select;
- PCA doesn't work when there is weak relationship between the features within the data.

1.0.7 Implementation of PCA using Python

1.1 Singular Values Decomposition (SVD)

The Singular Values Decomposition (SVD) is the most powerful matrix decomposition from numerical linear algebra used for high dimensionality data reduction. SDV reduces the dimension of the data by extracting the key relevant features for data analysis and data visualization.
The Singular Values Decomposition uses the concept of orthogonality to decompose high dimensional matrix into lower ones.

1.1.0 Orthogonality

Definition 1.1.1. ***[Inner product]*** *Let $\mathcal{V}$ be a vector space of finite dimension n. An inner product is a function $<,>$ which associate two vectors u, v which a scalar $< u, v >$, and which satisfies the following properties:*

- $< u, u > \geq 0$ *(positiveness);*
- $< u, u >= 0 \iff u = 0$
- $< u, v >=< v, u >$ *(symmetric);*
- $< \alpha u + v, w >= \alpha < u, v > + < v, w >$ *(bi-linearity).*

If the function $<,>$ is an inner product, and u and v are two vectors of a vector space $\mathcal{V}$ then, we have the following definition:

1. The scalar product of u and v is defined by $< u, v >= \|u\| \times \|v\| \times cos(u, v)$
2. The norm or length of u is noted $\|u\|$ and defined by $\|u\| = \sqrt{< u, u >}$;
3. The distance between u and v is $\|u - v\|$;
4. If $< u, v >= 0$, then u and v are said to be orthogonal;
5. The orthogonal projection of v spanned by u is equal to $p = \left(\frac{<u,v>}{<u,u>}\right) u$.

Two vectors $u \in \mathcal{V}$ and $v \in \mathcal{V}$ are said to be orthogonal if their inner products $< u, v >$ is equal to zero.

$$u \perp v \Longleftrightarrow < u, v >= 0$$

Definition 1.1.2. ***[Unitary matrix]*** *A matrix* $\mathbf{A}$ *is an unitary matrix if its transpose is equal to its inverse* ($\mathbf{A}^t = \mathbf{A}^{-1}$)*. In other words,* $\mathbf{A}$ *is an unitary matrix if multiplying by its transpose we obtain the identity matrix.*

$$\mathbf{A} \textbf{ unitary matrix} \quad \Longrightarrow \quad \mathbf{A}\mathbf{A}^t = \mathbf{I_n} \quad \textit{where } \mathbf{I_n} \textit{ is the identity matrix.} \tag{1.1.1}$$

Examples 1.1.3. ***[Unitary matrix]***

Definition 1.1.4. ***[Orthogonal matrix]*** *A matrix* $\mathbf{A}$ *is said to be orthogonal if* $\mathbf{A}$ *is an unitary matrix and its entries* $\mathbf{A}_{ij}$ *are real values. Equivalently,* $\mathbf{A}$ *is said to be orthogonal if all its columns and rows are orthogonal unit vectors.*

Examples 1.1.5. ***[Orthogonal matrix]***

1.1.1 Full SDV of a matrix

Definition 1.1.6. ***[Left eigenvectors]*** $\mathbf{X}_L$ *is a left eigenvector of* $\mathbf{A}$ *if* $\mathbf{X_L}\mathbf{A} = \lambda_\mathbf{L}\mathbf{X_L}$ *with* $\lambda_\mathbf{L}$ *the associated eigenvalue;*

Examples 1.1.7. ***[Left eigenvectors]*** *using the same matrix*

$$\mathbf{A} = \begin{pmatrix} 5 & -10 & -5 \\ 2 & 14 & 2 \\ -4 & -8 & 6 \end{pmatrix} \quad \textit{as in the example1.0.5}, \tag{1.1.2}$$

we have the spectrum of $\mathbf{A}$ *is given by the equation 1.1.8.*
We will compute the left eigenvectors associated to each eigenvalue

$$\mathrm{V_L} = \left\{ \mathbf{x} \in \mathbf{R}^3 \;\mid\; \mathbf{x}\mathbf{A} = \lambda\mathbf{x} \;\; with \;\; \lambda \in \{5, 10\} \right\}$$

Let $\mathbf{x} = (x_1, x_2, x_3)$

$$\mathbf{x}(\mathbf{A} - \lambda I_3) = 0 \implies \begin{pmatrix} x_1, x_2, x_3 \end{pmatrix} \begin{pmatrix} \lambda - 5 & 10 & 5 \\ -2 & \lambda - 14 & -2 \\ 4 & 8 & \lambda - 6 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

$$\implies \begin{cases} (\lambda - 5)x_1 - 2x_2 + 4x_3 = 0 \\ 10x_1 + (\lambda - 14)x_2 + 8x_3 = 0 \\ 5x_1 - 2x_2 + (\lambda - 6)x_3 = 0 \end{cases}$$

$$\lambda = 5 \implies \begin{cases} -2x_2 + 4x_3 = 0 \\ 10x_1 - 9x_2 + 8x_3 = 0 \\ 5x_1 - 2x_2 - x_3 = 0 \end{cases}$$

$$\implies \begin{cases} x_2 = 2x_3 \\ x_1 = x_2 \end{cases}$$

$$\implies (x_1, x_2, x_3) = x_1(1, 2, 1)$$

$$\lambda = 10 \implies 5x_1 - 2x_2 + 4x_3 = 0$$

$$\implies (x_1, x_2, x_3) = x_1(1, \frac{5}{2}, 0) + x_3(0, 2, 1)$$

$$\mathrm{V_L} = \left\{ (1, 2, 1); (1, \frac{5}{2}, 0); (0, 2, 1) \right\} \tag{1.1.3}$$

Definition 1.1.8. ***[Right eigenvectors]***

$\mathbf{X}_R$ *is a right eigenvector of* $\mathbf{A}$ *if* $\mathbf{AX_R} = \lambda_\mathbf{R}\mathbf{X_R}$ *with* $\lambda_\mathbf{R}$ *the associated eigenvalue.*

Examples 1.1.9. ***[Right eigenvectors]*** *An example of computation of right eigenvectors is given in the sub-section 1.0.8.*

Let $\mathbf{A}$ be an $n \times m$ data matrix. We use SDV method to decompose $\mathbf{A}$ such that

$$\mathbf{A} = \mathbf{UDV^t} \tag{1.1.4}$$

where $\mathbf{U}$ is an $n \times n$ unitary matrix with elements the left eigenvectors of $\mathbf{A}$ and it gives the amount of information containing in $\mathbf{A}$. $\mathbf{D}$ is an $n \times m$ diagonal matrix whose entries are square root of eigenvalues of $\mathbf{A^T A}$ entered in decreasing order. $\mathbf{D}$ is also called singular values and give the information on how variance are spread around each singular vector;

$\mathbf{V}$ is an $m \times m$ unitary matrix with elements the right eigenvectors of $\mathbf{A}$ and it shows how the information containing in $\mathbf{A}$ move in time.
Note that in the matrix decomposition given by the equation [1.1.4], $\mathbf{U}$ is the singular vector matrix for $\mathbf{AA^T}$, $\mathbf{V}$ is the singular vector matrix for $\mathbf{A^TA}$, and $\mathbf{D}$ is the singular values of $\mathbf{A^TA}$.
In fact

$$\begin{aligned}\mathbf{AA^t} &= (\mathbf{UDV^t})(\mathbf{UDV^t})^{\mathbf{t}} \\ &= \mathbf{UDV^tVD^tU^t} \quad \textit{since}\ \ \mathbf{V}\ \textit{is an unitary matrix we have}\ \mathbf{VV^t} = \mathbf{V^tV} = \mathbf{I_n} \\ &= \mathbf{UDD^tU^t} \quad \textit{So}\ \mathbf{U}\ \textit{is the singular vector matrix for}\ \mathbf{AA^t}.\end{aligned}$$

And

$$\begin{aligned}\mathbf{A^tA} &= (\mathbf{UDV^t})^{\mathbf{t}}(\mathbf{UDV^t}) \\ &= \mathbf{VD^tU^tUDV^t} \quad \textit{since}\ \ \mathbf{U}\ \textit{is an unitary matrix we have}\ \mathbf{UU^t} = \mathbf{U^tU} = \mathbf{I_n} \\ &= \mathbf{VDD^tV^t} \quad \textit{So}\ \mathbf{V}\ \textit{is the singular vector matrix for}\ \mathbf{A^tA}.\end{aligned}$$

The full SVD method provides a good approximation of the data point but, we can improve this approximation by using the **economy or thin svd** given by the decomposition

$$\mathbf{A} = \sum_{i=1}^{min(n,m)} \sigma_i \mathbf{u_i} \mathbf{v_i^t} \tag{1.1.5}$$

where u_i with $i = 1, ..., n$ denote the columns of the matrix $\mathbf{U}$, v_j with $j = 1, ..., m$ denote the columns of the matrix $\mathbf{V}$, and σ_i with $j = 1, ..., min(n, m)$ denote the singular values such that,

$$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_{min(n,m)}.$$

If $n > m$ then, we have m singular values and ifnot we have n singular values.
The first column $\mathbf{u_1}$ of $\mathbf{U}$ corresponding to the singular value σ_1 and the first column $\mathbf{v_1}$ of $\mathbf{V^t}$ corresponding also to the singular value σ_1 are more important in describing the information containing in $\mathbf{A}$ than the remaining columns of matrices $\mathbf{U}$ and $\mathbf{V^t}$.
We can furthermore improve by taking the rank r of the matrix $\mathbf{A}$. We define the rank of a matrix $\mathbf{A}$ as the number of its non zero linear independent vectors.
By truncating the dimension of the SVD of $\mathbf{A}$ at r, we obtain

$$\mathbf{A} = \mathbf{UDV^t} = \sum_{i=1}^{r} \sigma_i \mathbf{u_i} \mathbf{v_i^t} \tag{1.1.6}$$

with $\mathbf{U}$ an $n \times r$ matrix, $\mathbf{D}$ an $r \times r$ matrix, and $\mathbf{U}$ an $r \times m$ matrix.

If $\mathbf{A}$ is an invertible or non singular matrix, then $n = m = r$ and its economy SVD is equal to its full SVD.

1.1.1.1 SVD Implementation steps

- Computation of symmetric matrices $\mathbf{AA^t}$ and $\mathbf{A^tA}$;

- Computation of eigenvalues of the matrix $\mathbf{A^tA}$;
- Computation of **U** which entries are eigenvectors of the matrix $\mathbf{AA^t}$ and normalized by its norms;
- Computation of **V** which entries are eigenvectors of the matrix $\mathbf{A^tA}$ and normalized by its norms;
- Computation of **D** which entries are square root of eigenvalues of the matrix $\mathbf{AA^t}$.

Examples 1.1.10. *SVD of the matrix*

$$\mathbf{A} = \begin{pmatrix} 5 & -10 & -5 \\ 2 & 14 & 2 \\ -4 & -8 & 6 \end{pmatrix} \quad \textit{same matrix using in the example1.0.5.} \tag{1.1.7}$$

1.1.1.2 Step 1: getting symmetric matrices

The first step consists to convert the matrix **A** *into a symmetric matrix by multiplying it with it transpose* $\mathbf{A^t}$

$$\mathbf{A^tA} = \begin{pmatrix} 5 & 2 & -4 \\ -10 & 14 & -8 \\ -5 & 2 & 6 \end{pmatrix} \times \begin{pmatrix} 5 & -10 & -5 \\ 2 & 14 & 2 \\ -4 & -8 & 6 \end{pmatrix} = \begin{pmatrix} 45 & 10 & -45 \\ 10 & 360 & 30 \\ -45 & 30 & 65 \end{pmatrix}$$

$$\mathbf{AA^t} = \begin{pmatrix} 5 & -10 & -5 \\ 2 & 14 & 2 \\ -4 & -8 & 6 \end{pmatrix} \times \begin{pmatrix} 5 & 2 & -4 \\ -10 & 14 & -8 \\ -5 & 2 & 6 \end{pmatrix} = \begin{pmatrix} 150 & -140 & 30 \\ -140 & 204 & -108 \\ 30 & -108 & 116 \end{pmatrix}$$

1.1.1.3 Step 2: Eigenvalues

The second step consists on computing the eigenvalues of the symmetric matrix. The spectrum of $\mathbf{A^tA}$ *is given by the equation*

$$\sigma(\mathbf{A^tA}) = \{\lambda \in \mathbf{R} \mid \boldsymbol{det}(\lambda \mathbf{I}_3 - \mathbf{A^tA}) = 0\}$$

$$\boldsymbol{det}(\lambda \mathbf{I}_3 - \mathbf{A}^{\mathbf{t}}\mathbf{A}) = \boldsymbol{det}\left(\lambda \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 45 & 10 & -45 \\ 10 & 360 & 30 \\ -45 & 30 & 65 \end{pmatrix}\right)$$

$$= \begin{vmatrix} \lambda - 45 & -10 & 45 \\ -10 & \lambda - 360 & -30 \\ 45 & -30 & \lambda - 65 \end{vmatrix}$$

$$= -\lambda^3 + 470\lambda^2 - 39500\lambda + 250000$$

$$\sigma(\mathbf{A}) = \left\{185 - 15\sqrt{141}, 185 + 15\sqrt{141}, 100\right\} \tag{1.1.8}$$

1.1.1.4 Step 3: Eigenvectors of $\mathbf{A}^{\mathbf{t}}\mathbf{A}$

We will compute the eigenvectors associated to each eigenvalue

$$\mathrm{V} = \left\{\mathbf{x} \in \mathbf{R}^3 \mid \mathbf{A}^{\mathbf{T}}\mathbf{A}\mathbf{x} = \lambda \mathbf{x} \ \textit{with} \ \lambda \in \{185 - 15\sqrt{141}, 185 + 15\sqrt{141}, 100\}\right\}$$

For $\lambda = 185 + 15\sqrt{141}$,

$$V_1 = \begin{bmatrix} 1 \\ \frac{12\sqrt{141}+142}{5} \\ \frac{\sqrt{141}+16}{5} \end{bmatrix}$$

For $\lambda = 100$,

$$V_2 = \begin{bmatrix} 1 \\ \frac{1}{10} \\ -\frac{6}{5} \end{bmatrix}$$

For $\lambda = 185 - 15\sqrt{141}$,

$$V_3 = \begin{bmatrix} 1 \\ -\frac{12\sqrt{141}-142}{5} \\ -\frac{\sqrt{141}+16}{5} \end{bmatrix}$$

So,

$$\mathbf{V} = \begin{bmatrix} \frac{1}{\|V_1\|} & \frac{1}{\|V_2\|} & \frac{1}{\|V_3\|} \\ \frac{12\sqrt{141}+142}{5\|V_1\|} & \frac{1}{10\|V_2\|} & -\frac{12\sqrt{141}-142}{5\|V_3\|} \\ \frac{\sqrt{141}+16}{5\|V_1\|} & -\frac{6}{5\|V_2\|} & -\frac{\sqrt{141}+16}{5\|V_3\|} \end{bmatrix} \quad and \quad \mathbf{V^t} = \begin{bmatrix} \frac{1}{\|V_1\|} & \frac{12\sqrt{141}+142}{5\|V_1\|} & \frac{\sqrt{141}+16}{5\|V_1\|} \\ \frac{1}{\|V_2\|} & \frac{1}{10\|V_2\|} & -\frac{6}{5\|V_2\|} \\ \frac{1}{\|V_3\|} & -\frac{12\sqrt{141}-142}{5\|V_3\|} & -\frac{\sqrt{141}+16}{5\|V_3\|} \end{bmatrix}$$

1.1.1.5 Step 4: Eigenvectors of $\mathbf{AA^t}$

We will compute the eigenvectors associated to each eigenvalue

$$\mathrm{U} = \left\{ \mathbf{x} \in \mathbf{R}^3 \mid \mathbf{AA^t x} = \lambda \mathbf{x} \ \ with \ \ \lambda \in \{185 - 15\sqrt{141}, 185 + 15\sqrt{141}, 100\} \right\}$$

For $\lambda = 185 + 15\sqrt{141}$,

$$U_1 = \begin{bmatrix} 1 \\ -\frac{6\sqrt{141}+4}{55} \\ -\frac{\sqrt{141}-91}{110} \end{bmatrix}$$

For $\lambda = 100$,

$$U_2 = \begin{bmatrix} 1 \\ \frac{1}{10} \\ -\frac{6}{5} \end{bmatrix}$$

For $\lambda = 185 - 15\sqrt{141}$,

$$U_3 = \begin{bmatrix} 1 \\ \frac{6\sqrt{141}-4}{55} \\ \frac{\sqrt{141}+91}{110} \end{bmatrix}$$

So,

$$\mathbf{U} = \begin{bmatrix} \frac{1}{\|U_1\|} & \frac{1}{\|U_2\|} & \frac{1}{\|U_3\|} \\ -\frac{6\sqrt{141}+4}{55\|U_1\|} & \frac{1}{10\|U_2\|} & \frac{6\sqrt{141}-4}{55\|U_3\|} \\ -\frac{\sqrt{141}-91}{110\|U_1\|} & -\frac{6}{5\|U_2\|} & \frac{\sqrt{141}+91}{110\|U_3\|} \end{bmatrix}$$

1.1.1.6 Step 5: Singular values of $\mathbf{AA^t}$

The entries are square roots of eigenvalues of $\mathbf{AA^t}$

$$\mathbf{D} = \begin{bmatrix} \sqrt{185+15\sqrt{141}} & 0 & 0 \\ 0 & \sqrt{100} & 0 \\ 0 & 0 & \sqrt{185-15\sqrt{141}} \end{bmatrix}$$

$$\mathbf{A} = \mathbf{UDV^t} = \begin{bmatrix} \frac{1}{\|U_1\|} & \frac{1}{\|U_2\|} & \frac{1}{\|U_3\|} \\ -\frac{6\sqrt{141}+4}{55\|U_1\|} & \frac{1}{10\|U_2\|} & \frac{6\sqrt{141}-4}{55\|U_3\|} \\ -\frac{\sqrt{141}-91}{110\|U_1\|} & -\frac{6}{5\|U_2\|} & \frac{\sqrt{141}+91}{110\|U_3\|} \end{bmatrix} \times \begin{bmatrix} \sqrt{185+15\sqrt{141}} & 0 & 0 \\ 0 & \sqrt{100} & 0 \\ 0 & 0 & \sqrt{185-15\sqrt{141}} \end{bmatrix} \times$$

$$\begin{bmatrix} \frac{1}{\|V_1\|} & \frac{12\sqrt{141}+142}{5\|V_1\|} & \frac{\sqrt{141}+16}{5\|V_1\|} \\ \frac{1}{\|V_2\|} & \frac{1}{10\|V_2\|} & -\frac{6}{5\|V_2\|} \\ \frac{1}{\|V_3\|} & -\frac{12\sqrt{141}-142}{5\|V_3\|} & -\frac{\sqrt{141}+16}{5\|V_3\|} \end{bmatrix}.$$

Examples 1.1.11.

1.2 MultiDimensional Scaling (MDS)

Multidimensional Scaling (MDS) is a set of algorithms where each of them is used to capture a manifold representation [?] [1] for a given data set. Most of big data analysts use MDS to understand the key factors considered by their customers while evaluating products, service or company.

MDS can be used in some companies to identify keys factors that contribute to the evaluation of their products, services, and company by their customers. MDS can help those companies identify the number of factors and how significant each factor is for an evaluation of a service or product in a particular situation. MDS will turn respondent evaluation into distances represented in multi-dimensional spaces.

[1] A manifold representation of data is a mapping of original data points into a lower dimensional manifold in such away that pairwise distances between original data points are close as possible to the pairwise distances between those into the lower dimensional manifold.

Since MDS seeks to measure pairwise distance between objects, we use square matrix to implement the multidimensional scaling method and elements in the matrix are dissimilarities between two objects.

To build an MDS matrix, the first step is to compute distances between each pair of points. Then, try to build an algorithm that will attempt to solve an optimization problem.

The related optimization problem can be define of finding a set of Euclidean coordinates in a lower dimensional space such that the pairwise distances between those coordinates are closer as possible to pairwise distances of the original coordinates in the high dimensional space.

Assume we have a data $\mathbf{X}$ of dimension $n \times m$, the dimension of the MDS matrix is equal to $m \times m$ since we have m features and n observations.

Let denote by σ_{ij} the dissimilarity matrix of X whose entries are distances between each pair of points x_i and x_j with $x_i, x_j \in \mathbf{X}$.

where,

$$\sigma_{ij} = d(x_i, x_j) \text{ is an Euclidean distance.}$$

Examples 1.2.1. *Let's take an example in the banking sector. We can make a survey in a bank to find out whether or not customers will upgrade for a given product or service. Assume n customers have filled the survey for m products.*

The cell (i, j) of a customer i upgrading for a product j is filled with a 1 if a customer has upgraded, otherwise, it filled with a 0 if not.

	product 1	product 2	⋯	product m
customer 1	0	1	⋯	1
customer 2	1	0	⋯	1
⋯	⋯	⋯	⋯	⋯
customer n	0	1	⋯	0

Figure 1.1: Dataframe obtained from the survey.

After familiarizing our self with the contain of the figure 1.1, we focus on the products and the obtained matrix is,

$$\begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 1 & \cdots & 0 \end{pmatrix} \tag{1.2.1}$$

From the matrix 1.2.1, let's derive a 2×2 dissimilarity matrix for each pair of products.

That dissimilarity matrix can be seen as a confusion matrix [2] *given by the figure 1.2.*

[2] Here the confusion matrix is seen as a matrix where the YES represents the cases where customers upgraded for a given product and NO represents the cases where customers didn't upgrade for a given product

		product2	
		Yes	No
product 1	Yes	TP	FN
	No	FP	TN

Figure 1.2: Matrix showing the summary of the survey between product 1 and 2.

with,
TP is the total number of time customers have upgraded both products 1 and 2.
TN is the total number of time customers neither upgraded for product 1 nor product 2.
FP is the total number of time customers have upgraded for product 2 and not for product 1.
FN is the total number of time customers have upgraded for product 1 and not for product 2.
The next step is to compute the dissimilarity score and for doing that, we will consider only situations where customers have upgraded for at least one product between each pair of products. In this case, we can apply the Jaccard dissimilarity measure given by:

$$J(product\ 1, product\ 2) = \frac{FP + FN}{TP + FP + FN} = J_{12} = J_{21}. \tag{1.2.2}$$

Where, $J(1, 2)$ is the dissimilarity distance[3] *between product 1 and 2.*
We do the same computation for each pair of products and we then build an $m \times m$ dissimilarity matrix whose entries correspond to Jaccard dissimilarity score.

	product 1	*product 2*	$\cdots$	*product m*
product 1	J_{11}	J_{12}	$\cdots$	J_{1m}
product 2	J_{21}	J_{22}	$\cdots$	J_{2m}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
product m	J_{m1}	J_{m2}	$\cdots$	J_{mm}

With J_{ik} the Jaccard dissimilarity score corresponding to the couple of products i and k.
If $i = k$, we have $J_{ik} = 0$ because of $FP = FN = 0$.

In general, let denote by σ_{ij} the dissimilarity measure between two objects $x_i = (x_{i1}, x_{i2}, ..., x_{im})^t$ and $x_j = (x_{j1}, x_{j2}, ..., x_{jm})^t$ with $i, j = 1, 2, ..., n$ of matrix space $\mathbf{X}$.
Where,

$$\sigma_{ij} = \|x_i - x_j\| = \left(\sum_{k=1}^{m} (x_{ik} - x_{jk})^2 \right)^{1/2}. \tag{1.2.3}$$

Let prove that σ_{ij} is a distance metric.

- $\forall\ i, j = 1, 2, ..., n \quad \sigma_{ij} = \|x_i - x_j\| \geq 0$
- if $i = j$ we have $\sigma_{ij} = \|x_i - x_i\| = 0$

[3] Note that this distance is not standard. Depending on the situation, we can use an Euclidean distance or define the distance that reflects the most the situation.

- Let show that $\sigma_{ij} \leq \sigma_{ik} + \sigma_{jk} \; \forall \;\; k = 1, 2, ..., n$.

Theorem 1.2.1 (Cauchy-Schwarz inequality). *[?]*
For any vector u and v of a finite dimensional vectors space, we have $|\langle u, v \rangle| \leq \|u\| \times \|v\|$.

$$\sigma_{ij}^2 = \|x_i - x_k\|^2 + \|x_j - x_k\|^2 - 2(x_i - x_k)(X_j - x_k)$$

$$\leq \|x_i - x_k\|^2 + \|x_j - x_k\|^2 + 2|(x_i - x_k)(x_j - x_k)|$$

$$\leq \|x_i - x_k\|^2 + \|x_j - x_k\|^2 + 2\|x_i - x_k\| \times \|x_j - x_k\| \;\; (Cauchy \; - \; Schwarz\, inequality)$$

$$\leq \left(\|x_i - x_k\| + \|x_j - x_k\|\right)^2$$

$$\Longrightarrow \;\; \sigma_{ij} \leq \|x_i - x_k\| + \|x_j - x_k\| = \sigma_{ik} + \sigma_{jk} \quad So \;\; \sigma_{ij} \;\; is\ a\ distance\ metric.$$

Definition 1.2.2 (Inner Product space). *An inner product space or pre-Hilbert space is a vector space endowed with an inner product on it. furthermore, if the inner product defines a complete metric space then, the inner product space is called Hilbert space.*

Definition 1.2.3 (Hermitian matrix). *An Hermitian matrix is a square matrix whose entries are equal to its entries conjugate transpose.*
$\mathbf{A}$ *is an Hermitian matrix if* $\mathbf{A} = \mathbf{A^H}$ *where* $\mathbf{A^H}$ *denotes the conjugate transpose of* $\mathbf{A}$. *If* a_{ij} *are entries of an Hermitian matrix* $\mathbf{A}$ *then,* $a_{ij} = \bar{a}_{ji}$.

Examples 1.2.4 (Hermitian matrix). *Let* $\mathbf{A} = \begin{bmatrix} -1 & 1-2i & 0 \\ 1+2i & 0 & -i \\ 0 & i & 1 \end{bmatrix}$

The transpose of $\mathbf{A}$ *is* $\mathbf{A}^t = \begin{bmatrix} -1 & 1+2i & 0 \\ 1-2i & 0 & i \\ 0 & -i & 1 \end{bmatrix}$ *and the conjugate of it transpose is* $\overline{\mathbf{A}^t} = \begin{bmatrix} -1 & 1-2i & 0 \\ 1+2i & 0 & -i \\ 0 & +i & 1 \end{bmatrix}$
We can see that $\mathbf{A} = \overline{\mathbf{A}^t}$ *so* $\mathbf{A}$ *is an Hermitian matrix.*

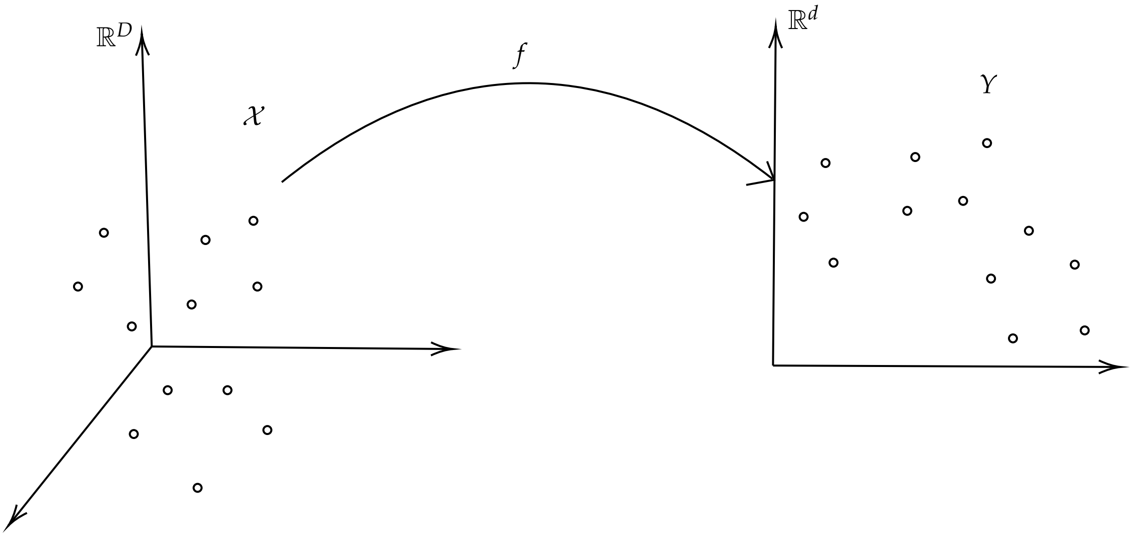
Definition 1.2.5 (Gram matrix). *The Gram matrix* $\mathbf{G}$ *of a set of vector* x_i, x_j *in an inner product space is the Hermitian matrix of inner products whose entries are given by the inner product* $\mathbf{G}_{ij} =< x_i, x_j >$ *of size m.*

Let $\mathbf{X}$ be a data set of dimension n and $x_i, x_j i, j = 1, ..., n$ two features vectors of $\mathbf{X}$. The dissimilarities measure between pairs of points (x_i, x_j) are stored in the multidimensional scaling matrix M of size $m \times m$.

The gram matrix $G = XX^t$ and the multidimensional scaling matrix M are licked by the equation:

$$G = -\frac{1}{2}JMJ \ \ with \ \ J = Id_M - \frac{1}{m}\nVdash^t \ , \ where \ \nVdash \ is \ the \ matrix \ of \ all \ ones. \tag{1.2.4}$$

The goal of multidimensional scaling is to find a set of d-dimensionality Euclidean coordinates $Y = \{y_1, ...y_d\} \in \mathcal{R}^d$ for each data sample such that the dissimilarity measure between pair of points of their Euclidean coordinates match the dissimilarity measure between pair of points of the original Euclidean coordinate as closely as possible.
To finds such Y, we can use different MDS techniques. In this project, we will focus on the classical MDS technique.



The function f is maps $\mathcal{X}$ from $\mathbb{R}^D$ to Y in $\mathbb{R}^d$ so that pairwise distances are close as possible

Using the classical MDS approach, finding such Y can be formulated as an optimization problem

$$\min_{Y \in \mathcal{R}^d} \|G_X - G_Y\|_2^2 \tag{1.2.5}$$

We have

$$\min_{Y \in \mathcal{R}^d} \|G_X - G_Y\|_2^2 = \min_{Y \in \mathcal{R}^d} \|XX^t - YY^t\|_2^2 \tag{1.2.6}$$

$$= \min_{Y \in \mathcal{R}^d} \| -\frac{1}{2}JM_XJ - (-\frac{1}{2}JM_YJ)\|_2^2 \tag{1.2.7}$$

$$= \frac{1}{2}\min_{Y \in \mathcal{R}^d} \|J(M_X - M_Y)J\|_2^2. \tag{1.2.8}$$

The equation 1.2.8 gives the relationship between the optimization problem and the multidimensional scaling matrices.

Furthermore,

$$\min_{Y\in\mathcal{R}^d} \|G_X - G_Y\|_2^2 = \min_{Y\in\mathcal{R}^d} \|XX^t - YY^t\|_2^2 \tag{1.2.9}$$

$$= \min_{Y\in\mathcal{R}^d} \sum_{i=1}^{m}\sum_{j=1}^{m}(x_i^t x_j - y_i^t y_j)^2 \tag{1.2.10}$$

Definition 1.2.6 (Trace of a matrix)**.** *The Trace of a Matrix is sum of its diagonal elements from the upper left to the lower right, of matrix.*

The Trace of the matrix is defined only for a Square Matrix. If **A** and **B** are two matrices then,

$$Tr(\mathbf{AB}) = Tr(\mathbf{BA}). \tag{1.2.11}$$

We have,

$$\left\{Tr\left(\sum_{i=1}^{m}\sum_{j=1}^{m}(x_i^t x_j - y_i^t y_j)^2\right),\ i,j=1,...n\right\} \subset \left\{\sum_{i=1}^{m}\sum_{j=1}^{m}(x_i^t x_j - y_i^t y_j)^2,\ i,j=1,...n\right\} \tag{1.2.12}$$

$$\implies \min_{Y\in\mathcal{R}^d}\left\{Tr\left(\sum_{i=1}^{m}\sum_{j=1}^{m}(x_i^t x_j - y_i^t y_j)^2\right),\ i,j=1,...n\right\} = \min_{Y\in\mathcal{R}^d}\left\{\sum_{i=1}^{m}\sum_{j=1}^{m}(x_i^t x_j - y_i^t y_j)^2,\ i,j=1,...n\right\} \tag{1.2.13}$$

$$\implies \min_{Y\in\mathcal{R}^d} \|G_X - G_Y\|_2^2 = \min_{Y\in\mathcal{R}^d} Tr\left((XX^t - YY^t)^2\right) \tag{1.2.14}$$

So the optimization problem can be reformulated as

$$\min_{Y\in\mathcal{R}^d} Tr\left((XX^t - YY^t)^2\right) \tag{1.2.15}$$

XX^t and YY^t are positive semi-define matrices so they can be decomposed as:

$$XX^t = VDV^t, \quad and \quad YY^t = UD'U^t \tag{1.2.16}$$

Where V is an invertible matrix of dimension $n \times n$, D is a diagonal matrix of dimension $n \times n$ with entries spectrum of XX^t in decreasing other. U is an invertible matrix of dimension $d \times d$

and D' is a diagonal matrix of dimension $n \times n$ with entries spectrum of YY^t in decreasing other.

$$\begin{aligned}
\min_{Y \in \mathcal{R}^d} Tr\left((XX^t - YY^t)^2\right) &= \min_{U,D'} Tr\left((VDV^t - UD'U^t)^2\right) && (1.2.17)\\
&= \min_{U,D'} Tr\left((D - VUD'U^tV^t)^2\right) && (1.2.18)\\
&= \min_{Q',D'} Tr\left((D - QD'Q^t)^2\right) \quad with \;\; Q = VV' && (1.2.19)\\
&= \min_{Q',D'} Tr\left((D^2 - 2DQD'Q^t + QD'Q^tQD'Q^t)\right) && (1.2.20)\\
&= \min_{Q',D'} \left\{Tr\left(D^2\right) - 2Tr\left(DQD'Q^t\right) + Tr\left(QD'Q^tQD'Q^t\right)\right\} && (1.2.21)\\
&= \min_{D'} Tr\left(D^2 - 2DD' + D'^2\right) && (1.2.22)\\
&= \min_{D'} Tr\left((D - D')^2\right) && (1.2.23)\\
&= \min_{D'} Tr\left((D - D')\right) && (1.2.24)
\end{aligned}$$

because D' is minimizing the equation 1.2.23 so it's also minimizing the equation 1.2.24. We get the minimum for

$$\mathcal{D}' = Diag(\lambda_1^{'}, \lambda_2^{'}...\lambda_d^{'}, 0, ...0)$$

Therefore, we can set $\mathcal{Q} = V^tU = Id_N$

Chapter 2

Mathematical Preliminaries of Riemannian Manifolds

In this section we will highlight some important mathematics concepts that are used for dimensionality reduction. The first part will focus on metric and topological spaces, the second part on differential and Riemannian geometry.

2.0 Metric Spaces

The general idea of metric spaces come from the study of continuity. In this section, we will develop the concept of metric spaces, define the notion of continuity, bounded sets, open balls, open sets, closure, limit points, interior, boundary, and convergence in a metric spaces.

Definition 2.0.1 (Metric Spaces). . *Let* **X** *be a non empty set and m a map defined by:*

$$\begin{aligned} m : X \times X &\to \mathbb{R}_+ \\ (x, y) &\mapsto m(x, y). \end{aligned}$$

If m satisfied the following properties:

1. $\forall x, y \in X, m(x, y) \geq 0$ (positiveness);
2. $m(x, y) = 0 \iff x = y \ \forall x, y \in X$ ();
3. $\forall x, y \in X, m(x, y) = m(y, x)$ (symmetry);
4. $\forall x, y, z \in X, m(x, z) \leq m(x, y) + m(y, z)$ (triangle inequality).

Then m is called a metric or a distance function and **X** endowed by a distance function m is called a metric spaces and denoted by $(\mathbf{X}, m)$. The elements of **X** are called points of the space.
If m satisfies only the conditions 2 and 3, then we call m a distance.

Examples 2.0.2. ***Metric Spaces.*** *The Euclidean n-space* $(\mathbb{R}^n,\ m_2)$ *with* m_2 *defined by:*

$$\begin{aligned} m_2 &: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+ \\ (x,y) &\mapsto m_2(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad \textit{is a metric spaces.} \end{aligned}$$

Proof. of example 2.0.2

- $\forall x,y \in X,\ (x_i - y_i)^2 \geq 0$ *with* $x \neq y \implies m_2(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \geq 0$ with $i = 1,2,...,n$. For $x = y$, *we have* $m_2(x,y) = 0$;
- $\forall x,y \in X,\ (x_i - y_i)^2 = (y_i - x_i)^2 \implies m_2(x,y) = m_2(y,x)$ with $i = 1,2,...,n$;
- $\forall x,y,z \in X$, Let show that

$$m_2(x,z) \leq m_2(x,y) + m_2(y,z) \iff \sqrt{\sum_{i=1}^{n}(x_i - z_i)^2} \leq \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} + \sqrt{\sum_{i=1}^{n}(y_i - z_i)^2}$$

Let set $x_i - y_i = r_i$, *and* $y_i - z_i = s_i$ with $i = 1,2,...,n$.
We want to show that

$$\sqrt{\sum_{i=1}^{n}(r_i + s_i)^2} \leq \sqrt{\sum_{i=1}^{n} r_i^2} + \sqrt{\sum_{i=1}^{n} s_i^2} \tag{2.0.1}$$

Since both sides of equation 2.0.1 are positives, we are to square each of them.

$$\left(\sqrt{\sum_{i=1}^{n}(r_i + s_i)^2}\right)^2 = \sum_{i=1}^{n} r_i^2 + \sum_{i=1}^{n} s_i^2 + 2\sum_{i=1}^{n} r_i s_i \tag{2.0.2}$$

$$\left(\sqrt{\sum_{i=1}^{n} r_i^2} + \sqrt{\sum_{i=1}^{n} s_i^2}\right)^2 = \sum_{i=1}^{n} r_i^2 + \sum_{i=1}^{n} s_i^2 + 2\sqrt{\sum_{i=1}^{n} r_i^2 \sum_{i=1}^{n} s_i^2} \tag{2.0.3}$$

By the taking the difference between 2.0.2 and 2.0.3 we obtain:

$$\left(\sqrt{\sum_{i=1}^{n}(r_i + s_i)^2}\right)^2 - \left(\sqrt{\sum_{i=1}^{n} r_i^2} + \sqrt{\sum_{i=1}^{n} s_i^2}\right)^2 = 2\sum_{i=1}^{n} r_i s_i - 2\sqrt{\sum_{i=1}^{n} r_i^2 \sum_{i=1}^{n} s_i^2} \leq 0 \ (\textit{Cauchy Schuarz Inequality}) \tag{2.0.4}$$

$$\begin{aligned} \textit{The equation 2.0.4} &\implies \left(\sqrt{\sum_{i=1}^{n}(r_i + s_i)^2}\right) \leq \left(\sqrt{\sum_{i=1}^{n} r_i^2} + \sqrt{\sum_{i=1}^{n} s_i^2}\right) \\ &\implies \sqrt{\sum_{i=1}^{n}(r_i + s_i)^2} \leq \sqrt{\sum_{i=1}^{n} r_i^2} + \sqrt{\sum_{i=1}^{n} s_i^2} \\ &\implies d_2(x,z) \leq d_2(x,y) + d_2(y,z) \end{aligned}$$

□

2.0.0 Open Balls in a Metric Spaces

The concepts of open balls are very useful to generalize or extend the notion of continuity. The definition of continuity in a metric spaces can be reformulated by using open balls. It can be also used to define the notion of boundary on a subspace of a metric space.

Definition 2.0.3. *Open ball*

Let $(\mathbf{X}, d)$ *be a metric space and let* $x \in \mathbf{X}$ *and* $r > 0$ *un real number.*
The subset $B_d(x, r) := \{y \in \mathbf{X} : d(x, y) < r\}$ *is called the open ball centered at* x *with the radius* r *with respect to the metric* d *in* $\mathbf{X}$.

Examples 2.0.4. *Any open interval* (a, b) *in* $\mathbb{R}$ *is an open ball.*

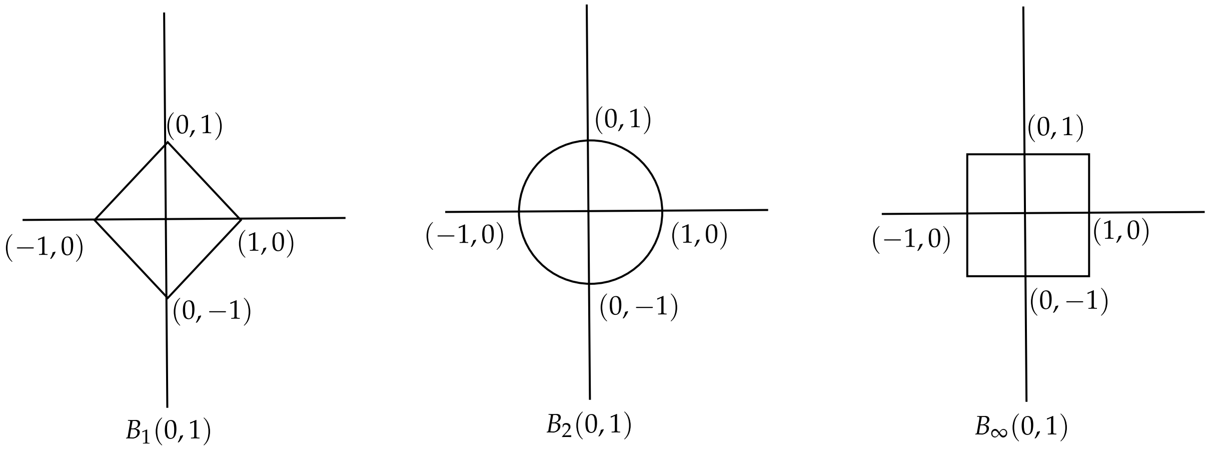
Proof. of example 2.0.4

Let $(a, b) = B(\frac{a+b}{2}, \frac{b-a}{2})$.

$$\begin{aligned} x \in B(\frac{a+b}{2}, \frac{b-a}{2}) &\implies \left| x - \frac{a+b}{2} \right| < \frac{b-a}{2} \\ &\implies \frac{a-b}{2} < x - \frac{a+b}{2} | < \frac{b-a}{2} \\ &\implies a < x < b \\ &\implies x \in (a, b) \end{aligned}$$

□

Examples 2.0.5. *Open balls with respect to the metrics* d_1, d_2, d_∞. .



$B_1(0,1)$ $B_2(0,1)$ $B_\infty(0,1)$

Figure 2.1: Open Balls respectively associated to metrics $d_1, d_2,$ *and* d_∞

Proposition 2.0.5.1. *Let* $(\mathbf{X}, d)$ *be a metric space and let* $x \in \mathbf{X}$ *such that* $B(x, r)$ *is an open ball in* $\mathbf{X}$. *For any* $y \in B(x, r)$, *there exists* $\sigma > 0$ *such that* $B(y, \sigma) \subset B(x, r)$.

Proof. of proposition 2.0.5.1 $y \in B(x,r) \implies d(x,y) < r$. Let set $\sigma = r - d(x,y)$.
Now we want to show that $B(y,\sigma) \subset B(x,r)$.
$z \in B(y,\sigma) \implies d(y,z) < \sigma$.
Furthermore, d is a distance so by the triangle inequality we have $d(x,z) \leq d(x,y) + d(y,z) < d(x,y) + \sigma = r$.
We then have $d(x,z) < r \implies z \in B(x,r)$ *so* $B(y,\sigma) \subset B(x,r)$. □

2.0.1 Open Sets in a Metric Spaces

The notion of Open sets in a metric spaces gives a generalization of the concepts of open balls in a metric spaces.

Definition 2.0.6. *Open set*

Let $(\mathbf{X}, d)$ *be a metric spaces and let* $\mathcal{O} \subseteq \mathbf{X}$ *be a subset of* $\mathbf{X}$.
$\mathcal{O}$ *is open in* $\mathbf{X}$ *if for every element* $x \in \mathcal{O}$ *we can find* $\epsilon_x > 0$ *such that* $B(x,\epsilon_x) \subseteq \mathcal{O}$.

Proposition 2.0.6.1. *Let* $(\mathbf{X}, d)$ *be a metric spaces and let* $\{\mathcal{O}_i\}_{i\in I}$ *be a family of open subsets* $\mathcal{O}_i \subseteq X$ *of* $\mathbf{X}$. *Then,* $\bigcup\limits_{i\in I} \mathcal{O}_i$ *is open in* $\mathbf{X}$.
In other words, the union of any collection of open subsets of a metric spaces is an open subset of the same metric spaces.

Proof. of Proposition 2.0.6.1

Let $x \in \bigcup\limits_{i\in I} \mathcal{O}_i \implies \exists i_0 \in I$ *such that* $x \in \mathcal{O}_{i_0}$.
We have $\mathcal{O}_{i_0}$ is open in $\mathbf{X}$ so, there exists $\epsilon > 0$ such that $B(x,\epsilon) \subseteq \mathcal{O}_{i_0} \implies B(x,\epsilon) \subseteq \bigcup\limits_{i\in I} \mathcal{O}_i$. We can conclude that $\bigcup\limits_{i\in I} \mathcal{O}_i$ is open in $\mathbf{X}$. □

Examples 2.0.7. *Open set*

tttrr

2.0.2 Convergence in a Metric Spaces

Definition 2.0.8 (Convergence). *Let* $(\mathbf{X}, d)$ *be a metric space, a sequence* $\{x_n\} \in \mathbf{X}$ *converges to* x^* *if*

$$\forall \epsilon > 0, \exists n_\epsilon \in \mathbb{N}, n > n_\epsilon \implies d(x_n, x^*) < \epsilon. \tag{2.0.5}$$

The Equation 2.0.5 is equivalent to :

$$\forall \epsilon > 0, \exists n_\epsilon \in \mathbb{N}, n > n_\epsilon \implies x_n \in B(x^*, \epsilon).$$

We say that $\{x_n\}$ converges to x^* and we note $\{x_n\} \to x^*$.

Proposition 2.0.8.1. *Let* $\{x_n\}$ *and* $\{y_n\}$ *two sequences in a normed vector space* $(\mathcal{V}, ||)$ *converging respectively to* x^* *and* y^*. *Then* $\{x_n\} + \{y_n\}$ *converge to* $x^* + y^*$.

Proof. of Proposition 2.0.8.1

$$\{x_n\} \to x^* \implies \forall \epsilon > 0, \exists n_{x^*} \in \mathbb{N}, n > n_{x^*} \implies \|x_n - x^*\| < \epsilon/2 \tag{2.0.6}$$

$$\{y_n\} \to y^* \implies \forall \epsilon > 0, \exists n_{y^*} \in \mathbb{N}, n > n_{y^*} \implies \|y_n - y^*\| < \epsilon/2 \tag{2.0.7}$$

$$\implies \|x_n - x^*\| + \|y_n - y^*\| < \epsilon. \tag{2.0.8}$$

Moreover,

$$\|(x_n + y_n) - (x^* + y^*)\| = \|(x_n - x^*) + (y_n - y^*)\| \leq \|x_n - x^*\| + \|y_n - y^*\| < \epsilon \tag{2.0.9}$$

$$\implies \|(x_n + y_n) - (x^* + y^*)\| < \epsilon. \tag{2.0.10}$$

Take $n_{x^*+y^*} = \max(n_{x^*}, n_{y^*})$. □

2.0.3 Continuity in a Metric Spaces

Definition 2.0.9 (Continuity).

Proposition 2.0.9.1. *Let $(\mathbf{X}, d_{\mathbf{X}})$ and $(\mathbf{Y}, d_Y)$ be two metric spaces. Let f be a function*

$$f : (\mathbf{X}, d_{\mathbf{X}}) \to (\mathbf{Y}, d_{\mathbf{Y}}).$$

we say that f is continuous in $x_0 \in \mathbf{X}$ iff $\forall\ \epsilon > 0$, we can find a real number $\sigma > 0$ such that $f(B_{\mathbf{X}}(x_0, \sigma)) \subseteq B_{\mathbf{Y}}(f(x_0), \epsilon)$.

Proof. of Proposition 2.0.9.1

- Suppose f is continuous in $a\mathbf{X}$, and let $a \in \mathbf{X}$ we want to show that $\forall \epsilon > 0, \exists \sigma > 0$ *such that* $f(B_{\mathbf{X}}(a, \sigma)) \subseteq B_{\mathbf{Y}}(f(a), \epsilon)$.
  f is continuous in $\mathbf{X}$ which implies that for an open subset $\mathbf{U} \subset \mathbf{Y}$, $f^{-1}(\mathbf{U})$ is an open subset of $\mathbf{X}$.
  Let $\epsilon > 0$, we have $B_{\mathbf{Y}}(f(a), \epsilon) = \{a \in \mathbf{X} \mid d_{\mathbf{Y}}(f(a), f(x)) < \epsilon\}$.
  $B_{\mathbf{Y}}(f(a), \epsilon)$ is an open ball in $\mathbf{Y}$ so, $B_{\mathbf{Y}}(f(a), \epsilon)$ is an open subset of $\mathbf{Y}$ which implies that $f^{-1}(B_{\mathbf{Y}}(f(a), \epsilon))$ is an open subset of $\mathbf{X}$ because f is continuous in $\mathbf{X}$.
  From the Proposition 2.0.5.1, there exists $\sigma > 0$ *such that* $B_{\mathbf{X}}(a, \sigma) \subseteq f^{-1}(B_{\mathbf{Y}}(f(a), \epsilon))$

  $$B_{\mathbf{X}}(a, \sigma) \subseteq f^{-1}(B_{\mathbf{Y}}(f(a), \epsilon)) \implies f(B_{\mathbf{X}}(a, \sigma)) \subseteq f\left(f^{-1}(B_{\mathbf{Y}}(f(a), \epsilon))\right) \subseteq B_{\mathbf{Y}}(f(a), \epsilon)$$
  $$\implies f(B_{\mathbf{X}}(a, \sigma)) \subseteq B_{\mathbf{Y}}(f(a), \epsilon).$$

- Let $a \in \mathbf{X}$, and suppose that $\forall \epsilon > 0, \exists\ \sigma > 0$ *such that*, $f(B_{\mathbf{X}}(a, \sigma)) \subseteq B_{\mathbf{Y}}(f(a), \epsilon)$. We want to show that f is continuous in $\mathbf{X}$.
  Let $\mathbf{U}$ be an open subset of $\mathbf{Y}$, we want to show that $f^{-1}(\mathbf{U})$ is an open subset of $\mathbf{X}$.

Let $a \in f^{-1}(\mathbf{U})$,

$$\begin{aligned} a \in f^{-1}(\mathbf{U}) &\implies f(a) \in \mathbf{U} \\ &\implies B_{\mathbf{Y}}(f(a), \epsilon) \subseteq \mathbf{U} \end{aligned}$$

From the hypothesis there exists $\sigma > 0$ *such that* $f(B_{\mathbf{X}}(a, \sigma)) \subseteq B_{\mathbf{Y}}(f(a), \epsilon)$

$$\begin{aligned} &\implies f(B_{\mathbf{X}}(a,\sigma)) \subseteq \mathbf{U} \\ \implies\ & B_{\mathbf{X}}(a,\sigma) \subseteq f^{-1}(\mathbf{U}). \end{aligned}$$

We have

$$\begin{aligned} f^{-1}(\mathbf{U}) &= \bigcup_{x \in f^{-1}(\mathbf{U})} \{x\} \subset \bigcup_{x \in f^{-1}(\mathbf{U})} B_{\mathbf{X}}(a,\sigma) \subseteq f^{-1}(\mathbf{U}) \\ \implies f^{-1}(\mathbf{U}) &= \bigcup_{x \in f^{-1}(\mathbf{U})} B_{\mathbf{X}}(a,\sigma) \end{aligned}$$

We can conclude that $f^{-1}(\mathbf{U})$ is open in $\mathbf{X}$ so f is continuous.

□

Corollary 2.0.1. *Let* $(\mathbf{X}, m_x)$ *and* $(\mathbf{Y}, m_y)$ *be two metric spaces. A function* $g : \mathbf{X} \to \mathbf{Y}$ *is said to be continuous at* $x \in \mathbf{X}$ *if and only if for every sequence* $(x_n)_{n \in \mathbb{N}} \in \mathbf{X}$ *converging to* x*, we have* $g(x_n)$ *is also converging to* $g(x)$*. Furthermore, if* g *is continuous at each point* $x \in \mathbf{X}$*, we say that* g *is continuous on* $\mathbf{X}$*.*

2.0.4 Bounded Sets in a Metric Spaces

Let $(\mathbf{X}, d)$ be a metric space and let $\mathbf{B}$ be a subset of $\mathbf{X}$. We say that $\mathbf{B}$ is bounded in $\mathbf{X}$ if for $x \in \mathbf{X}$, we can find a real number $\sigma > 0$ *such that* $\mathbf{B} \subset B(x, \sigma)$.

2.0.5 Boundary in a Metric Spaces

The notion of boundary helps define the concept of converge in a metric space.
Let $(\mathbf{X}, d)$ be a metric space and $\mathbf{A}$ a subset of $\mathbf{X}$.

- The **diameter** of $\mathbf{A}$ and denoted by $diam(\mathbf{A})$ is defined by:

$$diam(\mathbf{A}) = \sup\{d(a,b), a, b \in \mathbf{A}\};$$

- Let $x \in \mathbf{X}$. The **distance** from x to $\mathbf{A}$ and denoted by $d(x, \mathbf{A})$ is defined by:

$$d(x, \mathbf{A}) = \inf\{d(x,a), a \in \mathbf{A}\}.$$

Definition 2.0.10 (Boundary). *Let* $(\mathbf{X}, d)$ *be a metric space,* $\mathbf{A}$ *be any non-empty subset of* $\mathbf{X}$ *and* a *be any point in* $\mathbf{X}$*.*
If we have $d(a, \mathbf{A}) = 0 = d(a, \mathbf{A}^c)$ *then,* a *is called boundary point of* $\mathbf{A}$ *in* $\mathbf{X}$ *where* $\mathbf{A}^c$ *is the complement of* $\mathbf{A}$*.*

In other words, a is called boundary point of $\mathbf{A}$ in $\mathbf{X}$ if any small ball centered at a has non-empty intersections with both $\mathbf{A}$ and its complement $\mathbf{A}^c$.
Let $\{a_i\}_{i\in I} \subseteq \mathbf{X}$ be a family of all boundary points of $\mathbf{A}$ in $\mathbf{X}$. Then, $\{a_i\}_{i\in I}$ is called boundary of $\mathbf{A}$ in $\mathbf{X}$ and denoted by $\sigma_{\mathbf{X}}\mathbf{A}$.

Theorem 2.0.2. *Let $(\mathbf{X}, d)$ be a metric space, $\mathbf{A}$ be a subset of $\mathbf{X}$. The boundary of $\mathbf{A}$ in $\mathbf{X}$ is equal to the boundary of $\mathbf{A}^c$ in $\mathbf{X}$.*

$$\sigma_{\mathbf{X}}\mathbf{A} = \sigma_{\mathbf{X}}\mathbf{A}^c$$

Proof. of Theorem 2.0.2

Let $x \in \sigma_{\mathbf{X}}\mathbf{A}$, from the definition of a boundary point, we have

$$d(x, \mathbf{A}) = 0 = d(x, \mathbf{A}^c)$$

$$\begin{aligned} d(x, \mathbf{A}) = 0 = d(x, (\mathbf{A}^c)^c) = 0 = d(x, \mathbf{A}^c) \\ &\iff d(x, \mathbf{A}^c) = 0 = d(x, (\mathbf{A}^c)^c) \\ &\iff x \in \sigma_{\mathbf{X}}\mathbf{A}^c \\ &\iff \sigma_{\mathbf{X}}\mathbf{A} = \sigma_{\mathbf{X}}\mathbf{A}^c \end{aligned}$$

□

Examples 2.0.11 (Boundary point). *a is a boundary point of (a, b) for any $a, b \in \mathbb{R}$*

Proof. of Example 2.0.11

$$\begin{aligned} \text{Let } 0 < \epsilon < b - a, \; &\Longrightarrow a + \epsilon \in (a, b) \\ &\Longrightarrow d(a, (a, b)) \leq d(a, \epsilon + a) = \epsilon \\ &\Longrightarrow d(a, (a, b)) \leq \epsilon \\ &\Longrightarrow 0 < d(a, (a, b)) \leq \epsilon. \end{aligned}$$

Since ϵ is arbitrary positive number, we can say that $d(a, (a, b)) = 0$.
Now we want to show that $d(a, (a, b)^c) = 0$.
We have $(a, b)^c = (\infty, a] \cup [b, \infty)$, and $a \in (\infty, a] \cup [b, \infty) \implies d(a, (a, b)^c) = 0$ and we have the result.
Then $d(a, (a, b)) = 0 = d(a, (a, b)^c)$ so a is a boundary point of (a, b). □

The boundary of (a, b) is $\{a, b\}$

2.0.6 Normed Spaces

Normed spaces can be seen as length of vectors in a vector space. The notion of norm gives rise to the conceopt of inner product and the converse is still true.

Definition 2.0.12 (Normed Spaces). *Let $\mathcal{V}$ be a $\mathbb{K}$ with $\mathbb{K} = \mathbb{R}$ ou $\mathbb{C}$ linear space. $\mathcal{V}$ with a function $v \mapsto \|v\|$ which associates with every vector $v \in \mathcal{V}$ a real number $\|v\|$, called the norm of v, is said to be a normed space if for all vectors $v, w \in \mathcal{V}$ and $\lambda \in \mathbb{K} = (\mathbb{R}$ or $\mathbb{C})$ the following conditions hold.*

1. $\|v\| > 0 \; if \; x \notin O_{\mathcal{V}}$ *(positivity);*

2. $\|\lambda v\| = |\lambda| \|v\|$ *(Homogeneity);*

3. $\|v + w\| \leq \|v\| + \|w\|$ *(Triangle inequality).*

Note that every normed space ca be endowed with a structure of a metric space defined by:

$$\begin{aligned} d : \mathcal{V} \times \mathcal{V} &\to \mathbb{R}_+ \\ (x, y) &\mapsto d(x, y) = \|x - y\| \end{aligned}$$

Let's now define the concepts of **semi-norm** and **norm** in a vector space.

Definition 2.0.13 (Semi-norm). *Let $\mathcal{V}$ be a linear space over a set of real number $\mathbb{R}$ or complex numbers $\mathbb{C}$.*
A semi-norm in $\mathcal{V}$ is a function P mapping every vector $v \in \mathcal{V}$ to a non-negative real number $p(v)$ such that the following conditions hold

1. $p(\alpha v) = |\alpha| p(v), \quad \alpha \in \mathbb{K} = (\mathbb{R} \text{ or } \mathbb{C}), \;\; v \in \mathcal{V};$

2. $p(v + w) \leq p(v) + p(w), \quad v, w \in \mathcal{V};$

3. $p(v) = 0 \implies v = O_{\mathcal{V}} \quad v \in \mathcal{V}.$

When p satisfy the above conditions, we say that p is a norm of p is a norm of v, we note $p(v)$, and we write $\|v\| = p(v)$.

Definition 2.0.14 (Bounded Operator). *Let $\mathcal{V}$ and $\mathcal{W}$ be two normed spaces. An operator $B : \mathcal{V} \to \mathcal{W}$ between $\mathcal{V}$ and $\mathcal{W}$ is called a bounded a bounded operator if we ca find a positive real number M such that*

$$\|Bx\| \leq M\|x\| \forall x \in \mathcal{V}$$

Definition 2.0.15 (Operator Norm). *Let $B : \mathcal{V} \to \mathcal{W}$ be a linear operator. The operator norm of B is defined by*

$$\|B\| = \inf\{M \geq 0 \; \textit{such that} \; \forall x \in \mathcal{V} \; \|Bx\| \leq M\|x\|\} \; \textit{with} \; \inf \emptyset = \infty$$

Furthermore, if $\|B\|$ is finite we say that B is bounded.

2.0.7 Complete Metric Spaces

Definition 2.0.16 (Complete Metric Spaces)**.** *Let* $(\mathbf{X}, d)$ *be a metric space,* $\mathbf{X}$ *is* ***complete*** *if every Cauchy sequence in* $\mathbf{X}$ *converges in* $\mathbf{X}$*.*

Examples 2.0.17.

- $(\mathbb{R}, \|\|)$ *is a complete metric space*

- $(0, 1) \subseteq \mathbb{R}$ *is not a complete metric space because we have* $\{\frac{1}{n}\}_{n \in \mathbb{N}^*}$ *is a Cauchy sequence which doesn't converge to any point in* $(0, 1)$*.*

Let $\mathcal{V}$ be a normed linear space.$\mathcal{V}$ is a **Banach** space if $\mathcal{V}$ is complete. Furthermore, if the norm endowed by $\mathcal{V}$ is an inner product we then say that $\mathcal{V}$ is a **Hilbert** space.

Definition 2.0.18 (Isometry)**.** *Let* $(\mathbf{X}, \phi)$ *and* $(\mathbf{Y}, \mu)$ *be two metric spaces, and* $f : (\mathbf{X}, \phi) \to (\mathbf{Y}, \mu)$ *a map. We say that* f *is an isometry if* f *preserves the metric. Which means:*

$$\mu(f(x), f(y)) = \phi(x, y) \quad \forall x, y \in \mathbf{X}.$$

Examples 2.0.19 (Isometry)**.**

$$\begin{aligned} f &: \mathbb{C} \to \mathbb{R}^2 \\ z = (x + iy) &\mapsto (x, y) \textit{is an isometry.} \end{aligned}$$

2.0.8 Embedding

The term of embedding is useful in representing a space into another one while preserving distances.
Let $(\mathbf{X}, \phi)$ and $(\mathbf{Y}, \mu)$ be two metric spaces. A map f defined from $\mathbf{X}$ to $\mathbf{Y}$ is called embedding.
Embedding method is one of the most powerful method when it comes to design approximation algorithms. Basically, embedding try to reformulate a problem defined over a difficult metric into one more easier to understand.
Let a map f defined by: $f : (\mathbf{X}, \phi) \to (\mathbf{Y}, \mu)$ be an embedding. Let's define the notions of **Expansions** and **Contractions**.
The Expansions of f and denoted $Expansions(f)$ is defined by:

$$Expansions(f) = \max_{x,y \in \mathbf{X}} \frac{\mu(f(x), f(y))}{\phi(x, y)}$$

The Contractions of f and denoted $Contractions(f)$ is defined by:

$$Contractions(f) = \max_{x,y \in \mathbf{X}} \frac{\phi(x, y)}{\mu(f(x), f(y))} \leq 1.$$

2.1 Topological Spaces

We need to study the notion of topological space to give a generalization of the concepts of metric spaces.

Definition 2.1.1 (Topological Space). *Let* $\mathbf{X}$ *be an non empty-set and* $\mathcal{T}$ *a fixed family of subsets of* $\mathbf{X}$. *We said that* $(\mathbf{X}, \mathcal{T})$ *is a topological space if it satisfies the following properties:*

1. $\mathbf{X}, \emptyset \in \mathcal{T}$;
2. *The intersection of any two sets in* $\mathcal{T}$ *is in* $\mathcal{T}$;
3. *The union of any collection of sets in* $\mathcal{T}$ *is in* $\mathcal{T}$.

The elements of $\mathcal{T}$ are called open sets.
The smallest topology containing all the open balls $B(x,r) = \{y \in \mathbf{X} \mid m(x,y) < r\}$ is called the metric topology on $\mathbf{X}$ generated by m.

Examples 2.1.2 (Topological Spaces).

- *Euclidean spaces, where the topology is the unions of open intervals;*
- *Metric spaces, where the topology is the unions of open balls;*
- *Manifolds, where the topology is the unions of open intervals because manifold is very closed to Euclidean space in a sense where each point of an n-dimensional manifold has a neighborhood that is homeomorphic to the Euclidean space of dimension n.*

Definition 2.1.3 (Continuity in a Topological Spaces). *Let* $(\mathbf{X}, \mathcal{T}_\mathbf{X})$ *and* $(\mathbf{Y}, \mathcal{T}_\mathbf{Y})$ *be two topological spaces and* $f : \mathbf{X} \to \mathbf{Y}$ *a map such defined.* f *is continuous if for* $\mathbf{U} \in \mathcal{T}_\mathbf{Y}$ *we have* $f^{-1}(\mathbf{U}) \in \mathcal{T}_\mathbf{X}$.

After defining the notion of topological space, we are interested of knowing the concept of topological equivalence between two topological spaces and for that, we need to introduce the concept of homeomorphism.

Definition 2.1.4 (Homeomorphism). *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be two topological spaces and* $f : \mathbf{X} \to \mathbf{Y}$ *a map.* f *is an homeomorphism if* f *is bijective, continuous, and its inverse is also continuous.*

Examples 2.1.5 (Homeomorphism).

2.2 Differential Geometry

Differential Geometry can be defined as a branch of mathematics that use tools from linear algebra, differential calculus, integral calculus, and multi-linear algebra to study the geometry of curves, manifolds and surfaces.
Differential Geometry is very helpful in computer vision to deal with images segmentation problems. It has been used to describe the structure of a data points and to approximate the nearest neighbors of a given point on a manifold by computing the minimum distances from that point to its neighbors.

In this section, we will describe some basic notions in differential geometry as smooth manifold, chart, atlas, geodesic, Riemannian metric, geodesic curve, tangent space, exponential map, logarithm map, and connected points.

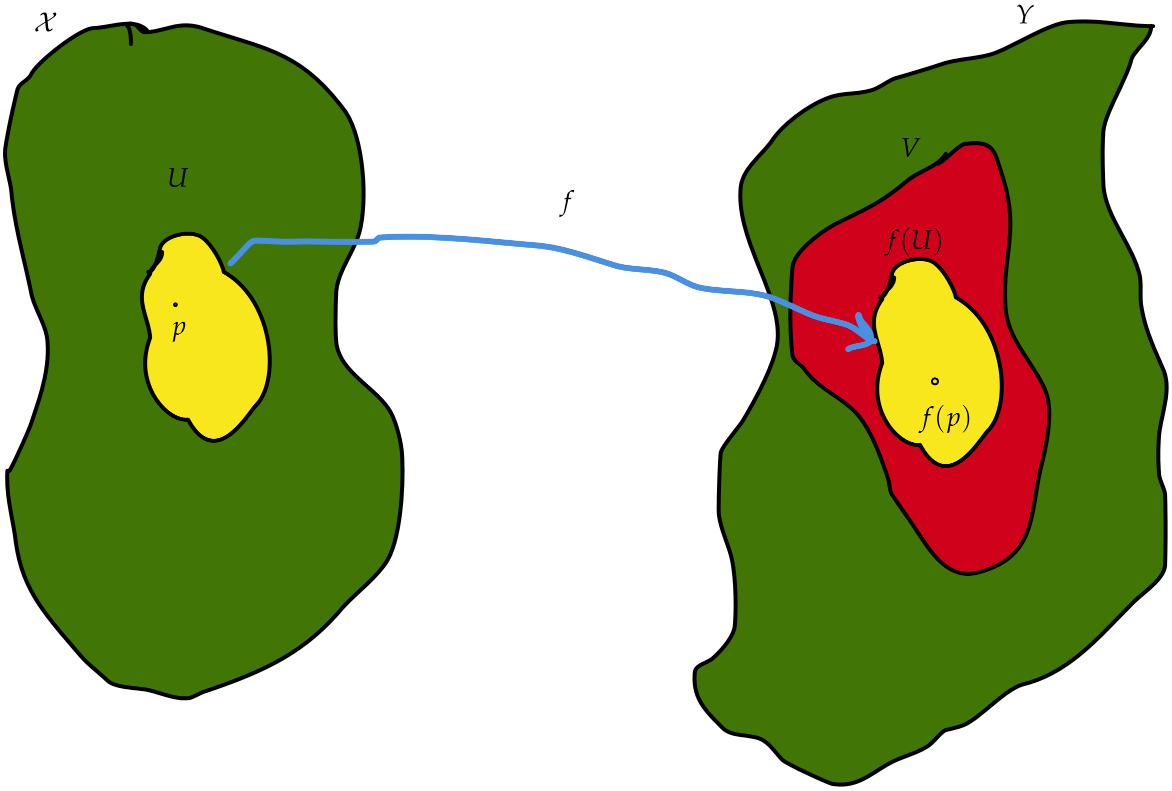


Figure 2.2: Example of Homeomorphism f

2.3 Manifolds

In this section, we will introduice manifolds, topological manifolds which is the most simple type of manifolds. We will explain also the notion of smooth manifolds, smooth topological space, local chart and atlas.

Definition 2.3.1 ($\mathcal{M}$anifold)**.** *A $\mathcal{M}$anifold $\mathcal{M}$ of dimension n is a topological space where every point has a neighborhood topologically equivalent to the unit ball of $\mathbb{R}^n$.*

Examples 2.3.2 ($\mathcal{M}$anifolds)**.**

2.3.1 Topological $\mathcal{M}$anifolds

Let $(\mathcal{X}, \mathcal{T})$ be a topological space, $\mathcal{X}$ is a topological manifold of dimension n if the following properties are satisfied

1. $\mathcal{X}$ is a Hausdorff space: means tat every paire of distinct point a, $b \in \mathcal{X}$ admit disjoint open subsets $\mathcal{U}$, $\mathcal{V} \subseteq \mathcal{X}$ such that $a \in \mathcal{U}$ and $b \in \mathcal{V}$;

2. $\mathcal{X}$ is second-countable: means there exist a countable basis for the topology of $\mathcal{X}$;

3. $\mathcal{X}$ is locally Euclidean of dimension n: means that every point of $\mathcal{X}$ has a neighborhood that is homeomorphic to an open subset of $\mathbb{R}^n$. In other words, for every point $a \in \mathcal{X}$, there exists
   - an open subset $\mathcal{U} \subseteq \mathcal{X}$ containing a;
   - an open subset $\mathcal{V} \subseteq \mathbb{R}^n$;
   - a homeomorphism $\phi : \mathcal{U} \to \mathcal{V}$.

Examples 2.3.3 (Topological $\mathcal{M}$anifolds). *This example shows a topological manifold with its charts.*

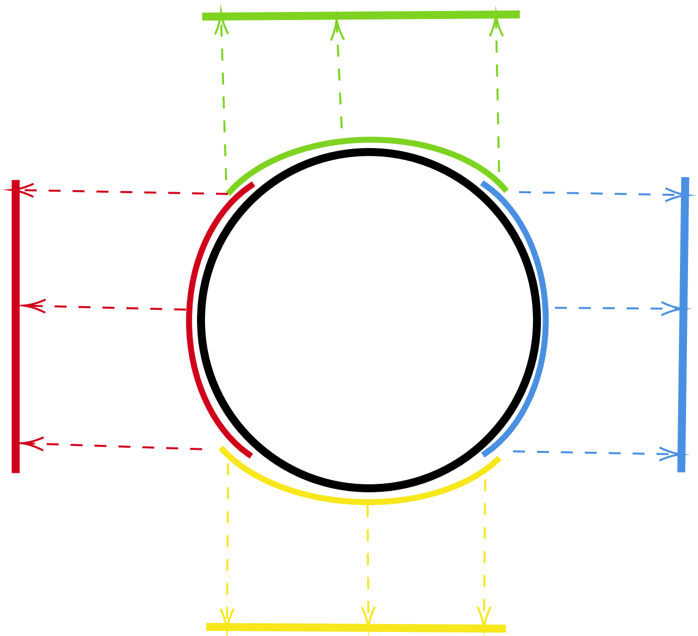


Figure 2.3: Topological $\mathcal{M}$anifolds

The figure 2.3 is a circle which is a typical example of a topological manifold.

2.3.1.1 Smooth $\mathcal{M}$anifolds and smooth topological space

A smooth manifold can be seen as a set with two structures: a topology structure and a smooth structure.

Definition 2.3.4 (Smooth Manifolds). *A smooth manifold of dimension n is a topological space homeomorphic to the Euclidean space $\mathbb{R}^n$.*

More explicitly, a smooth manifold $\mathcal{M}$ of dimension n is a collection of points forming a topological space such that each point $p \in \mathcal{M}$ admits a neighborhood $\mathcal{U} \subseteq \mathcal{M}$ such that $\mathcal{U}$ is homeomorphic to all open subset of $n - dimensional$ Euclidean space. In other words, $\mathcal{M}$ is a smooth manifold if for all $p \in \mathcal{M}$ there exists $\mathcal{U} \subseteq \mathcal{M}$ subset of $\mathcal{M}$ and neighborhood of p such

that for all open subset $\mathcal{V} \subseteq \mathbb{R}^n, \mathcal{U}$ is homeomorphic to $\mathcal{V}$.
Let ϕ be that homeomorphism, we then have $\phi : \mathcal{U} \subset \mathcal{M} \to \phi(\mathcal{U}) = \mathcal{V} \subset \mathbb{R}^n$ is called coordinate map of p and its reverse $\Phi = \phi^{-1} : \mathcal{V} \subset \mathbb{R}^n \to \mathcal{U} \subset \mathcal{M}$ is called local parametrization of P on $\mathcal{M}$.

Examples 2.3.5 (smooth manifold).

$$S^1 = \{(x,y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1\}$$
$$Torus = \{(x,y,z) \in \mathbb{R}^3 \mid (c - \sqrt{x^2+y^2})^2 + z^2 = a^2, with\ a > 0,\ c \in \mathbb{R}\ \}$$

Definition 2.3.6 (local parametrization on a manifold). *Let $\mathcal{M}$ be a smooth manifold of dimension n, p an element of $\mathcal{M}, \mathcal{U}$ a neighborhood of p, and $\mathcal{V}$ an open subset of $\mathbb{R}^n$.*
Let $\phi : \mathcal{U} \subset \mathcal{M} \to \mathcal{V} \subset \mathbb{R}^n$ be the homeomorphis between $\mathcal{U}$ and $\mathcal{V}$ which at any point $p \in \mathcal{M}$ associate $\phi(p) = (x^1(P), x^2(P), ..., x^n(P))$ where $x^i(p)$ are the i^{th} coordinate of $\phi(p)$.
we can define n functions such that

$$\begin{aligned} x^1 : \mathcal{U} &\to \mathbb{R} \\ p &\mapsto x^1(p) \\ &... \\ &... \\ &... \\ x^n : \mathcal{U} &\to \mathbb{R} \\ p &\mapsto x^n(p) \end{aligned}$$

and those functions are called local coordinate and the local parametrization of p on $\mathcal{M}$ is defined by:

$$p = (\phi^{-1} \circ \phi)(p) = \phi^{-1}(\phi(p)) = \phi^{-1}(x^1, x^2, ..., x^n) = (u^1, u^2, ..., u^n).$$

Note that P can also be labeled on a manifold $\mathcal{M}$ using its local parametrization.

Definition 2.3.7 (smooth topological space). *A topological space $\mathcal{M}$ is smooth if for any couple $(\mathcal{U}_\alpha, \phi_\alpha)_{\alpha \in \mathbf{I}}$ where $\mathcal{U}_\alpha$ is an open subset of $\mathcal{M}$ $(\mathbf{U}_\alpha \subset \mathcal{M})$ and ϕ_α: $\mathcal{U}_\alpha \longrightarrow (\phi_\alpha)(\mathcal{U})$ is an homeomorphism we have:*

- $\mathcal{M} = \bigcup_{\alpha \in \mathbf{I}} \mathcal{U}_\alpha$ *which means $\mathcal{U}$ covers $\mathcal{M}$.*
- $\mathcal{U}_\alpha \cap \mathcal{U}_\beta \neq \emptyset \implies (\phi_\beta) \circ (\phi_\alpha)^{-1}$ *is smooth.*

Definition 2.3.8 (local charts). *Let $\mathcal{M}$ be a $n-$dimensional smooth manifold, $\mathcal{U} \in \mathcal{M}$ a subset of $\mathcal{M}$, and a map $\phi : \mathcal{U} \to \mathbb{R}^n$ such that $\phi(\mathcal{U}) \subseteq \mathbb{R}^n$ is open, and $\phi : \mathcal{U} \to \phi(\mathcal{U})$ is a bijection. Then, the $n-$dimensional coordinate $(\mathcal{U}, \phi)$ is called a local chart.*

Let $p \in \mathcal{M}$ and $(\mathcal{U}, \phi)$, $(\mathcal{V}, \psi)$ two charts such that $p \in \mathcal{U} \cap \mathcal{V}$. There exists a map h such that

$$h : \phi(\mathcal{U} \cap \mathcal{V}) \to \psi(\mathcal{U} \cap \mathcal{V})\ is\ an\ homeomorphism.$$

We note $h = \phi o \psi^{-1}$ restricts to $\phi(\mathcal{U} \cap \mathcal{V})$ and called a transition of charts. where $\phi(\mathcal{U} \cap \mathcal{V})$ and $\psi(\mathcal{U} \cap \mathcal{V})$ are open subset in $\mathbb{R}^n$.

Definition 2.3.9 (compatibility between charts)**.** *Let $\mathcal{M}$ be a $n-$dimensional smooth manifold, $(\mathcal{U}, \phi)$ and $(\mathcal{V}, \psi)$ two charts on $\mathcal{M}$, $\phi : \mathcal{U} \to \phi(\mathcal{U})$, and $\psi : \mathcal{V} \to \psi(\mathcal{V})$ two homeomorphisms.*
We say that $(\mathcal{U}, \phi)$ and $(\mathcal{V}, \psi)$ are compatible if $\phi(\mathcal{U} \cap \mathcal{V})$ and $\psi(\mathcal{U} \cap \mathcal{V})$ are open and the map

$$\psi \rho \phi^{-1} : \phi(\mathcal{U} \cap \mathcal{V}) \to \psi(\mathcal{U} \cap \mathcal{V}) \; is \; a \; diffeomorphism^{1}.$$

Definition 2.3.10 (Atlas)**.** *Let $\mathcal{M}$ be a $n-$dimensional smooth manifold, and $(\mathcal{U}_i, \phi_i)$ charts on $\mathcal{M}$. The family $\{(\mathcal{U}_i, \phi_i)\}_{i \in I}$ is called atlas on $\mathcal{M}$ if*

1. *$\mathcal{M} = \cup_{i \in I} \mathcal{U}_i$: means $\mathcal{U}_i$ covers $\mathcal{M}$;*
2. *$\psi \rho \phi^{-1}$ is smooth with $\mathcal{U} \cap \mathcal{V} \neq \emptyset$.*

Examples 2.3.11 (Atlas)**.**

$$Let's \;\; S^2 = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 = 1$$

Let's $p \in S^2$, $N = (0, 0, 1)$ and $S = (0, 0, -1)$ be respectively the North pole and the south pole of S^2; Let's $U_N = S^2\{N\}$ and $U_S = S^2\{S\}$ be two open subsets of S^2 and ϕ_N and ϕ_S two homeomorphisms defined as:

$$\begin{aligned} \phi_N : U_N &\to \mathbb{R}^2 \\ p &\mapsto \phi_N(p) \end{aligned}$$

$$\begin{aligned} \phi_S : U_S &\to \mathbb{R}^2 \\ p &\mapsto \phi_S(p) \end{aligned}$$

Let's consider the triangle NQp with $Q = (0, 0, z)$, $p = (x, y, z)$, $O = (0, 0, 0)$, $\phi_N(p) = (X_N, Y_N)$;

$$\frac{pQ}{\phi_N(p)O} = \frac{NQ}{NO} = 1 - z \tag{2.3.1}$$

$$\phi_N(p)O = Y_N \;\; and \;\; pQ = y \implies \frac{pQ}{\phi_N(p)O} = \frac{y}{Y_N} \tag{2.3.2}$$

$$2.3.1 \; and \; 2.3.2 \implies 1 - z = \frac{y}{Y_N} \implies Y_N = \frac{y}{1 - z} \tag{2.3.3}$$

[1] A diffeomorphism is a bijective map, differentiable and with its inverse also differentiable

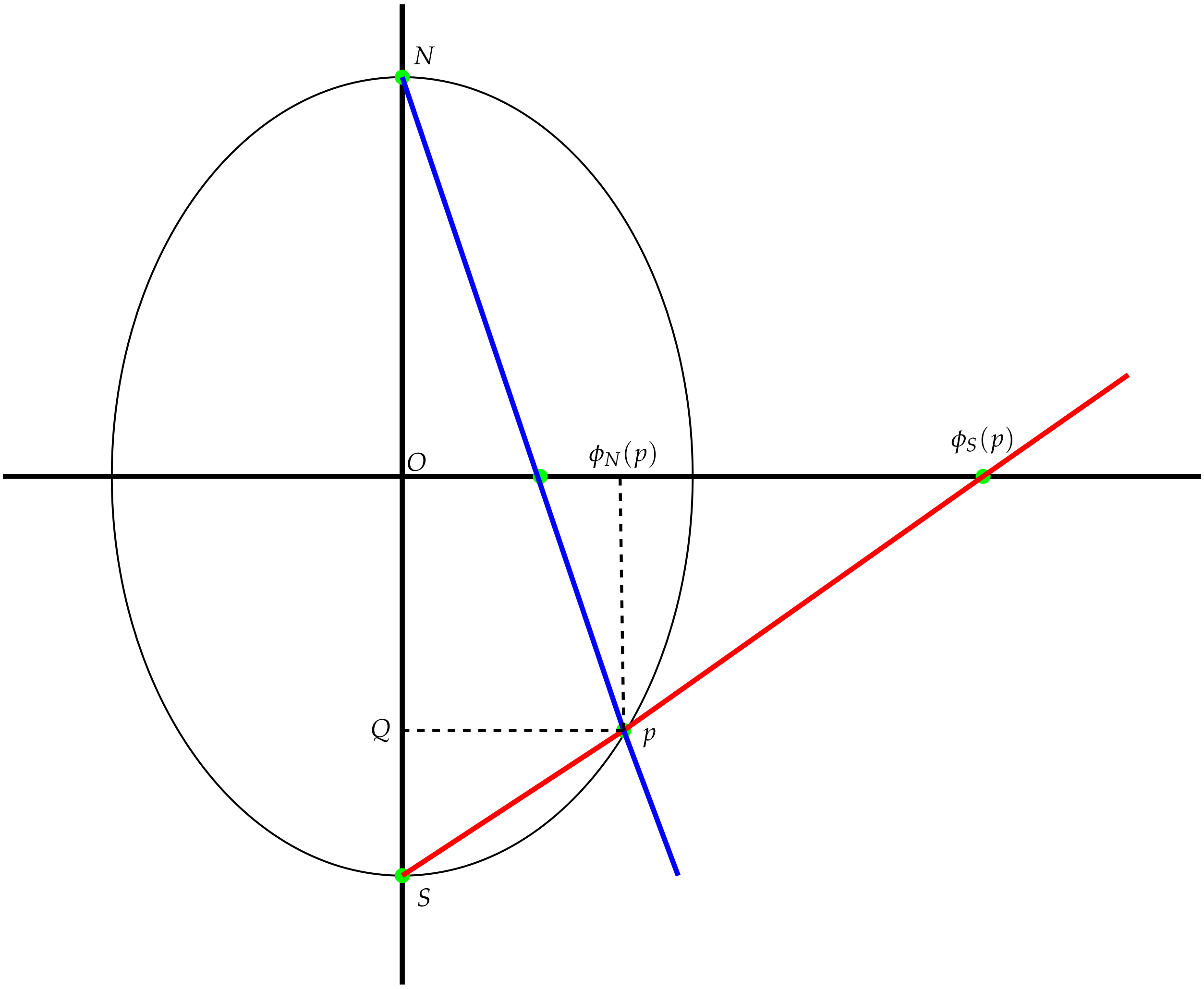


Figure 2.4: Smooth manifold

Furthermore, we have

$$\phi_N(p)O^2 = X_N^2 + Y_N^2 \text{ and } pQ^2 = x^2 + y^2 \implies \frac{pQ^2}{\phi_N(p)O^2} = \frac{x^2+y^2}{X_N^2+Y_N^2} \tag{2.3.4}$$

$$\implies \frac{x^2+y^2}{X^2+Y^2} = (1-z)^2 \tag{2.3.5}$$

$$\implies X_N^2 = \frac{x^2+y^2}{(1-z)^2} - \frac{y^2}{(1-z)^2} \text{ (Using equation 2.3.3)} \tag{2.3.6}$$

$$\implies X_N = \frac{x}{1-z} \tag{2.3.7}$$

So we have

$$\phi_N(p) = \left(\frac{x}{1-z}, \frac{y}{1-z}\right) \tag{2.3.8}$$

Using the same approach, let's consider the triangle $\phi_S(p)OS$ with $\phi_S(p) = (X_S, Y_S)$, and a point $R = (x, y, 0)$.

$$\frac{SO}{pR} = \frac{\phi_S(p)O}{\phi_S(p)R} \implies -\frac{1}{z} = \frac{Y_S}{Y_S - y} \tag{2.3.9}$$

$$\implies Y_S = \frac{y}{1+z} \tag{2.3.10}$$

Furthermore, we have:

$$\phi_S(p)O^2 = X_S^2 + Y_S^2 \tag{2.3.11}$$

$$\phi_S(p)R^2 = (x - X_S)^2 + (y - Y_S)^2 \tag{2.3.12}$$

The equations 2.3.9, 2.3.11, and 2.3.12 give

$$\frac{1}{z}^2 = \frac{X_S^2 + Y_S^2}{(x - X_S)^2 + (y - Y_S)^2} \implies z^2(X_S^2 + Y_S^2) = (x - X_S)^2 + (y - Y_S)^2 \tag{2.3.13}$$

$$\implies z^2(X_S^2 + \frac{y^2}{(1+z)^2}) = (x - X_S)^2 + (y - \frac{y}{(1+z)})^2 \tag{2.3.14}$$

$$\implies z^2 X_S^2 = (x - X_S)^2 \tag{2.3.15}$$

$$\implies zX_S = +(x - X_S) \tag{2.3.16}$$

$$\implies X_S = \frac{x}{1+z} \tag{2.3.17}$$

So we have

$$\phi_S(p) = (\frac{x}{1+z}, \frac{y}{1+z}) \tag{2.3.18}$$

$\{(U_N, \phi_N), (U_S, \phi_S)\}$ *form an atlas on* S^2.

2.3.2 Differentiable curves

This section will focus on differentiable functions and curves, tangent vector, arc length, and tangent spaces.

Definition 2.3.12 (differentiable function). *Let f be a real function of a real variable. f is said to be differentiable if it admits at all points continuous derivatives of all orders.*

Definition 2.3.13 (differentiable curve). *Let $\mathcal{M}$ be a $n-$dimensional smooth manifold, a curve or differentiable curve on $\mathcal{M}$ is a differentiable map*

$$\begin{aligned} \gamma : I &\to \mathcal{M} \\ t &\mapsto \gamma(t) \end{aligned}$$

where t is the parameter of the curve, $I \subseteq \mathbb{R}$ is an open interval of $\mathbb{R}$.
Note that $\gamma(I)$ is called trace of γ.

Examples 2.3.14 (differentiable curve)**.**

$$\begin{aligned}\gamma : \mathbb{R} &\to \mathbb{R}^3 \\ t &\mapsto (a\cos(t), b\sin(t), bt)\end{aligned}$$

is a differentiable curve.
In fact let $(x(t), y(t), z(t)) = (a\cos(t), b\sin(t), bt)$*.*

$$\textit{we have} \begin{cases} x(t) = a\cos(t) \\ y(t) = b\sin(t) \\ z(t) = bt \end{cases}$$

$x(t), y(t), z(t)$ *such defined are* C^∞ *functions, so they are differentiable. Then* γ *is a differentiable curve on* $\mathbb{R}^3$*.*

Examples 2.3.15 (not differentiable curve)**.**

$$\begin{aligned}\gamma : \mathbb{R} &\to \mathbb{R}^2 \\ t &\mapsto (x(t), y(t)) = (t, \|t\|)\end{aligned}$$

is not a differentiable curve because $\|t\|$ *is not differentiable at 0.*

Examples 2.3.16 (differentiable curve)**.**

$$\begin{aligned}\gamma : \mathbb{R} &\to \mathbb{R}^2 \\ t &\mapsto (t^3, t^2)\end{aligned}$$

is a differentiable curve.
In fact let $(x(t), y(t)) = (t^3, t^2)$*.*

$$\textit{we have} \begin{cases} x(t) = t^3 \\ y(t) = t^2 \end{cases}$$

$x(t), y(t)$ *such defined are* C^∞ *functions as polynomial functions, so they are differentiable. Then* γ *is a differentiable curve on* $\mathbb{R}^3$*.*

Examples 2.3.17 (parameterized curve of a trace)**.** *Let* $S^1 = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1\}$ *be the unit circle. The parameterized curve* γ *of* S^1 *passing through* $\gamma(0) = (0, 1)$ *is given by the equation*

$$\begin{cases} x(t) = \sin(t) \\ y(t) = \cos(t) \end{cases}$$

Definition 2.3.18 (tangent vector). *Let $\mathcal{M}$ be a manifold of dimension n. and let*

$$\begin{aligned} \gamma : \mathbb{R} &\to \mathcal{M} \\ t &\mapsto \gamma(t) = (x_1(t), x_2(t), ..., x_n(t)) \text{ be a differentiable curve on } \mathcal{M}; \end{aligned}$$

Let denote by

$$\gamma'(t) = (x'_1(t), x'_2(t), ..., x'_n(t)) \text{ the first derivative of } \gamma \text{ at the point } t$$

A vector v is a tangent vector on $\mathcal{M}$ at a point $p \in \mathcal{M}$ if

$$\gamma(0) = p \;\; and \;\; \gamma'(0) = v$$

Definition 2.3.19 (regular curve).

$$\begin{aligned} let\ \gamma : I &\to \mathcal{M} \\ t &\mapsto \gamma(t) = (x_1(t), x_2(t), ..., x_n(t)) \text{ be a differentiable curve on } \mathcal{M}; \end{aligned}$$

$\gamma(t)$ admits singular points at t if $\gamma'(t) \neq 0$. Moreover, if $\gamma'(t) \neq 0 \;\; \forall\, t \in I$, then we say that γ is a regular curve. If $\gamma'(t) = 0 \;\; \forall\, t \in I$, then γ is a singular curve.

Definition 2.3.20 (tangent line). *The tangent line to γ at t is the line passing by the point $\gamma(t)$ and containing the vector $\gamma'(t)$ with $\gamma'(t) \neq 0 \, \forall\, t \in I$.*

2.3.3 Length spaces

In this section we will focus on three different concepts namely path, length, and length metric.

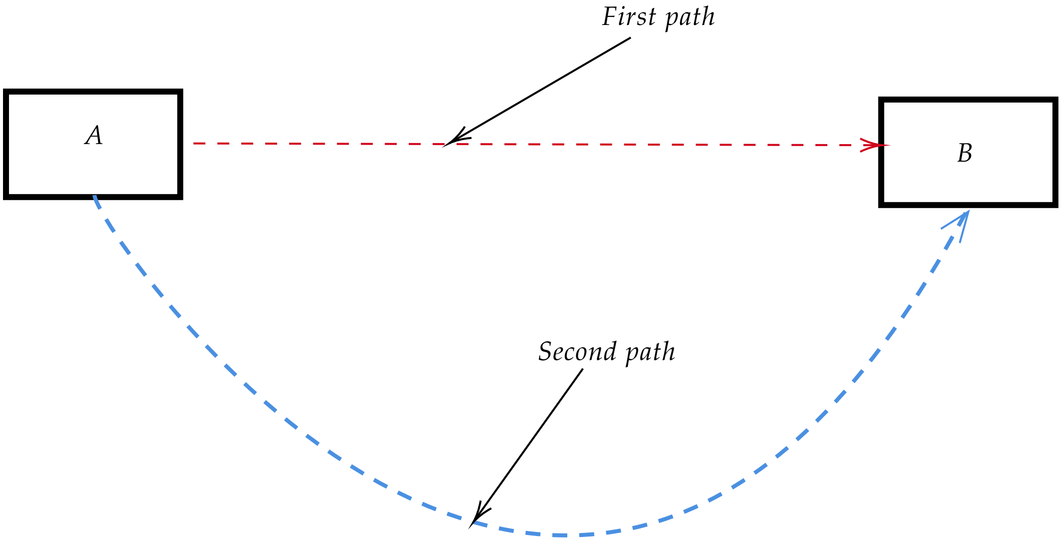


Figure 2.5: Example of paths between two points

Definition 2.3.21 (path)**.** *Let* $(\mathbf{X}, d_{\mathbf{X}})$ *be a metric space. A path in* $(\mathbf{X}, d_{\mathbf{X}})$ *is a map* $\Gamma : [a, b] \subset \mathbb{R} \mapsto \mathbf{X}$ *such that* Γ *is continuous.*

For a given path Γ, we can always associates a length to Γ and we denote by $L(\Gamma)$ the length associated to the path Γ and we say that $L(\Gamma)$ is the length of Γ.
Length is a useful concept to derive the notion of distance as it uses the concept of path to compute the distance between two points.
To illustrate the fact that different lengths are associated to different paths, we will use that graph from the figure 12. From the figure 2.5 we have a bird and an an initially on the summit of a mountain A, and they are interested in moving from the summit of the mountain A to the summit of a mountain B. From the figure, we can see the bird a path supposed linear and the an is moving along the curve. So, both are moving in two different metric spaces and so they will reach the summit of the mountain B with different distances. To reach the same goal, the bird and the an took different paths. In this section we will focus on the notion of length because

the next section will focus on minimal length which is the distance given by the shortest path between two points.

Definition 2.3.22. *length of a path*

Let Γ be a path between the point a, and b of a metric space $(\mathbf{X}, d_{\mathbf{X}})$. The length of a path Γ between the points a and b is the quantity

$$L(\Gamma) = \sup \sum_{i=1}^{k} d_{\mathbf{X}}(\Gamma(t_{i-1}), \Gamma(t)) \in [0, \infty]$$

where the t_i are subdivisions representing a partition of $[a, b]$ and the supremum is taken over all partitions of $[a, b]$.

Moreover, if $L(\Gamma)$ is finite, means that $L(\Gamma) < \infty$, then the path Γ is rectifiable.
We can also compute the length of a path Γ between two points a and b using the Riemann integral which is defined by:

$$L(\Gamma) = \int_a^b \|\Gamma(\dot{t})\|_2 dt.$$

where $\Gamma(\dot{t})$ is the derivative of $\Gamma(t)$ with respect to the parameter t.
We can notice that using the Riemann integral, the length is measured using the Euclidean distance $\|\|_2$.
Note that the length of a path joining two points in a same metric space gives rise to the concept of distance. The length itself takes two points in a metric space and assigns a non-negative number. Let's define the notion of length metric.

Definition 2.3.23. *length metric*

Let $(\mathbf{X}, d_{\mathbf{X}})$ be a metric space, Γ a path between two points $a, b \in \mathbf{X}$, and $L(\Gamma)$ the length of Γ. A length metric between the points $a, b \in \mathbf{X}$ and denoted by $d_L(x, y)$ the distance with the shortest path Γ that minimizes $L(\Gamma)$.
$d_L(x, y)$ is a length metric if:

$$d_L(x, y) = \inf_{\Gamma}\{L(\Gamma)\}.$$

When Γ doesn't exist, means that there is no path between x and y then, we have

$$d_L(x, y) = \infty.$$

Definition 2.3.24 (arc length of a curve). *Let $\gamma : I \to \mathcal{M}$ be a regular curve; For $t_0 \in I$, the arc length of γ from the point t_0 and denoted by $s(t)$ is defined as*

$$s(t) = \int_{t_0}^{t} \|\gamma'(t)\| dt, \tag{2.3.19}$$

with,

$$\|\gamma'(t)\| = \sqrt{(x_1'(t))^2 + (x_2'(t))^2 + ... + (x_n'(t))^2}$$

called the length of the tangent vector $\gamma'(t)$.

Since $\gamma(t)$ is a regular curve, we have $\gamma'(t) \neq 0$ and then $s(t)$ differentiable with

$$\frac{ds}{dt} = \|\gamma'(t)\|.$$

Examples 2.3.25 (arc of length). *Let* $\gamma(t) = (a\cos(t), a\sin(t))$ *with* $t \in [0, 2\pi]$, *then* $\gamma'(t) = (-a\sin(t), a\cos(t))$.
The length of $\gamma(t)$ *is given by*

$$\begin{aligned} s(t) =& \int_0^{2\pi} \|\gamma'(t)\| dt \\ =& \int_0^{2\pi} \sqrt{a^2\sin^2(t) + a^2\cos^2(t)} \\ =& \int_0^{2\pi} dt \\ =& 2\pi a \end{aligned}$$

2.3.4 Tangent Space

Let $\mathcal{M}$ be an $n - dimensional$ smooth manifold and $p \in \mathcal{M}$. Let $\mathcal{U} \subseteq \mathcal{M}$ and $(\mathcal{U}, \phi)$ a chart of $\mathcal{M}$ with

$$\begin{aligned} \phi : \mathbb{R} &\to \mathcal{M} \\ \lambda_0 &\mapsto \phi(\lambda_0) = p \end{aligned}$$

a curve at least C^1. The concept of tangent spaces is defined using tangent vectors. Before introducing tangent vectors, We will first introduce the notion of velocity.

Definition 2.3.26 (Velocities). *Let* $(\mathcal{M}, \tau, \mathcal{A})$ *be a topological manifold with a smooth atlas* $\mathcal{A}$.
The velocity of ϕ *at* p *is the linear map*

$$\begin{aligned} v_{\phi,p} : C^\infty(\mathcal{M}) &\to \mathbb{R} \\ f &\mapsto v_{\phi,p}(f) = (fo\phi)'(\lambda_0) \end{aligned}$$

where

$$C^\infty(\mathcal{M}) = \{f : \mathcal{M} \to \mathbb{R} \mid f \textit{is a smooth function}\}$$

The addition and multiplication on $C^\infty(\mathcal{M})$ *are defined as follow:*

1. $(f+g)(p) \mapsto f(p) + g(p)$

2. $(\lambda\ .g) \mapsto \lambda.g(p)$.

2.3.5 Tangent vector spaces

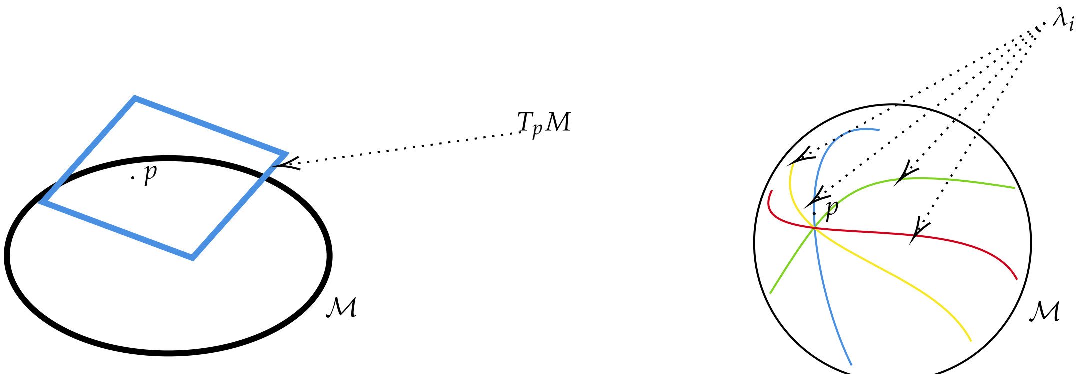
In differential geometry, a tangent vector is a vector that is tangent to a curve on a smooth manifold or surface at a given point on that smooth manifold.

Definition 2.3.27 (Tangent spaces)**.** *Let $\mathcal{M}$ be a smooth manifold and $p \in \mathcal{M}$. The tangent space to $\mathcal{M}$ at p and denoted $\mathbf{T}_p\mathcal{M}$ is a collection of all possible tangent vectors to all possible smooth curves through the point p.*

$$\mathbf{T}_p\mathcal{M} = \{v_{\lambda,p} \mid \lambda \textit{ is a smooth curve}\}$$

A tangent space on a manifold $\mathcal{M}$ at $p \in \mathcal{M}$ is the vectors space tangent to $\mathcal{M}$ through p.

Examples 2.3.28 (Tangent vectors on a manifold)**.** *In figure 2.6, the picture on the left side shows an example of Riemannian manifold $\mathcal{M}$ with a tangent spaces on it passing through a point $p \in \mathcal{M}$. The picture on the right side shows an example of Riemannian manifold with some smooth curves passing through a point p on that manifold.*



Manifold with a tangent spaces at p

Manifold with some smooth curves passing through p

Figure 2.6: Tangent spaces

2.3.6 Riemannian metric

Riemannian metrics are very small objects that can be seen as a family of smoothly varying inner product on tangent spaces of a smooth manifold. They can be used to measure distances between two points on a smooth manifold or to measure the lengths between tangent vectors. Every smooth manifold is equipped with different Riemannian metrics.

Definition 2.3.29. *Riemannian metric*

Let $\mathcal{M}$ be a smooth manifold, let $x \in \mathcal{M}$, and $\mathbf{T}_x\mathcal{M}$ the tangent spaces on $\mathcal{M}$ at x. The Riemannian metric on $\mathbf{T}_x\mathcal{M}$ of $\mathcal{M}$ which associates to every $x \in \mathcal{M}$ is a symmetric positive defined bi-linear function

g_x such that

$$g_x : \mathbf{T}_x\mathcal{M} \times \mathbf{T}_x\mathcal{M} \to \mathbb{R}.$$

Let $u, v \in \mathbf{T}_x\mathcal{M}$ and g_x a Riemannian metric. Then g_x satisfies the following conditions:

1. $g_x(u, v) = g_x(v, u)$;
2. $g_x(u, u) \geq 0$;
3. $g_x(u, v + w) = g_x(u, v) + g_x(u, w)$ For all $w \in \mathbf{T}_x\mathcal{M}$.

Given a smooth manifold $\mathcal{M}$ and a Riemannian metric g we can derive the concept of Riemannian manifold.

Definition 2.3.30. *Riemannian manifold*

A Riemannian manifold is defined as a couple $(\mathcal{M}, g)$ where $\mathcal{M}$ is a smooth manifold and g is a Riemannian metric on $\mathcal{M}$.

g_x will be called a pseudo Riemannian metric if:

1. $g_x(u, v) = g_x(v, u)$;
2. $g_x(u, u) \geq 0$ with $g_x(u, u) = 0$ if $u = 0$;
3. $g_x(u, v) = 0 \forall u \in \mathbf{T}_x\mathcal{M} \implies v = 0$
4. $g_x(u, v + w) = g_x(u, v) + g_x(u, w)$ For all $w \in \mathbf{T}_x\mathcal{M}$.

We say that the couple $(\mathcal{M}, g)$ is a pseudo-Riemannian manifold.
Note that if g is a Riemannian metric, then all eigen values are positive, otherwise, if g is a pseudo-Riemannian metric, then we can have both positive and negative eigen values.
Let i be the number of positive eigen values and j the number of negative eigen values for a pseudo-Riemannian manifold. Then, the couple (i, j) is called index of metric. Furthermore, if $i = j$, then the metric g is called the Lorentz metric and the couple $(\mathcal{M}, g)$ the Lorentz manifold.

Definition 2.3.31. *Riemannian distance*

A Riemannian distance between two points $\mathbf{x}$ *and* $\mathbf{y}$ *on a manifold vM is the minimum length over all possible smooth curves passing through* $\mathbf{x}$ *and* $\mathbf{y}$ *on $\mathcal{M}$. The smooth curve with the minimum length is called* ***geodesic curve****.*

Definition 2.3.32. *The* ***geodesic*** *is the smooth curve representing the shortest path between two points on a Riemannian manifold.*

Let γ being a curve connecting two points a, b on a manifold $\mathbf{M}$ the length of γ and noted $\mathbf{L}(\mathbf{fl})$ is defined by

$$\mathbf{L}(\mathbf{fl}) = \int_0^1 |\dot{\gamma}(t)|\, dt \ \ such \ \ that \ \ \gamma(0) = a \ \ and \ \ \gamma(1) = b$$

The shortest path between a and b is called geodesic and denoted γ_{ab}.
We have

$$\mathbf{L}(\mathbf{fl_{ab}}) = d_M(a,b) = \inf_{\gamma \in \mathbf{M}} \{L(\gamma(t)) \ \mid \gamma(t) \in \mathbf{M},\ t \in [0,1],\ \gamma(0) = a,\ \gamma(1) = b\}$$

When having a data point distributed along a non linear sub-manifold, the geodesic is used to determine the minimum distance from each point to its K-nearest neighbors.

Examples 2.3.33. *Geodesic*

2.3.7 Exponential Map

An exponential map is mapping a tangent vector into the point $\gamma(1)$ on the manifold. An exponential map takes as input a tangent vector $v \in \mathbf{T_x M}$ and maps it into the point $\gamma(1)$ where γ is smooth curve on $\mathbf{M}$.

$$\begin{aligned} \mathbf{Exponential\ \ Map} : \mathbf{T_x M} &\longrightarrow \mathbf{M} \\ v \quad &\longmapsto \gamma(1) \end{aligned}$$

The function that takes two points on a manifold and maps them into a tangent vector is the inverse map of the exponential map and is called **logarithm map**

Examples 2.3.34. *Exponential map*

2.3.8 Logarithm Map

A logarithm map is the inverse of the exponential map that takes two points $\mathbf{x}_i$, $\mathbf{x}_j \in \mathbf{M}$ within the data points and map them into a tangent vector $\mathbf{v} = \overrightarrow{\mathbf{x}_i\mathbf{x}_j} \in \mathbf{T_x M}$.

$$\begin{aligned} \mathbf{Logarithm\ \ Map} : \quad \mathbf{M} \quad &\longrightarrow \mathbf{T_x M} \\ (\mathbf{x}_i, \mathbf{x}_j) &\longmapsto \overrightarrow{\mathbf{x}_i\mathbf{x}_j} = log_{\mathbf{x}_i}(\mathbf{x}_j) \end{aligned}$$

.

Examples 2.3.35. *Logarithm map https://analyticsindiamag.com/a-guide-to-locally-linear-embedding-for-dimensionality-reduction/*

Chapter 3

Non Linear Methods for Dimensionality Reduction

Nonlinear dimensionality reduction (NLDR) algorithms are a class of techniques that are used to transform high-dimensional data into low-dimensional space while preserving the essential geometric properties of the original data. These algorithms are important for data visualization, data compression, and feature extraction, and have found applications in various fields, including computer vision, machine learning, and data analysis. In this section, we will develop algorithms, namely Locally Linear Embedding (LLE), Isomap, and Hessian Locally Linear Embedding (HLLE). We will discuss the principles, advantages, and limitations of each algorithms.

3.0 Isometric Mapping (IsoMap)

Linear dimensionality reduction methods are effective when data points are concentrated around a hyperplane. However, when data points are concentrated around a non-linear manifold, a different approach is required. In this case, the underlying geometry structure is characterized by the manifold, rather than a subspace. Therefore, using Euclidean distance to measure distances between points will produce inaccurate results. To solve this problem, we must use the geodesic metric on the underlying manifold, which represents the shortest path between two points on a sub-manifold of a Riemannian manifold. This metric provides a better measurement of the smallest distance between data points lying on a smooth manifold. Non-linear dimensionality reduction methods focus on measuring the dissimilarity of points by using the neighbor structure of the data. Points are considered dissimilar if they do not belong to the same neighborhood, and a metric is required to measure such dissimilarities. However, different non-linear dimensionality reduction methods use different approaches, leading to different metrics. Isometric mapping, for example, uses the geodesic metric of the underlying manifold. However, it is impossible to measure the true distance between data points, so an approximation is necessary. A graph, consisting of points connected by edges, is used to accurately approximate the geodesic distance. An isometric map is any map that preserves the distance defined on the data, and the process of mapping while preserving distance is called isometric mapping. Isometric mapping aims to preserve the distances between points on a manifold while mapping them into a lower-dimensional space. Simply put, isometric mapping ensures that the geodesic distance defined on the original data is equal to the Euclidean distance defined on the new data embedded in a lower-dimensional space. We can define Isometric mapping algorithm as a powerful nonlinear dimensionality reduction technique that aims to preserve the global structure of high-dimensional data in a lower-dimensional space. It was introduced by Joshua B. Tenenbaum, Vin de Silva, and John C. Langford in 2000 and has since been widely used in various applications, including image and speech processing, computer vision, and bio-informatics.

The basic idea behind Isomap is to represent the high-dimensional data as a graph of interconnected points, where the distances between the points reflect the underlying structure of the data. The construction of $K-$ nearest neighbor graph goes as follow: each data point $\mathbf{X}_i$ is connected by edges to another data point $\mathbf{X}_j$ if $i \neq j$ and $\mathbf{X}_j$ is among the $K-$ nearest neighbors of $\mathbf{X}_i$ or vice versa. In particular, Isomap uses a geodesic distance metric, which measures the shortest path between two points along the manifold or surface that the data lies on, rather than the Euclidean distance metric, which measures the straight-line distance between the points.

To construct the graph, Isomap first selects k-nearest neighbors for each data point based on the Euclidean distance metric. Then, it computes the pairwise geodesic distances between the data points using a variant of the Floyd-Warshall algorithm, which finds the shortest path between all pairs of points in the graph.

Once the geodesic distances are computed, Isomap performs classical multidimensional scaling (MDS) to embed the data in a lower-dimensional space, while preserving the pairwise geodesic distances as much as possible. In other words, Isomap tries to find a lower-dimensional representation of the data that best approximates the underlying manifold or surface that the data lies on, based on the geodesic distances between the points.

To express this mathematically, let $X = \{x_1, x_2, ..., x_n\}$ be a set of data points in $\mathbb{R}^D$ that lie on a smooth d-dimensional manifold $\mathcal{M}$, where d is much less than D. Let $d_{\mathcal{M}}$ be the geodesic distance defining on $\mathcal{M}$. We can define an embedding map,

$$\begin{aligned} f : \mathcal{M} &\to \mathbb{R}^D \\ x_i &\mapsto f(x_i) = y_i \end{aligned}$$

This embedding map preserves the distance in $\mathbb{R}^d$, which means that for any two points x and z in $\mathcal{M}$, the distance between their corresponding mapped points $f(x)$ and $f(z)$ in $\mathbb{R}^d$ is equal to the geodesic distance between x and z in $\mathcal{M}$. In other words,

$$d_2(f(x), f(z)) = d_{\mathcal{M}}(x, z) \; \forall \; x, z \in \mathcal{M}.$$

In particular, we have

$$d_2(y_i, y_j) = d_{\mathcal{M}}(x_i, x_j) \; \forall \; 1 \leq i \leq i \leq j \leq n.$$

We say then

$$y = \{y_1, y_2, ..., y_n\} \text{ is a dimensional reduction of } X.$$

Note that since f cannot be explicitly defined on $\mathcal{M}$, we must provide an approximation of $d_{\mathcal{M}}$ on the data X using the graph distance. With $d_{\mathcal{M}}$ the geodesic distance on $\mathcal{M}$ on the data X. Assuming that there is a well-defined neighborhood system on the data set X, we can construct a graph G

$$G = [X, F] \ \ such \ \ that \ \ (x_i, x_j) \in F \iff x_i, \ x_j \text{ are adjacent.}$$

To construct the graph metric d_G on G, on G, we define the graph distance between $x, y \in X$ as follows.

1. If $(x, y) \in F$, then, $d_G(x, y) = d_2(x, y)$;
2. If $(x, y) \notin F,$ let consider a path Λ connecting x and y.

$$\Lambda = (x_0, x_1, ..., x_{s+1})$$

The path distance is defined by:

$$d_\Lambda = d(x_0, x_1) + d(x_1, x_2) + ... + d(x_s, x_{s+1}).$$

Let Γ be the set of all paths that connect x and y.Then the graph distance between x and y is defined by:

$$d_G(x, y) = \min_{y \in \Gamma} d_\Lambda(x, y)$$

The graph distance d_G can approximate the geodesic distance $d_{\mathcal{M}}$ well if the data points are dense enough on the manifold $\mathcal{M}$. To address this issue, we can add a kernel to an underlying metric defined on $\mathcal{M}$.

Given a data set X lying on a manifold $\mathcal{M} \subset \mathbb{R}^D$, and an isometric mapp

$$f : \mathcal{M} \to \mathbb{R}^d$$
$$x_i \mapsto y_i$$

The isometric kernel of a graph $G = [X, F]$ is constructed from the graph metric $D_G = [d_G(i, j)]$ on G. Here $d_G(i, j)$ is the graph distance between x_i and x_j.

When X is dense enough on $\mathcal{M}$, the graph distance $d_G(i, j)$ approximates well the geodesic distance $d_{\mathcal{M}(i,j)}$ which implies to the graph metric D_G approximating well the geodesic metric D_M. Leading to the graph metric approximating well the Euclidean metric on the dimensionnality reduction set of X denoted $y = \{y_1, ..., y_n\}$. we then have

$$D_G \approx D \stackrel{\text{def}}{=} [d_2(y_i, y_j)]$$

Definition 3.0.1 (Centered matrix, Centralizing matrix).

- *the centered matrix of the matrix X is denoted $\hat{H} = \{\hat{x_1}, \hat{x_2}, ..., \hat{x_2}\}$ where $\hat{x}_i = x_i - \bar{x}_i$, with $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$*
- *Let $\nVdash = [1, 1, ..., 1]^t \in \mathbb{R}^n$ be the matrix of all one, $E = \nVdash\nVdash^{\approx}$, and I the $n \times n$ identity matrix. The centralizing matrix is the $n \times n$ matrix defined by*

$$H = I - \frac{1}{n} E$$

Lemma 3.0.1. *If H is the centralizing matrix, then H has the following properties:*

1. $H^2 = H$;
2. $\mathbb{1}^t H = H\mathbb{1} = 0$;
3. *X is centered data set if and only if $XH = X$;*
4. *A positive semi-defined matrix C is a centered Gram matrix if and only if $HCH = C$.*

Lemma 3.0.2. *Let X be a data matrix and G be its Gram matrix. The centered data set of X is denoted XH and the centering Gram matrix of X is*

$$G^c = HGH$$

.

Theorem 3.0.3. *The Euclidean squared distance matrix S and the centering Gram matrix G^c of a data set X are linked by the following relationship:*

$$G^c = \frac{1}{2}S^c$$

.

3.0.1 Advantages

The Isomap algorithm has several advantages over other nonlinear dimensionality reduction techniques. For example, it can handle complex manifolds and surfaces, such as loops, twists, and holes, that cannot be represented by linear or affine transformations. Moreover, Isomap can preserve the global structure of the data, even if the data is highly nonlinear or has a high-dimensional intrinsic structure.
However, Isomap also has some limitations. It may suffer from the curse of dimensionality, where the computational complexity and memory requirements increase exponentially with the dimensionality of the data. Moreover, Isomap may be sensitive to noise and outliers in the data, as it relies on the geodesic distances between the points.

3.0.2 Conclusion

In summary, Isomap is a powerful and versatile dimensionality reduction technique that can be used to analyze and visualize complex high-dimensional data. Its ability to preserve the global structure of the data and handle nonlinear manifolds makes it particularly useful for applications where the intrinsic structure of the data is important. However, careful parameter selection and preprocessing are necessary to obtain meaningful and stable embeddings.

3.1 Locally Linear Embedding (LLE).

This section focuses on Locally Linear Embedding (LLE), it explains how the algorithms works, outline its advantages and limitations.

Locally Linear Embedding (LLE) is a powerful unsupervised machine learning nonlinear dimensionality reduction technique that aims to preserve the local geometry of high-dimensional data in a lower-dimensional space. It was introduced by Sam T. Roweis and Lawrence K. Saul in 2000 and has since been widely used in various applications, including image and speech processing, computer vision, and bioinformatics.

The basic idea behind LLE is to approximate the high-dimensional data as a locally linear patchwork of lower-dimensional manifolds, and then embed these manifolds in a lower-dimensional space, while preserving the local relationships between the data points. In other words, LLE tries to find a lower-dimensional representation of the data that is consistent with the local linear structure of the data, rather than the global structure.

The assumption behind LLE is that each data point and its neighborhood lie on a local linear path of a manifold.

Suppose we have $N-$real values vectors $\mathbf{X}_i$ lying on a smooth manifold $\mathcal{M}$. To compute a representation of these vectors into a lower space, LLE compute a local neighborhood $\mathbf{X}_j$ of each vectors $\mathbf{X}_i$, such that $\mathbf{X}_j$ are closest as possible to $\mathbf{X}_i$ means that the distance between $\mathbf{X}_i$ and $\mathbf{X}j$ is minimized as much as possible. We can then reconstruct each data point $\mathbf{X}_i$ from its local neighborhood $\mathbf{X}_j$ using the linear coefficients associates to it. Note that both $\mathbf{X}_i$ and its neighborhood $\mathbf{X}_j$ lie on a locally linear path of the manifold $\mathcal{M}$. LLE does that using the Riemannian distance.

The LLE algorithm consists of three main steps: neighborhood selection, weight computation, and embedding. In the first step, LLE selects k-nearest neighbors for each data point based on the Euclidean distance metric. The neighborhood graph is then constructed by connecting each point to its k-nearest neighbors. In the second step, LLE computes the weights for each data point as a linear combination of its neighbors, such that the sum of the weights is equal to one and the reconstructed data point is as close as possible to the original data point. The weights are computed using the optimization problem that minimizes the reconstruction error:

$$\epsilon(w) = \sum_i \|X_i - \sum_j W_{ij} X_j\|^2, \tag{3.1.1}$$

Where W_{ij} are the matrix of weight that minimizes the reconstruction error $\epsilon(w)$ with $W_{ij} = 0$ if $\mathbf{X}_j$ does not belong to the set of neighborhood of $\mathbf{X}_i$ because W_{ij} respect the constraint that each data point $\mathbf{X}_i$ is reconstructed only from its neighborhood $\mathbf{X}_j$.

The Equation 3.1.1 comes from the fact that since the exact number of neighbors for each data point is unknown, some error may occur during the reconstruction of each point. That error can be measured using the formula:

$$\epsilon = \|X_i - \sum_j W_{ij} X_j\|^2.$$

The total sum of errors is then recorded by looping the reconstruction error on a range of n. We then obtain Equation 3.1.1.

The linear interpolation of each point $\vec{X}_i$ by its neighbors X_j is given by:

$$X_j = X_i + \sum_j W_{ij} X_i X_j \|$$

In the final step, LLE embeds the data in a lower-dimensional space by minimizing the cost function that measures the difference between the distances of the original data points and their reconstructed counterparts in the low-dimensional space:

$$C(Y) = \sum_i \|Y_i - \sum_j W_{ij} Y_j\|^2, \tag{3.1.2}$$

where Y_i is the embedding of the i^{th} data point in the lower-dimensional space.

3.1.1 LLE's algorithm

1. Choose the $K-$ nearest neighbors associated to each data points;
2. Determine the matrix of weigh W_{ij} that minimizes the reconstruction error given by the equation 3.1.1;
3. Define the new vector space Y that minimizes the cost of equation 3.1.2.

3.1.2 Choice of KNN

To choose the optimal k-nearest neighbors, we can use the module neighbors.NearestNeighbors from the scikit-learn's library. The neighbors.NearestNeighbors search will calculate the distances and nearest neighbors for each point in the input X_i and then returns only the indices of the neighbors (excluding the first column, which is always the index of the point itself). To implement the neighbors.NearestNeighbors algorithm, we can consider four key parameters as follow:

1. K: an integer representing the number of neighbors to search for;
2. t: a float representing the threshold used in the distance metric (default is 2.0 for Euclidean distance);
3. $dist - metric$: a string representing the distance metric to use (default is "euclidean");
4. algorithm: a string representing the algorithm to use for finding nearest neighbors (default is "$ball - tree$").

3.1.3 Computation of matrix of weights

When the k is greater than the number of input dimensions of the data, the matrix of weights might take many zeros because there are equations than unknowns which can lead to overfitting and poor generalization performance. This problem can be formulated as a regularization problem.

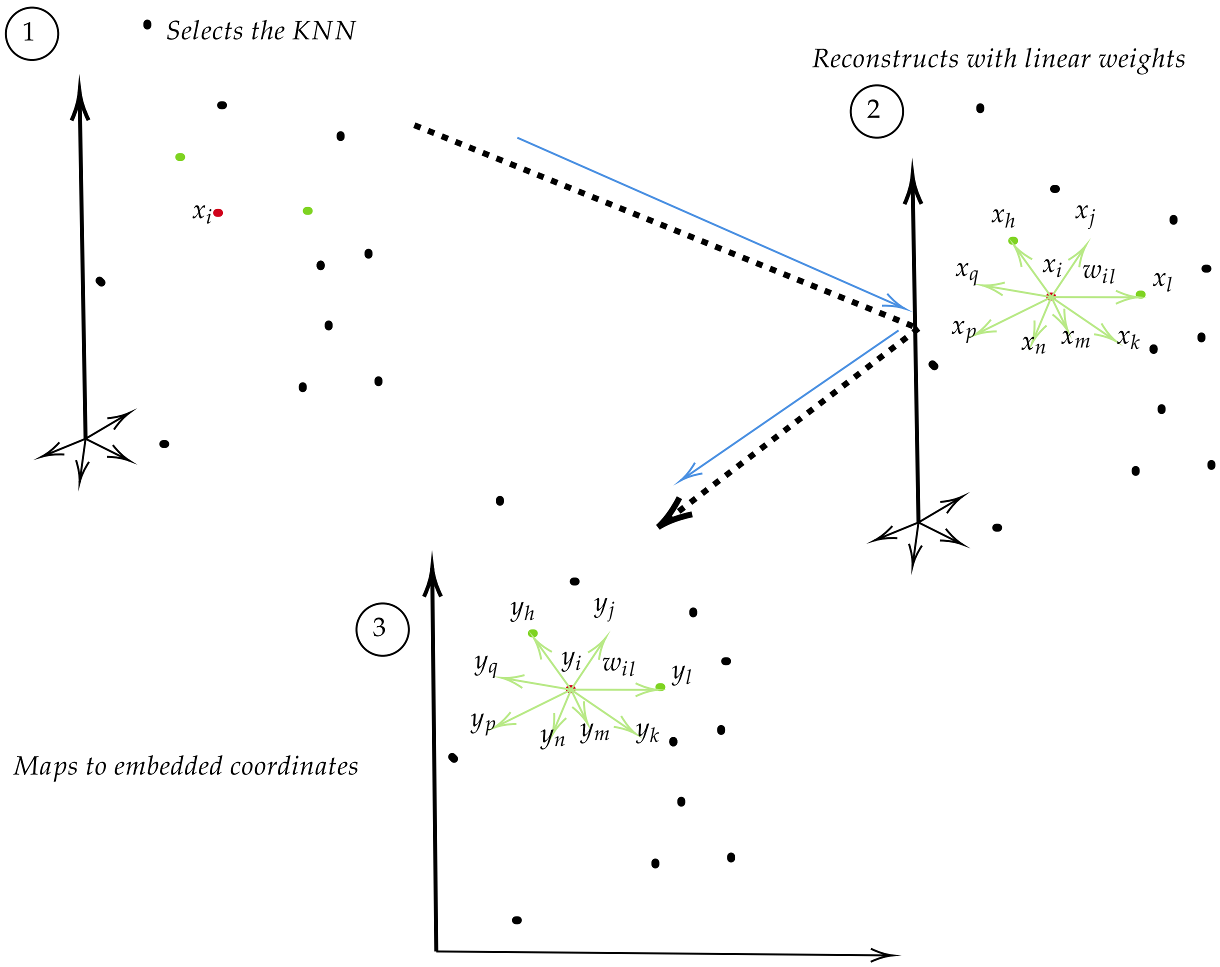


Figure 3.1: LLE workflow

3.1.4 Advantages of LLE

One of the advantages of LLE is its robustness to noise and outliers in the data, as it only considers the local structure of the data. Moreover, LLE can preserve the nonlinear structure of the data, even if the data lies on a complex manifold or has a non-convex shape and it requires only one parameter to choose which is the number of nearest neighbors.
However, LLE also has some limitations.

3.1.5 Limitations of LLE

LLE is sensitive to the choice of the number of nearest neighbors k because if k is chosen to be too small or too large, it may not be able to accommodate the geometry of the original data and also the quality of the embedding may vary depending on the chosen value. Moreover, LLE does not provide a unique solution, and the embedding may depend on the initial conditions and the optimization method used.

3.1.6 Conclusion

In summary, LLE is a powerful and versatile dimensionality reduction technique that can be used to analyze and visualize complex high-dimensional data. Its ability to preserve the local structure of the data makes it particularly useful for applications where the global structure of the data is unknown or irrelevant. However, careful parameter selection and optimization are necessary to obtain meaningful and stable embeddings.

3.2 Hessian Locally Linear Embedding

The Hessian Locally Linear Embedding (HLLE) algorithm is a nonlinear dimensionality reduction technique that aims to preserve the global and local structures of high-dimensional data in a lower-dimensional space. It was proposed by R. R. Coifman and S. Lafon in 2006 and is an extension of the popular Locally Linear Embedding (LLE) algorithm.

The basic idea behind HLLE is similar to LLE in that it seeks to represent the high-dimensional data as a graph of interconnected points, where the distances between the points reflect the underlying structure of the data. However, HLLE uses a Hessian-based approach to estimate the local structure of the data, which is more robust to noise and outliers than the tangent-based approach used in LLE.

To construct the graph, HLLE first selects k-nearest neighbors for each data point based on the Euclidean distance metric. Then, it computes the Hessian matrix of the distance function for each point and its k-nearest neighbors. The Hessian matrix is a symmetric matrix that describes the local curvature of the distance function at the point, and its eigenvectors and eigenvalues provide information about the local geometry of the data. Next, HLLE computes the linear coefficients that best approximate the distances between each data point and its k-nearest neighbors, subject to the constraint that the coefficients sum to one. These linear coefficients are used to define the weights of the edges in the graph, which connect each data point to its k-nearest neighbors. Finally, HLLE performs nonlinear dimensionality reduction using an optimization algorithm that seeks to preserve the pairwise distances between the data points in the graph, while minimizing the total squared error of the approximation. This optimization problem can be solved using various techniques, such as gradient descent, singular value decomposition, or sparse matrix factorization.

The HLLE algorithm has several advantages over other nonlinear dimensionality reduction techniques. For example, it can handle noisy and high-dimensional data, as it uses the local geometry of the data to estimate the global structure. Moreover, HLLE can preserve both the global and local structures of the data, which is important for applications where both scales of structure are relevant.

However, HLLE also has some limitations. For example, it may be sensitive to the choice of parameters, such as the number of nearest neighbors and the regularization parameter, which can affect the quality of the embedding. Moreover, HLLE may be computationally expensive, especially for large datasets or high-dimensional embeddings. In summary, the Hessian Locally Linear Embedding algorithm is a powerful and versatile nonlinear dimensionality reduction technique that can be used to analyze and visualize complex high-dimensional data. Its ability to preserve both the global and local structures of the data and handle noisy and high-dimensional data makes it particularly useful for applications where the intrinsic structure of the data is important. However, careful parameter selection and preprocessing are necessary to obtain meaningful and stable embeddings.

Chapter 4

Conclusion and perspectives

4.0 Conclusion

This thesis provides a comprehensive overview of three popular NLDR algorithms, namely LLE, Isomap, and HLLE. We have discussed the basic principles, advantages, and limitations of each algorithm, and compared their performance on various datasets. We have also provided guidelines for selecting the optimal parameters for each algorithm, and discussed the potential applications of these algorithms in different fields. Overall, NLDR algorithms are powerful tools for transforming high-dimensional data into low-dimensional space while preserving the essential geometric properties of the data, and will continue to play an important role in data analysis and machine learning.

4.1 Perspectives

The conclusion of this thesis highlights the importance of NLDR algorithms in data analysis and machine learning. The overview of three popular algorithms, LLE, Isomap, and HLLE, provides a comprehensive understanding of their basic principles, advantages, and limitations.
The conclusion also emphasizes the potential applications of these algorithms in different fields, indicating their versatility and broad applicability. The ability of NLDR algorithms to preserve the essential geometric properties of the data while transforming high-dimensional data into low-dimensional space is an essential property that makes them powerful tools in data analysis and machine learning.

Overall, this conclusion provides a well-rounded perspective on the importance and potential of NLDR algorithms, providing insights into how these algorithms can be used to address different data analysis problems. It emphasizes the importance of understanding the basic principles and limitations of these algorithms to obtain the best possible results for a given application. Finally, it highlights the need for continued research to develop new and improved NLDR algorithms that can further enhance the capabilities of data analysis and machine learning.

Appendix A

Python codes

You might list here the Python codes used in the work.

References

[1] Lokenath Debnath and Piotr Mikusinki. *Introduction to Hilbert spaces with applications*. Academic press, 2005.

[2] Christopher Heil. Metric and normed spaces. In *Introduction to Real Analysis*, pages 15–32. Springer, 2019.

[3] S Kumaresan. Topology of metric spaces, alpha science international, 2011.

[4] YU XIAO. An introduction to hilbert spaces and the heisenberg uncertainty principle. 2017.

References

[1] Lokenath Debnath and Piotr Mikusinki. *Introduction to Hilbert spaces with applications*. Academic press, 2005.

[2] Christopher Heil. Metric and normed spaces. In *Introduction to Real Analysis*, pages 15–32. Springer, 2019.

[3] S Kumaresan. Topology of metric spaces, alpha science international, 2011.

[4] YU XIAO. An introduction to hilbert spaces and the heisenberg uncertainty principle. 2017.