UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

Development of a software prototype for interactive dimensionality reduction including a representation quality measurement

Trabajo de integración curricular presentado como requisito
para la obtencióna
del título de Ingeniero en tecnologías de la Información

Autor:

Josué Nicolás Marín Gaviño

Tutor:

Manuel Eugenio Morocho Cayamcela, Ph.D.

Urcuquí - Abril, 2021

Urcuquí, 4 de agosto de 2021

SECRETARÍA GENERAL
(Vicerrectorado Académico/Cancillería)
ESCUELA DE CIENCIAS MATEMÁTICAS Y COMPUTACIONALES
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
ACTA DE DEFENSA No. UITEY-ITE-2021-00027-AD

A los 4 días del mes de agosto de 2021, a las 11:30 horas, de manera virtual mediante videoconferencia, y ante el Tribunal Calificador, integrado por los docentes:

Presidente Tribunal de Defensa	Dr. MAYORGA ZAMBRANO, JUAN RICARDO , Ph.D.
Miembro No Tutor	Dr. ANTON CASTRO , FRANCESC , Ph.D.
Tutor	Dr. MOROCHO CAYAMCELA, MANUEL EUGENIO , Ph.D.

El(la) señor(ita) estudiante **MARIN GAVIÑO, JOSUE NICOLAS**, con cédula de identidad No. **1723331573**, de la **ESCUELA DE CIENCIAS MATEMÁTICAS Y COMPUTACIONALES**, de la Carrera de **TECNOLOGÍAS DE LA INFORMACIÓN**, aprobada por el Consejo de Educación Superior (CES), mediante Resolución **RPC-SO-43-No.496-2014**, realiza a través de videoconferencia, la sustentación de su trabajo de titulación denominado: **Development of a software prototype for interactive dimensionality reduction including a representation quality measurement** , previa a la obtención del título de **INGENIERO/A EN TECNOLOGÍAS DE LA INFORMACIÓN.**

El citado trabajo de titulación, fue debidamente aprobado por el(los) docente(s):

Tutor	Dr. MOROCHO CAYAMCELA, MANUEL EUGENIO , Ph.D.

Y recibió las observaciones de los otros miembros del Tribunal Calificador, las mismas que han sido incorporadas por el(la) estudiante.

Previamente cumplidos los requisitos legales y reglamentarios, el trabajo de titulación fue sustentado por el(la) estudiante y examinado por los miembros del Tribunal Calificador. Escuchada la sustentación del trabajo de titulación a través de videoconferencia, que integró la exposición de el(la) estudiante sobre el contenido de la misma y las preguntas formuladas por los miembros del Tribunal, se califica la sustentación del trabajo de titulación con las siguientes calificaciones:

Tipo	Docente	Calificación
Miembro Tribunal De Defensa	Dr. ANTON CASTRO , FRANCESC , Ph.D.	8,0
Presidente Tribunal De Defensa	Dr. MAYORGA ZAMBRANO, JUAN RICARDO , Ph.D.	9,5
Tutor	Dr. MOROCHO CAYAMCELA, MANUEL EUGENIO , Ph.D.	9,5

Lo que da un promedio de: **9 (Nueve punto Cero)**, sobre 10 (diez), equivalente a: **APROBADO**

Para constancia de lo actuado, firman los miembros del Tribunal Calificador, el/la estudiante y el/la secretario ad-hoc.

Certifico que *en cumplimiento del Decreto Ejecutivo 1017 de 16 de marzo de 2020, la defensa de trabajo de titulación (o examen de grado modalidad teórico práctica) se realizó vía virtual, por lo que las firmas de los miembros del Tribunal de Defensa de Grado, constan en forma digital.*

JOSUE
NICOLAS
MARIN GAVINO

Firmado digitalmente
por JOSUE NICOLAS
MARIN GAVINO
Fecha: 2021.08.20
17:26:58 -07'00'

MARIN GAVIÑO, JOSUE NICOLAS
Estudiante

Firmado electrónicamente por:
**JUAN RICARDO
MAYORGA
ZAMBRANO**

Dr. MAYORGA ZAMBRANO, JUAN RICARDO , Ph.D.
Presidente Tribunal de Defensa

Firmado electrónicamente por:
MANUEL EUGENIO MOROCHO CAYAMCELA

Dr. MOROCHO CAYAMCELA, MANUEL EUGENIO , Ph.D.
Tutor

FRANCESC ANTON CASTRO

Signé électroniquement par FRANCESC ANTON CASTRO
cn=FRANCESC ANTON CASTRO, ou= ENTIDAD DE CERTIFICACION DE INFORMACION, o= SECURITY DATA S.A. 1, c= EC
Date: 2021.08.19 13:41:20 ECT

Dr. ANTON CASTRO , FRANCESC , Ph.D.
Miembro No Tutor

TATIANA BEATRIZ TORRES MONTALVAN

Firmado digitalmente por TATIANA BEATRIZ TORRES MONTALVAN
Fecha: 2021.08.06 11:26:00 -05'00'

TORRES MONTALVÁN, TATIANA BEATRIZ
Secretario Ad-hoc

Autoría

Yo, **Josué Nicolás Marín Gaviño**, con cédula de identidad **1723331573**, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el autor del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, Marzo 28 del 2021.

Firmado electrónicamente por:
JOSUE NICOLAS
MARIN GAVINO

Josué Nicolás Marín Gaviño
CI: 1723331573

Autorización de publicación

Yo, **Josué Nicolás Marín Gaviño**, con cédula de identidad **1723331573**, cedo a la Universidad de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Urcuquí, Marzo del 2020.

Firmado electrónicamente por:
JOSUE NICOLAS
MARIN GAVINO

Josué Nicolás Marín Gaviño
CI: 1723331573

Dedication

To my mother Adriana Gaviño and my father José Marín.

Acknowledgments

I want to express my sincere gratitude to several individuals and organizations for supporting me throughout my Graduate study. First, I wish to express my sincere appreciation to professors Diego Peluffo and Manuel Morocho for their enthusiasm, patience, insightful comments, helpful information, practical advice, and unceasing ideas that have helped me tremendously in my research writing this thesis. Their immense knowledge, profound experience, and professional expertise in Dimensionality Reduction Techniques have enabled me to complete this research successfully. Without their support and guidance, this project would not have been possible.

This journey would not have been possible without the support of my family, professors and mentors, and friends. To my family, thank you for encouraging me in all of my pursuits and inspiring me to follow my dreams. I am incredibly grateful to my parents, who supported me emotionally and financially. I always knew that you believed in me and wanted the best for me. Thank you for teaching me that my job in life was to learn, be happy, and know and understand myself; only then could I know and understand others.

Abstract

Dimensionality reduction (DR) is a data transformation process that provides a low-dimensional (attribute or variable) representation of high-dimensional data sets. The main goals of DR are noise reduction, storage space reduction, data visualization, efficient data processing, and the concentration of important information in fewer variables than the original set. A visual performance measure in DM is topology preservation. Quality curves R_{NX}, proposed by Lee and Verleysen, evaluate performance generating a graphical representation of topology preservation. Nowadays, there is a tool for topology conservation evaluation of DM algorithms, also developed by Lee and Verleysen (2009). To the best of our knowledge, such a tool is available only in MatLab. Therefore, a deployment challenge arises since MATLAB may be limited in portability and hardly used over different technologies such as frameworks for dimensionality reduction programmed in other programming languages. In this work, we provide an implementation in the Python programming language of a software evaluation module of the curve R_{NX}, a versatile and package-driven coding tool that enables its use in multiple technologies.

Keywords: Data topology, dimensionality redution, Python, R_{NX} curve.

Resumen

La reducción de dimensionalidad (DR) es un proceso de transformación de datos que proporciona una representación de baja dimensión (atributos o variables) de conjuntos de datos de alta dimensión. Los principales objetivos de la recuperación ante desastres son la reducción de ruido, la reducción del espacio de almacenamiento, la visualización de datos, el procesamiento eficiente de datos y la concentración de información importante en menos variables que el conjunto original. Una medida de rendimiento visual en DM es la preservación de la topología. Las curvas de calidad R_{NX}, propuestas por Lee y Verleysen, evalúan el rendimiento generando una representación gráfica de la preservación de la topología. En la actualidad, existe una herramienta para la evaluación de la conservación de la topología de los algoritmos de DM, también desarrollada por Lee y Verleysen (2009). A nuestro leal saber y entender, dicha herramienta solo está disponible en MatLab. Por lo tanto, surge un desafío de implementación ya que MATLAB puede tener una portabilidad limitada y apenas se usa en diferentes tecnologías, como marcos para la reducción de dimensionalidad programados en otros lenguajes de programación. En este trabajo, proporcionamos una implementación en el lenguaje de programación Python de un módulo de evaluación de software de la curva R_{NX}, una herramienta de codificación versátil y basada en paquetes que permite su uso en múltiples tecnologías.

Palabras Clave: Curva R_{NX}, reduccion de dimensiones, Python, topología de los datos.

Contents

Dedication **iii**

Acknowledgments **iv**

Abstract **v**

Resumen **vi**

Contents **vii**

List of figures **ix**

List of tables **x**

1 Introduction **1**
- 1.1 Problem statement 2
- 1.2 Objectives 2
  - 1.2.1 General objective 2
  - 1.2.2 Specific objectives 2
- 1.3 Contribution 2
- 1.4 Document organization 3

2 Theoretical background **4**
- 2.1 Introduction 4
- 2.2 Dimensionality reduction 4
  - 2.2.1 Distance preservation 5
  - 2.2.2 Topology 6
  - 2.2.3 Topology preservation 6
- 2.3 Dimensionality reduction convex techniques 6
  - 2.3.1 Full spectral techniques 6
  - 2.3.2 Sparse spectral techniques 10
  - 2.3.3 Non-convex techniques for dimensionality reduction 11
- 2.4 Kernel approximation techniques 13
  - 2.4.1 LLE kernel 13

2.4.2 Isomap kernel 14
2.4.3 LE kernel 14
2.5 Data representation quality 14
2.5.1 Spearman's rho siegel 15
2.5.2 Topological product 15
2.5.3 Konig's measure. 15
2.5.4 Trustworthiness and continuity 15
2.5.5 Local continuity meta-criterion 16
2.5.6 mean relative rank errors 16
2.5.7 Co-ranking matrix 17

3 Methodology 20
3.1 Method flowchart 20
3.1.1 Data matrix and dimensionality reduction 21
3.1.2 Distance matrix calculation 21
3.1.3 Ranking matrices calculation 21
3.1.4 Co-ranking matrix 21
3.1.5 Intrusions and extrusions 22
3.1.6 QNX 22
3.1.7 RNX 22
3.2 Experimental setup 23
3.2.1 Databases 24
3.2.2 Python implementation vs MAT-LAB implementation 24
3.2.3 Evaluating kernel approximation 24

4 Results and Discussion 27
4.1 Implementation comparison results 27
4.2 Evaluating kernel approximation results 28

5 Conclusions 37

Bibliography 38

Appendices 43
5.1 Appendix 1 44
5.1.1 Python implementation of RNX curve 44

List of figures

2.1 Dimensionality reduction techniques categories. 5
2.2 Diagram of an Autoencoder with three hidden layers. 12
2.3 Co-ranking Matrix. 19

3.1 Proposed R_{NX} curve flowchart. It contains the steps for obtaining R_{NX} curve from a high dimensional and low dimensional matrix. 20
3.2 Different types of intrusions and extrusions. 23
3.3 The four data sets used in the experiments are the spherical shell, the COIL-20 image bank, the Swiss Roll, and a random subset of the MNIST image bank. The COIL-20 and MNIST images are just vectorized before dimensionality reduction. In all four cases, a two-dimensional embedding is sought. 25
3.4 Methodology for comparing the Matlab implementation against Python implementation. 26
3.5 Methodology for comparing the Kernel methods KLE, KLLE, and KIsomap with their respective conventional method. 26

4.2 R_{NX} quality curve comparison between Matlab and Python using the MNIST database 29
4.3 R_{NX} quality curve comparison between Matlab and Python using the Sphere database. 30
4.4 R_{NX} quality curve comparison between Matlab and Python using the Swiss Roll database. 31
4.5 R_{NX} curve comparison from each DR spectral method and their respective kernel approximation, applied to Coil-20 database 33
4.6 R_{NX} curve comparison from each DR spectral method and their respective kernel approximation, applied to MNIST database. 34
4.7 R_{NX} curve comparison from each DR spectral method and their respective kernel approximation, applied to Sphere database. 35
4.8 R_{NX} curve comparison from each DR spectral method and their respective kernel approximation, applied to Swiss Roll database. 36

List of tables

2.1 Kernel functions in $\mathbb{R}^D$ 14

4.1 AUC comparison from the Matlab and Python implementation, and their corresponding R_{NX} mean squared error. 32

Chapter 1

Introduction

The amount of data produced in real-time has exploded at an unknown rate. According to IDC's "Data Age 2025" whitepaper, sponsored by Seagate, the sum of the world's data will grow to 175 zettabytes by 2025. Thus, researchers have developed new improvement techniques to extract and represent the valuable information in data [1]. These techniques fall in the field called data analytics, which can obtain hidden patterns in the data to make decisions accordingly or for data representation.

Among the fields of data mining and pattern recognition, large amounts of data are handled, and high-dimensional data. The latter refers by dimension to the number of characteristics or variables in the input data [2]. This much data is prone to errors and, most of the time has much redundancy. Moreover, to process such higher amounts of data requires a lot of computer power, and it is sought to be the most efficient possible. Because of this, it is mandatory to perform a preprocessing of the data. It has various objectives: noise reduction, storage reduction, data visualization, efficient data processing, and data compression [3, 4].

Dimensionality reduction represents high dimensional data into a lower dimension, abstracting the most critical details from the data and cutting redundancy [5]. Nowadays, there are many methods to achieve it; among these are spectral methods, divergence methods, heuristic methods, deep learning methods, neural networks, among others [6]. All these methods are designed to generate embedded low-dimensional spaces trying to conserve the original data's topology. Only spectral methods are taken into account in this work because they also count with a kernel approximation counterpart, and also they have been broadly used in many applications [5, 1]. Consequently, because of the high quantity of DR methods, the questioning of their quality assessment and comparison appears. A way to measure the quality of this dimensionality reduction is topology preservation. It means that the data will keep its spatial relations after the embedding in a lower dimension. This work implements the R_{NX} curve proposed by Lee and Verleysen [7]. This curve evaluates the performance generating a graphical representation of preservation of the local and global topologies. A tool for creating this curve exists in Matlab, but this makes it inconvenient to use when there is a pipeline of DR methods in other programming languages that are more portable. Python has become one of the more used programming languages by data scientists in the last years. For this motive, the necessity of such a python tool, which is not only oriented to mathematicians or prototyping as Matlab is, arises [8].

1.1 Problem statement

Dimensionality reduction (DR) is a data transformation process that provides a low-dimensional representation (attribute or variable) of high-dimensional data sets. In the last few years, many new nonlinear DR methods have been proposed. Moreover, the interrogation about their quality assessment and comparison remains open. A visual performance measure in DR is topology preservation. Quality curves R_{NX}, proposed by Lee and Verleysen, evaluate performance generating a graphical representation of topology preservation [7]. Nowadays, there is a tool developed by Lee y Verleysen (2009) that evaluates the topology preservation of DR algorithms using the R_{NX} quality curves. However, this tool is only programmed for scientific purposes in MATLAB, and it is not open source, limiting the implementation of this algorithm in other technologies.

1.2 Objectives

1.2.1 General objective

To develop, in python, a module of curves R_{NX} to evaluate the performance of dimensionality reduction and data representation based on data topology conservation and able to work on different technologies.

1.2.2 Specific objectives

- Implementation of mathematical routines which allow one to measure topology conservation using the R_{NX} curve in dimentionality reduction methods.
- Development of a plot in order to show the quality of dimensionality reduction algorithms through the R_{NX} curve.
- Integrate the module as a PyPI package in order to contribute the data science community and the already developed modules.

1.3 Contribution

As a solution for this issue, in this work, we present a python module, called nxcurve, based on Lee y Verleysen R_{NX} curve whose purpose is to assess the quality of dimensionality reduction showing a plot of the curve and its area under it. Broadly, nxcurve works as follows: it receives five parameters, high dimensional data, low dimensional data, which was obtained by using any dimensionality reduction technique, number of neighbors used for reduction, an option “r” for telling the module we want the R_{NX} curve, and a finally a boolean variable which if true the module will show the R_{NX} graph and the contrary if it is false. The module will not only return the graph. Also, it will return the vector containing the values of the R_{NX} curve, its area under the curve, and its name. These values are returned in case the user wants to draw its curve or multiple curves. Therefore, the main goal of nxcurve is to perform a quality evaluation of the low-dimensionality representation of high-dimensional data. The results are promising and open the possibility of implementing this algorithm in other applications. In order to compare the Python implementation against the Matlab implementation, two experiments were created, and measures taken are explained in Section 3.2.

1.4 Document organization

This thesis is divided into five chapters as follows:

- Chapter 1 (Introduction) generally outlines the aspects of the work, the problem statement (1.1), the contribution made (1.3), and the general and specific objectives (1.2).
- Chapter 2 (Theoretical Background) explains the main idea of dimensionality reduction (2.2). It presents a taxonomy for DR methods and goes over a review of the former. It introduces the goal of performing quality assessment in DR techniques (2.5). It shows different quality assessment techniques, and the framework created by Lee and Verleysen [7], from which many methods can be obtained.
- Chapter 3 (Methodology and Experimental Setup) It goes through the steps to obtain the R_{NX} and their respective algorithm (3.1). It describes the databases and the DR methods used in the experiments, and the taken metrics (3.2).
- Chapter 4 (Results and Discussion) presents the results from the two experiments and compares the two algorithm implementations along with a comparison between Kernel approximations and conventional DR methods.
- Chapter 5 (Conclusion) draws the final remarks about the work.

Chapter 2

Theoretical background

2.1 Introduction

In recent decades, the use of dimension reduction techniques has increased because of the complexity of analyzing high-dimensional data [9]. Dimensionality reduction allows eliminating redundant data, noise, reduction of features to improve data processing, identification of the essential features, and data visualization [4]. The use of dimensionality reduction also implies a loss of quality, affecting the understanding and meaning of the data. On the other hand, every DR algorithm is different, leading to a different percentage of quality loss at the time of the reduction depending on the method.

2.2 Dimensionality reduction

Real-world data such as photographs or sound signals usually present high dimensionality. For it to be handled efficiently a dimensionality reduction is needed (DR). Dimensionality reduction refers to the remodeling of high-dimensional data into a lower dimension retaining the geometry of the data as much as possible. The lower dimensional representation should have a minimum number of parameters to fulfill the observed properties of the data, this is called intrinsic dimensionality [10]. Mathematically, the objective of dimensionality reduction is to embed a data matrix

$$\mathbf{X} = [x_i]_{1 \leq i \leq N} : x_i \in \mathbb{R}^D, \tag{2.1}$$

consisting of n datavectors $\mathbf{x}_i$ with dimensionality D into a new dataset $\mathbf{Y}$ with dimensionality d.

$$\mathbf{Y} = [y_i]_{1 \leq i \leq N} : y_i \in \mathbb{R}^d, \text{ where } d < D. \tag{2.2}$$

Dimensionality reduction can be achieved by:

- Feature Elimination: Some features of the high dimensional data are eliminated to get the low dimensional representation [11]. [**?**].
- Feature selection: Here, statistical tests are applied to the features, and then they are ranked according to their importance. Finally, a subset of features is selected.

The disadvantage is the information loss and its stability as different statistical tests can throw different importance scores to the features [12].

- Feature extraction: New independent features are created from old dependent features. These techniques can be divided into linear and non-linear, and their disadvantage depends on the mathematical method applied [13].

Figure 2.1 shows a classification of dimensionality reduction techniques where convex and non-convex techniques are two major groups. The optimization is different for both groups; convex techniques focus on the optimization an objective function with no local optima. On the other hand, non-convex techniques seek the optimization of functions with a local optima. The remainder of techniques are discussed in the following sections.

2.2.1 Distance preservation

Dimensionality reduction uses the criterion of distance preservation. This ensures that the data in lower dimensionality representation preserves its geometrical properties. However, in nonlinear cases, distance is not entirely preserved because we are dealing with manifolds. A manifold is a generalization of the notion of a curve surface which is closely modeled on a Euclidean space [14].

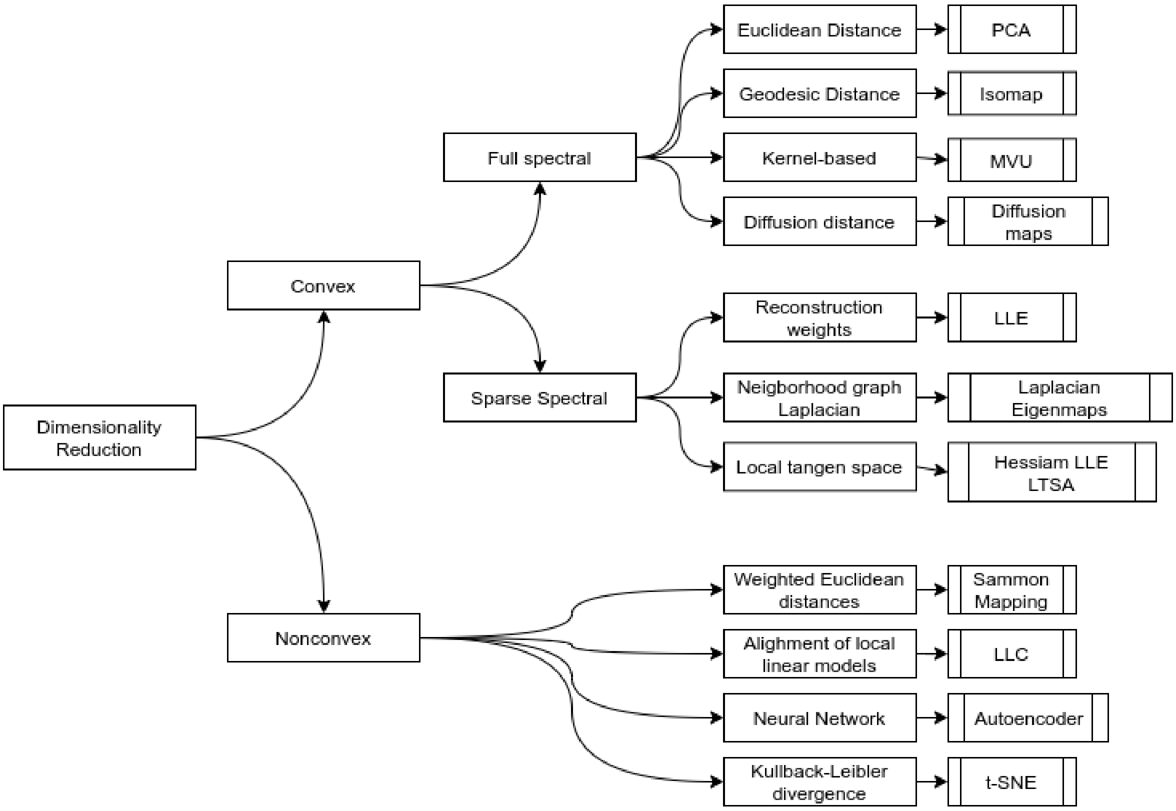


Figure 2.1: Dimensionality reduction techniques categories.

2.2.2 Topology

From a geometrical perspective, the support of the joint distribution of two or more dependent variables does not span the whole space. This dependence produces some structure in the distribution, in the form of an object in space. The sphere and the Swiss Roll in figure 3.3 represent these objects. Furthermore, as already mentioned, Dimensionality reduction seeks to give a new representation of these objects while preserving their structure [15].

Topology in mathematics studies the preserved properties in objects between deformations, twistings, and stretching. Tearing is not allowed because to guarantee the structure preservation or connectivity of the object. As an example, we can say that topologically, a circle is equivalent to an ellipsoid. Topology encapsulates the connectivity of objects ignoring the detail form. If two objects hold the same topological properties, they are said to be homeomorphic.

These objects are formally called topological spaces. Geometrically, a topological space is defined using neighborhoods and Haussdorf's axioms. The neighborhood of a point $\mathbf{y} \in \mathbb{R}^D$ is a set of points inside a D-dimensional hollow space with radius $\epsilon > 0$ centered in $\mathbf{y}$. Then, a manifold M is defined as a locally Euclidean topological space. Generally, a manifold is an object which is nearly "flat" on small scales. The representation of a topological object in a specific space $\mathbb{R}^D$ is called an embedding [16].

2.2.3 Topology preservation

Other DR methods, instead of preserving the distance look for preserving the topology. Topology preservation techniques are also called local preservation approaches. The difference from distance preservation techniques is that topological preservation does not constraint distance conditions leading to better flexibility of subregions that in many cases require to be locally stretched or shrunk to construct a good embedding.

2.3 Dimensionality reduction convex techniques

Convex techniques pursue the optimization of an objective function without local optima, which means that the solution space is convex [17]. A great quantity of dimensionality reduction techniques belongs to this category. The form of the objective function to be optimized with the solution of an eigenproblem is $\phi(\mathbf{Y}) = \frac{\mathbf{Y}^T\mathbf{A}\mathbf{Y}}{\mathbf{Y}^T\mathbf{B}\mathbf{Y}}$ (a (generalized) Rayleigh quotient). It is well known that a function of this form can be optimized by solving a generalized eigenproblem. Convex dimensionality reduction subdivides into techniques that perform a full matrix eigendecomposition and techniques that perform sparse matrix eigendecomposition.

2.3.1 Full spectral techniques

Full spectral dimensionality reduction techniques carry out a full matrix eigendecomposition (also called spectral decomposition) which gets the covariances between dimensions or the similarities between data points. These spectral techniques also allow dimensionality reduction in a feature space that is constructed through a kernel function.Six techniques are discussed in this subsection.

PCA

Principal Component Analysis (PCA) is the most popular linear dimensionality reduction technique. Many fields have used it since it first appeared, such as biology [18], psychometry, geophysics [19], medicine [20], and statistical processes. PCA's objective is to extract relevant information from the linear combination of the original data's characteristics. It embeds data into a linear subspace of lower dimensionality, which describes as much of the original data variance as possible. PCA makes this reduction by maximizing the variance of a linear basis of lower dimension [21, 22]. Mathematically, PCA pursues the maximization of the cost function given by $\operatorname{trace}(\mathbf{M}^T \operatorname{cov}(\mathbf{X})\mathbf{M})$, aiming to find a mapping $\mathbf{M}$. This linear mapping comprises the d principal eigenvectors of the sample covariance matrix $\operatorname{cov}(\mathbf{X})$, which are also called principal components. Consequently, PCA solves the equation:

$$\operatorname{cov}(\mathbf{X})\mathbf{M} = \lambda \mathbf{M}, \tag{2.3}$$

Multidimensional scaling (MDS)

Classical scaling is identical to the most used technique: PCA [23]. The main difference is that when using PCA, the maximum variance is preserved, whereas when using MDS, maximum distance is preserved between pairs of low dimensional data points [24]. MDS can be metric (classical) or non-metric. Both use a matrix distance to characterize the points according to their similarity or dissimilarity.

The input of classical scaling is a pairwise Euclidean distance matrix $\mathbf{D}$. This matrix contains the euclidean distances between the points in the high dimensional representation. Classical scaling finds the mapping M minimizing the cost function:

$$\phi(\mathbf{Y}) = \sum_{ij} \left(d_{ij}^2 - \left\| \mathbf{y}_i - \mathbf{y}_j \right\|^2 \right), \tag{2.4}$$

where $\left\| \mathbf{y}_i - \mathbf{y}_j \right\|^2$ is the Euclidean distance between the points $\mathbf{y}_i$ and $\mathbf{y}_j$, $\mathbf{y}_i$ in the low dimensional representation and it is restricted to be $\mathbf{x}_i\mathbf{M}$, and $\left\| \mathbf{m}_j \right\|^2 = 1 \ \forall j$. In order to minimize the cost function the the eigendecomposition of the Gram matrix $\mathbf{K} = \mathbf{X}\mathbf{X}^T$ is needed. The entries of this matrix can be obtained by computing:

$$k_{ij} = -\frac{1}{2} \left(d_{ij}^2 - \frac{1}{n} \sum_l d_{il}^2 - \frac{1}{n} \sum_l d_{jl}^2 + \frac{1}{n^2} \sum_{lm} d_{lm}^2 \right), \tag{2.5}$$

this equation is in charge of centering the matrix containing pairwise distances. Next, the cost function minimization can be obtained with the multiplication between the principal eigenvectors obtained from the distance matrix and the square root of their corresponding eigenvalues. Classical scaling is flexible, and it can accept as input scalar products as well as Euclidean distances. It also presents high memory usage due to the storage of the Gram matrix $N \times N$. On the other hand, PCA does not present this inconvenience as the covariance matrix is $D \times D$ [16]. From the equation are obtained d higher eigenvalues. Then, In order to obtain the low dimensional representation, the linear basis: $\mathbf{Y} = \mathbf{XM}$ is mapped.

Isomap

CMDS has proven to be useful in many applications. However, because it retains pairwise distances, it does not consider the distribution of the neighboring points, which means that if data lie in a curvilinear manifold, such as a sphere or swiss roll [25], MDS would consider two data points as if they were near, whereas their distance over the manifold is much larger than the typical distance between points. On the other hand, Isomap is a method that attempts the preservation of pairwise geodesic distances between high dimensional data points, solving the previously mentioned issue. Geodesic distance refers to the measured distance between two points over the surface of the manifold. The geodesic distance between two points can be represented as $S_{ij} = \phi\left(\mathbf{x}_i, \mathbf{x}_j\right)$, with $\phi(\cdot)$ being the geodesic distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. For the calculation of the distances, a graph of neighborhoods G is constructed. In this graph, every datapoint $\mathbf{x}_i$ is connected with its k neighbors $\mathbf{x}_{ij}$ in the high dimensional representation $\mathbf{X}$. Dijkstra's algorithm can be used to get the geodesic distance between two data points. With this precious process, a Gram matrix containing geodesic distances is obtained. Next, Classical scaling is then performed on this matrix to obtain the low-dimensional representation $\mathbf{Y}$.

Isomap also presents some issues such that it is topologically unstable [26]. It can construct faulty connections in the neighborhood graph G. Also, if the manifold is not convex this method is prone to fail. In spite of these issues, Isomap was properly used in tasks such as intrusion detection and data visualization [27, 28].

KPCA

Kernel PCA deals with linearly inseparable data projected onto a higher dimensional space where it becomes separable [29, 30]. It computes the principal eigenvectors from a kernel matrix obtained using a nonlinear mapping function instead, called kernel function, of the covariance matrix. Applying PCA in the kernel space has the advantage of constructing nonlinear mappings. The items in the kernel matrix K computed from the data points $\mathbf{x}_i$ are defined by:

$$k_{ij} = \kappa\left(\mathbf{x}_i, \mathbf{x}_j\right), \tag{2.6}$$

where κ is a kernel function that gives us a positive-semidefinite kernel $\mathbf{K}$. Next, the kernel matrix $\mathbf{K}$ is centered using

$$k_{ij} = -\frac{1}{2}\left(k_{ij} - \frac{1}{n}\sum_{l} k_{il} - \frac{1}{n}\sum_{l} k_{jl} + \frac{1}{n^2}\sum_{lm} k_{lm}\right), \tag{2.7}$$

if we look at the traditional PCA, the centering is performed by subtracting the mean of the columns of features. In a similar way, Kernel PCA center the data by subtracting the mean of the columns of the kernel function. As a next step, from the kernel matrix d eigenvectors, are obtained [31] and the eigenvectors of the covariance matrix $\mathbf{a}_i$ is calculated because of the relation:

$$\mathbf{a}_i = \frac{1}{\sqrt{\lambda_i}}\mathbf{v}_i, \tag{2.8}$$

as a final step, the original data is projected onto the latter calculated eigenvectors. The result of the projection of the low dimensional data representation is acquired by

$$\mathbf{y}_i = \left\{ \sum_{j=1}^{n} a_1^{(j)} \kappa \left(\mathbf{x}_j, \mathbf{x}_i \right), \ldots, \sum_{j=1}^{n} a_d^{(j)} \kappa \left(\mathbf{x}_j, \mathbf{x}_i \right) \right\}, \tag{2.9}$$

in this equation, $a_1^{(j)}$ represents the jth value in the vector a_1 and κ, the kernel function used for obtaining the kernel matrix.

MVU/Semidefinite embedding

As mentioned before, Kernel PCA performs PCA in a space defined by a kernel function κ. MVU is a technique that infers the kernel matrix to be used by defining a neighborhood graph on the data and retaining the pairwise distances in the resulting graph [6]. The principal difference from Isomap is that the goal of MVU is to unfold the data manifold. A manifold is a topological space that resembles Euclidean space near each point locally. It achieves this by maximizing the Euclidean distances between data points under the condition that the distances in the neighborhood don't have changes. MVU first constructs the graph of nearest neighbors G where each point is connected to its k nearest neighbors $\mathbf{x}_{ij}$.The constraint that the distance inside the graph G are conserved applied in the maximization of the sum of squares of the distances from the high dimensional data points. Mathematically:

$$\begin{aligned} &\text{Maximize } \sum_{ij} \left\| \mathbf{y}_i - \mathbf{y}_j \right\|^2 \text{ subject to (1), with:} \\ &(1)\ \left\| \mathbf{y}_i - \mathbf{y}_j \right\|^2 = \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2 \text{ for } \forall (i,j) \in G. \end{aligned} \tag{2.10}$$

This optimization can be seen as a semidefinite programming problem by defining the kernel matrix $\mathbf{K}$ as the outer product of the low dimensional data representation $\mathbf{Y}$ [32]. Next, the kernel matrix is determined, maximizing the trace of $(\mathbf{K})$ subject to the following equations.

1. $k_{\mathrm{zi}} + k_{jj} - 2k_{i,j} = \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2$ for $\forall (i,j) \in G$,
2. $\sum_{ij} k_{ij} = 0$,
3. K $\succeq 0$, (Semidefinite K)

The solution of this problem is the kernel matrix $\mathbf{K}$ which is then passed to Kernel PCA to obtain the low dimensional representation $\mathbf{Y}$ of the data

Diffusion maps

Diffusion maps is a nonlinear spectral method that uses Markov chains to perform a random walk through the data. Once the finite walk is finished, a measure of proximity between data points is obtained [33]. The diffusion distance is defined using this measure, and in the low dimensional embedding, the diffusion distances are preserved [34]. This method first constructs a graph of the data where the edges are computed using the Gaussian kernel function. Then W is normalized such that its rows add up to 1, forming matrix P with entries:

$$p_{ij}^{(1)} = \frac{w_{ij}}{\sum_k w_{ik}}, \tag{2.11}$$

$p_{ij}^{(1)}$ is considered a Markov matrix that represents the probability of change from one data point to another in a single timestep. Later, the probability matrix for t steps $P^{(t)}$ is given by $(P^{(1)})^t$. Then the low dimensional representation $\mathbf{Y}$ is given by obtaining d nontrivial principal eigenvectors of the eigenproblem:

$$\mathbf{P}(t)\mathbf{v} = \lambda \mathbf{v}. \tag{2.12}$$

2.3.2 Sparse spectral techniques

Full Spectral Techniques perform a low-dimensional representation of the high dimensional data to obtain a low dimensional representation using a full matrix eigendecomposition. On the other hand, the following four techniques discussed in this section solve a sparse generalized eigenproblem, and all of them focus on keeping the local structure of the data.

LLE

Locally Linear Embedding, as well as Isomap and MVU, constructs a graph of the data points. The main difference is that LLE seeks to preserve solely local properties of the data allowing successful embedding of non-convex manifolds [35]. LLE performs adjusted mapping, which means that it preserves local angles (local scalar product). The local properties of a point $\mathbf{x}_i$ in the manifold are expressed as a linear combination $\mathbf{w}_i$ of its k nearest neighbors $\mathbf{x}_{ij}$. If the geometry of the manifold is preserved in the low dimensional representation, the weight that reconstructs a point in the high dimensional representation also reconstruct a point from its neighbors in the low dimensional one. The d-dimensional representation can be found minimizing the cost function.

$$\phi(\mathbf{Y}) = \sum_i \left\| \mathbf{y}_i - \sum_{j=1}^{k} w_{ij} \mathbf{y}_{i_j} \right\|^2 \text{subject to} \left\| \mathbf{y}^{(k)} \right\|^2 = 1 \ \forall k, \tag{2.13}$$

this minimization can be found by solving the eigenproblem of the inner-product $(\mathbf{X} - \mathbf{W})^T(\mathbf{X} - \mathbf{W})$, where W is a sparse $n \times n$ matrix [35]. if i and j are not connected in the constructed graph the entries of $\mathbf{W}$ are put to zero ; if they do are connected, the value is set to their corresponding weight.

Laplacian eigenmaps

Laplacian Eigenmaps obtain a low-dimensional representation of high-dimensional data preserving the local properties of the manifold. In this method, pairwise distances between the neighbors are the foundation of the local properties [36]. Then, to find the low dimensional representation, the distances of a data point and its k nearest neighbors are minimized. The minimization can be computed using spectral graph theory fundamentals and a Laplacian graph's notions. Also, it is defined as an eigenproblem. First, a graph G is constructed where every data point in the high dimensional representation is connected to its k nearest neighbors. Then an adjacency matrix W which entries are the weights of each connection computed using the Gaussian kernel [37]. The cost function is given by

$$\phi(\mathbf{Y}) = \sum_{ij} \left\|\mathbf{y}_i - \mathbf{y}_j\right\|^2 w_{ij}, \tag{2.14}$$

the minimization can be seen as an eigenproblem by the calculation of the degree diagonal matrix M which contain the sum rows of W. $L = M - W$. It can be shown that

$$\phi(\mathbf{Y}) = \sum_{ij} \left\|\mathbf{y}_i - \mathbf{y}_j\right\|^2 w_{ij} = 2\mathbf{Y}^T\mathbf{L}\mathbf{Y}, \tag{2.15}$$

next, minimizing $\phi(\mathbf{Y})$ is equivalent to minimizing $\mathbf{Y}^T\mathbf{L}\mathbf{Y}$. The low dimensional representation Y is found by solving the eigenproblem for the d smallest nonzero eigenvalues and eigenvectors from the following equation.

$$\mathbf{L}\mathbf{v} = \lambda\mathbf{M}\mathbf{v}, \tag{2.16}$$

LTSA

Local Tangent Space Analysis (LTSA) describes the local properties of high dimensional data using each data point's tangent space [38]. This method assumes local linearity in the manifold, leading to the existence of a linear mapping between a high-dimensional data point and its tangent space and the existence of a linear mapping from the corresponding low dimensional data point to the same local tangent space. This method starts by computing the bases for the local tangent spaces in all data points resulting in a matrix M, a mapping from the neighborhoods to the local tangent spaces. As mentioned before, a mapping between the local tangent space L to the low dimensional representation exists. Then the following minimization is performed.

$$\mathcal{H}_{lm} = \sum_i \sum_j \left((\mathbf{H}_i)_{jl} \times (\mathbf{H}_i)_{jm}\right) \tag{2.17}$$

2.3.3 Non-convex techniques for dimensionality reduction

The last section reviewed techniques that obtain a low-dimensional representation from a high-dimensional one by performing the optimization of a convex objective function using eigendecomposition. On the other hand, in this section we reviewed techniques that used non-convex functions.

Sammon Mapping

Sammon mapping starts from classical scaling cost function (Equation (2.4)). It modifies this cost function by calculating the contribution of each (i,j) to the cost function using the inverse of their pairwise distance in the high dimensional space [39]. Hence, Local structure (small pairwise distances) is conserved better than in CMDS . The cost function for Sammon is given by:

$$\phi(\mathbf{Y}) = \frac{1}{\sum_{ij} d_{ij}} \sum_{i \neq j} \frac{\left(d_{ij} - \left\|\mathbf{y}_i - \mathbf{y}_j\right\|\right)^2}{d_{ij}}, \tag{2.18}$$

in this equation, d_{ij} constitute the Euclidean distances in the high dimensional manifold. The minimization of this cost function is achieved through a pseudo Newton method [40].

Multilayer autoencoders

Multilayer autoencoders are symmetrical neural networks with an odd number of hidden layers [41]. Commonly weights are shared between the bottom and upper layers. The middle hidden layer consists of d nodes and the input and output layer count with D nodes. Figure 2.2 shows an schema of an autoencoder. The neural network aims to minimize the mean square error between the network's input and output layers, which ideally should be equal. The middle hidden layer results in a d-dimensional representation $\mathbf{Y}$ of the high dimensional data with structure preservation (small pairwise distances preservation). If a nonlinear mapping is wanted, sigmoid functions are used among the neurons except for the middle hidden layer, where a linear activation function is used.

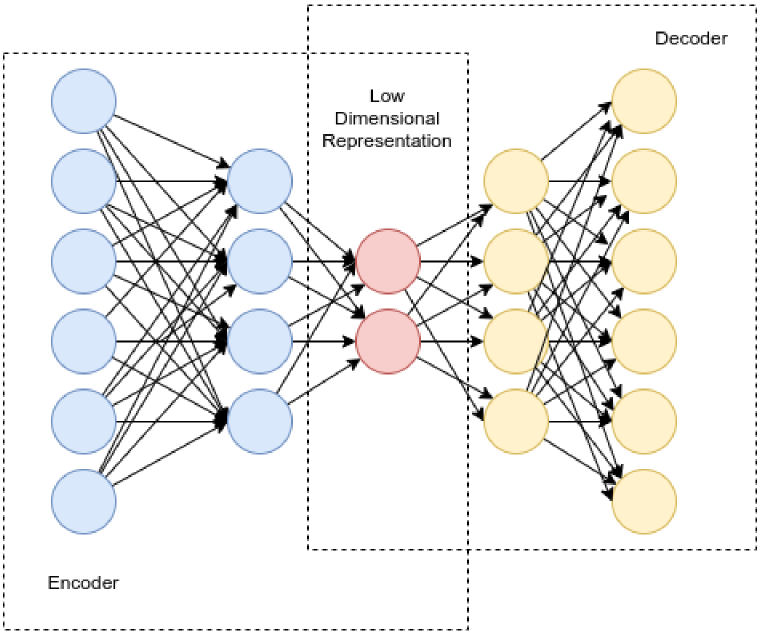


Figure 2.2: Diagram of an Autoencoder with three hidden layers.

t-SNE

t-distributed Stochastic Neighbor Embedding is a non-linear dimensionality reduction technique. It begins by constructing a probability distribution on pairs in higher dimensions such that similar objects are assigned a higher probability, and dissimilar objects are assigned a lower probability. It minimizes Kullback-Leibler divergence D between two distributions, P and Q, where Q is a t-distribution [1]. Kullback-Leibler divergence is a measure of the difference between the probability distributions P and Q. The cost function has the form

$$E_{\text{t-SNE}}(\boldsymbol{X}) = \sum_{n=1}^{N} \mathrm{D}_{\mathrm{KL}}\left(\boldsymbol{P}_n \| Q_n\right) = \sum_{n,m=1}^{N} p_{nm} \log \frac{p_{nm}}{q_{nm}}. \tag{2.19}$$

2.4 Kernel approximation techniques

In Section 2.3.1, Kernel PCA was mentioned. This method uses a kernel matrix rather a covariance matrix fot he computing of the principal eigenvectors. The use of a kernel allows for the generalization of the methods and solving problems that standard PCA and CMDS present [42]. Unfortunately, it is unclear how the kernel function k should be selected. MVU (section 2.3.1) tries to resolve this problem by learning a kernel matrix. Another approach is to pick a kernel function to a specific problem to approximate the existing methods. To pick a kernel function, it is necessary to define some restrictions [43].

1. The kernel matrix k must be positive definite.
2. The kernel matrix must contain sets of linear constraints on its elements.
3. The mappings between inputs and features are restricted from fully general nonlinear transformations to the particular class of isometries.

Let us call a nonlinear function ϕ so that the mapping of sample x can be written as $\mathbf{x} \rightarrow \phi(\mathbf{x})$ which is called kernel function [44]. This function calculates the dot product of the images of the samples x under ϕ. In other words, ϕ maps the features of the original data into a larger k-dimensional feature space creating a nonlinear combination of the original features.

$$\kappa\left(\mathbf{x_i}, \mathbf{x_j}\right) = \phi\left(\mathbf{x_i}\right) \phi\left(\mathbf{x_j}\right)^T, \tag{2.20}$$

this way, kernel functions allow a better representation of the high dimensional data features, and proximity measures can be expressed using these functions [45]. Among these measures, there are positive symmetric properties like the distances Euclidean, Minskowki, Hamming, Mahalanobis. Also, there are binary similarity measures such as cosine similarity, the Jaccard coefficient, and the Pearson coefficient. Table 2.1 shows some kernel functions which are reviewed in [45].

2.4.1 LLE kernel

Kernel LLE can be approximated using quadratic forms with a matrix W which reconstructs observed data and contains linear coefficients that sum to one. The kernel is represented by [42].

$$\boldsymbol{K}_{LLE} = \lambda_{\max} \boldsymbol{I}_N - \boldsymbol{M}, \tag{2.21}$$

in this equation, M is an $N \times X$ with $\lambda_{\max}$ as its maximum eigenvalue and $M = (I_N - W)$.

Kernel Functions	
Function	Notation
Lineal	$\left\langle \boldsymbol{y}_i, \boldsymbol{y}_j \right\rangle$
Polinomial	$\left\langle \boldsymbol{y}_i, \boldsymbol{y}_j \right\rangle^D$
Quadratic	$1 - \frac{\left\lVert \boldsymbol{y}_i - \boldsymbol{y}_j \right\rVert^2}{\left\lVert \boldsymbol{y}_i - \boldsymbol{y}_j \right\rVert^2 + \sigma}, \sigma \in \mathbb{R}^+$
Exponential	$\exp\left(-\frac{\left\lVert y_i - \boldsymbol{y}_i \right\rVert}{2\sigma^2} \right), \sigma \in \mathbb{R}^+$
Gaussian	$\exp\left(-\frac{\left\lVert y_i - y_j \right\rVert^2}{2\sigma^2} \right), \sigma \in \mathbb{R}^+$
RBF	$\exp\left(-\sum_{i=1}^{D} \gamma_i \left(\boldsymbol{y}_i - \boldsymbol{y}'_i \right)^\beta \right), \gamma_i > 0, \beta \in (0, 2]$
ANOVA	$\left(\exp\left\{ -\sum_{i=1}^{D} \gamma_i \left(\boldsymbol{y}_i - \boldsymbol{y}'_i \right)^2 \right\} \right)^m, \gamma_i > 0, m \in \mathbb{N}$
hyperbolic tangent kernel	$\tanh\left(a \left\langle \boldsymbol{y}_i, \boldsymbol{y}_j \right\rangle + b \right), a > 0, b > 0$
Camberra	$1 - \frac{1}{D} \sum_{i=1}^{D} \gamma_i \frac{\left\lvert \boldsymbol{y}_i - \boldsymbol{y}_j \right\rvert}{\boldsymbol{y}_i + \boldsymbol{y}_j}, \gamma_i \in (0, 1]$
Euclidean	$\frac{1}{D} \sum_{i=1}^{D} \max\left\{ 0, 1 - \frac{\left\lvert \boldsymbol{y}_i - \boldsymbol{y}_j \right\rvert}{\gamma_i} \right\}, \gamma_i > 0$

Table 2.1: Kernel functions in $\mathbb{R}^D$.

2.4.2 Isomap kernel

In Isomap, the geodesic distance can be represented as a kernel [46]. The kernel has the form

$$K = -\frac{1}{2} H D^2 H, \tag{2.22}$$

where D^2 is the geodesic distance matrix and H the centering matrix, which is given by

$$H = I_n - \frac{1}{N} e_N e_N^T, \quad \text{where } e_N = [1 \dots 1]^T \in \mathbb{R}^N, \tag{2.23}$$

2.4.3 LE kernel

the kernel representation for LE is the pseudoinverse of the Laplacian graph $L = M - w$ as seen in section 2.3.2.

2.5 Data representation quality

The existence of many dimensionality reduction algorithms opens the question of their quality assessment. Most of these mathods evaluate local neighborhood preservation and the geometric structure preserved after the reduction. In this section, we present some of the most used approaches.

2.5.1 Spearman's rho siegel

Spearman's Rho Siegel was one of the first topology preservation estimation measures after a dimensionality reduction technique called Spearman's rho [47]. It assesses how well the dimensionality reduction preserves the order of the pairwise distances in the high dimensional space. For this measure, the following equation is used

$$S_R = 1 - \frac{6\sum_{i=1}^{T}(z(i) - \widehat{z}(i))^2}{T^3 - T}, \tag{2.24}$$

where $z(i)$ are the pairwise distances of the high dimensional data space in ascending order. T represents the total number of distances to be compared. This measure vary between the values [-1, 1] where 1 means perfect preservation.

2.5.2 Topological product

The following technique aims to get the quality of the low dimensional representation of self-organized maps [48]. It is a measure of distance preservation among local neighborhoods. This method is based on two distances Q_1, the distance between point x_i to the j_{th} nearest neighbor in the high dimensional data and Q_1 and Q_2 with the analogous information in the low dimensional representation. These two measures are combined giving.

$$T_{Pr} = \frac{1}{n(n-1)}\sum_{g=1}^{n}\sum_{f=1}^{n-1}\log\left(\Pi_{p=1}^{f}Q_1(g,p)Q_2(g,p)\right)^{\frac{1}{2f}} \tag{2.25}$$

where if T_{Pr} it is equal to zero means a perfectly order preserving map.

2.5.3 Konig's measure.

It is a measure that focuses on maps formed using self-organizing neural networks. Konig K_M measures local preservation in self-organizing neural networks [49]. It analyses the rank order in the high and low dimensional spaces, and it is calculated with

$$K_M = \frac{1}{3k_1 n}\sum_{i=1}^{n}\sum_{j=1}^{k_1} KM_{ij}, \tag{2.26}$$

where, if the result of this equation is one means perfect preservation of small distances. The topology is represented with KM_{ij} and k_1 is the neighborhood value.

2.5.4 Trustworthiness and continuity

Trustworthiness measures the degree of data points initially far entering a neighborhood, and continuity measures the degree of points that are initially in a neighborhood were push away from it [50]. This method (T&C)exchanges indices of neighbors in high dimensional data and low dimensional data. The equations for trustworthiness and continuity are given by the formulas.

$$\begin{aligned} M_T &= 1 - \tfrac{2}{nk(2n-3k-1)}\sum_{i=1}^{n}\sum_{j\in U_k(i)\notin V_k(i)}(r(i,j)-k), \\ M_C &= 1 - \tfrac{2}{nk(2n-3k-1)}\sum_{i=1}^{n}\sum_{j\in V_k(i)\notin U_k(i)}(\widehat{r}(i,j)-k). \end{aligned} \tag{2.27}$$

Here, k represents the size of the neighborhood, $r(i,j)$ the high dimensional data ranks, and $\widehat{r}$ the lower-dimensional data ranks. Then M_T and M_C are combined in the equation

$$Q_T = \alpha M_T + (1-\alpha)M_C, \tag{2.28}$$

the result is in the interval $(0,1)$ and higher values mean good preservation of trustworthiness and continuity.

2.5.5 Local continuity meta-criterion

Local continuity meta-criterion also refered as LCMC checks the degree of overlap between the neighboring sets of a data sample and their corresponding low dimensional representation [51]. The equation for this method is

$$\mathrm{U}_{LC} = 1 - \frac{1}{nk}\sum_{i=1}^{n}\left|\Psi_k^{\chi}(i)\bigcap\Psi_k^{y}(i)\right| - \frac{k^2}{n-1}, \tag{2.29}$$

where k is the number of neighbors, $\Psi_k^{\chi}(i)$ is the index set of k points in the high dimension and $\Psi_k^{y}(i)$ the index st of points in the lower dimension. The resultant values of U_{LC} are in the interval [0,1]. The values close to 1 mean high neighborhood overlap, and values close to 0 low means low neighborhood overlap.

2.5.6 mean relative rank errors

Mean Relative Rank Errors (MRRE) is a quality assessment method based on ranks of pairwise Euclidean distances among local neighborhoods, developed by Lee and Verleysen [7, 16]. The present method is similar to Truswortines and Continuity method, and it also has two components defined as

$$W_T = 1 - \frac{1}{H_k}\sum_{i=1}^{n}\sum_{j\in U_k(i)}\frac{|r(i,j)-\hat{r}(i,j)|}{r(i,j)}, \tag{2.30}$$

$$W_C = 1 - \frac{1}{H_k}\sum_{i=1}^{n}\sum_{j\in V_k(i)}\frac{|r(i,j)-\hat{r}(i,j)|}{\hat{r}(i,j)}, \tag{2.31}$$

as in other methods, k represents the size of the neighborhood. H_k is a normalizing factor given by equation 2.32 and MRRE is given by Q_M $n[0,1]$ (equation 2.33)where values near to 0 show small rank error in the lower dimensional representation

$$H_k = n\sum_{i=1}^{k}\frac{|n-2i+1|}{i}, \tag{2.32}$$

$$Q_M = \beta W_T + (1-\beta)W_C. \tag{2.33}$$

2.5.7 Co-ranking matrix

Given a high dimensional data $\mathbf{X}$ and its low dimensional representation $\mathbf{Y}$ calculate the dissimilarity matrix for both (D_X and D_Y respectively). The symbol δ_{ij} indicate the distance from x_i to x_j in the high-dimensional space and d_{ij} represents the distance from y_i to y_j in the low dimensional space. It is assumed that $\delta_{ij} = \delta_{ji}$ and $d_{ij} = d_{ji}$, however this conjecture does not always holds true. For example, when δ_{ij} and δ_{ji} come from different experimental measures. Also, there is no assumption as to the metrics associated with the high and low dimensional spaces that can differ.

Starting from distances ranks are computed. In the high-dimensional space, the rank of x_i relative to x_j is represented as

$$\rho_{ij} = |\delta_{ik} < \delta_{ij} \text{ or } (\delta_{ik} = \delta_{ij} \text{ and } k < j)|, \tag{2.34}$$

where $|.|$ stands for the cardinality of the set. The same way in the low-dimensional space, the rank of x_i respect to x_j is

$$r_{ij} = |d_{ik} < d_{ij} \text{ or } (d_{ik} = d_{ij} \text{ and } k < j)|, \tag{2.35}$$

consequently, reflexive ranks are zero $\rho_{ii} = r_{ii} = 0$ and ranks are unique, for example $\rho_{ij} \neq \rho_{ik}$ for $k \neq j$, even if $\delta_{ij} = \delta_{ik}$

Having the rank matrices we calculate the co-ranking matrix Q. Computing this matrix demand 2N sorting operations resulting on a time complexity of $O(N^2)$ with a typical sorting algorithm [7]. Errors of a DR mapping correspond to elements that are not in the diagonal of this matrix. A point j where $\rho_{ij} > r_{ij}$ represent a intrusion and $\rho_{ij} <$ is an extrusion. The co-ranking matrix is defined by:

$$\mathrm{Q} = [q_{kl}]_{1 \leqslant k,l \leqslant N-1} \qquad \text{with } q_{kl} =| \left\{ (i,j) : \rho_{ij} = k \text{ and } r_{ij} = l \right\} |, \tag{2.36}$$

the errors after the dimensional reduction process are contained in the non-diagonal entries of the co-ranking matrix Q. This matrix is a histogram of the combinations of the ranks. The co-ranking matrix can also be exhibited as a Shepard diagram [52], and with this viewpoint, it suggested that the essential criteria should concentrate in the entries of the upper triangle matrix and lower triangle matrix of the co-ranking matrix Q. Then, we define the rank errors as the difference $\rho_{ij} - r_{ij}$. An intrusion is defined as a set of points entering a neighborhood n_i^K erroneously with respect to the original neighborhood v_i^K and extrusion refers to sets of points leaving the neighborhood n_i^K erroneously [7]. As the focus is on K-ary neighborhoods, K-intrusion is defined as two events happening simultaneously: an extrusion with $r_{ij} < K$ and K-extrusion: an intrusion with $\rho_{ij} < K$. Furthermore, mild and hard K-intrusions are defined. The former refers when $r_{ij} < \rho_{ij} \leq K$ and the latter $r_{ij} \leq K < \rho_{ij}$. The co-ranking matrix is divided into four blocks separating the first K rows and columns to associate intrusions and extrusions. If $\mathbb{F}_K = \{1, \ldots, K\}$ and $\mathbb{L}_K = \{K+1, \ldots, N-1\}$ are defined, upper-left, upper-right, lower-left and lower-right index sets blocks are

$$\begin{aligned} \mathbb{UL}_K = \mathbb{F}_K \times \mathbb{F}_K \mathbb{UR}_K = \mathbb{F}_K \times \mathbb{L}_K, \\ \mathbb{LL}_K = \mathbb{L}_K \times \mathbb{F}_K \mathbb{LR}_K = \mathbb{L}_K \times \mathbb{L}_K. \end{aligned} \tag{2.37}$$

Again, The upper left block can be divided into its main diagonal and lowe and upper triangles:

$$\mathbb{D}_K = \{(i,i) : 1 \leqslant i \leqslant K\}, \tag{2.38}$$

$$\mathbb{LT}_K = \{(i,j) : 1 < i \leqslant K \text{ and } j < i\}, \tag{2.39}$$

$$\mathbb{UT}_K = \{(i,j) : 1 \leqslant i < K \text{ and } j > i\}, \tag{2.40}$$

taking into account the mentioned division, The lower and upper trapezes represent K-intrusions and K-extrusions, respectively. Hard K-intrusions and K-extrusions are found in the blocks $\mathbb{LL}_K$ and $\mathbb{UR}_K$, and mild K-intrusions and K-extrusions in $\mathbb{LT}_K$ and $\mathbb{UT}_K$. Previously mentioned quality measures based on ranks can be defined in terms of the co-ranking matrix. Then T&C from subsection 2.5.4 is redefined as

$$M_T = 1 - \frac{2}{G_K} \sum_{(k,l)\in\mathbb{I}_K} (k-K) q_{kl}, \tag{2.41}$$

$$M_C = 1 - \frac{2}{G_K} \sum_{(k,l)\in U\mathbb{R}_K} (l-K) q_{kl}, \tag{2.42}$$

the MRRE from subsection 2.5.6 which also rely on the same principle as T&C.

$$W_T = W_{\mathrm{N}}^{v,w}(K) = \frac{1}{C_K} \sum_{(k,l)\in\mathbb{LT}_K\cup\mathbb{LL}_K} \frac{(k-l)^v}{k^w} q_{kl}, \tag{2.43}$$

$$W_C = W_{\mathrm{X}}^{v,w}(K) = \frac{1}{C_K} \sum_{(k,I)\in\mathbb{UT}_K\cup\mathbb{UR}_K} \frac{(l-k)^v}{l^w} q_{kl}, \tag{2.44}$$

the difference between MRRE and T&C are the weighting of elements of q_{kl} and the blocks of $\mathbf{Q}$ covered. MRRE covers the first K rows and columns of $\mathbf{Q}$. Thus, the first error involves all K-intrusions and the mild K-extrusions and the second conveys K-extrusions and the mild K-intrusions. Another method that can be defines in terms of the co-ranking matrix is LCMC (subsection 2.5.5) such as

$$U_{LC} = \frac{K}{1-N} + \frac{1}{NK} \sum_{(k,l)\in U\mathbb{L}_K} q_{kl}, \tag{2.45}$$

this method is computed over the block $\mathbb{UL}_K$ from $\mathbf{Q}$ which elements are not weighted and the normalization is simpler. The unified framework defined by Lee and Verleysen relates the co-ranking matrix with the concepts of precision and recall [53] which is also related with false positive and false negative classification [7]. In order to define Q_{NX} and R_{NX} this framework defines fractions of mild K-intrusions and mild K-extrusions as

$$U_{\mathrm{N}}(K) = \frac{1}{KN} \sum_{(k,l)\in\mathbb{LT}_K} q_{kl} \text{ and } U_{\mathrm{X}}(K) = \frac{1}{KN} \sum_{(k,l)\in\mathbb{UT}_K} q_{kl}, \tag{2.46}$$

and the fraction of vectors that keep the same rank in both neighborhoods v_i^K and n_i^K as

$$U_{\mathrm{P}}(K) = \frac{1}{KN} \sum_{(k,l)\in\mathbb{D}_K} q_{kl}, \tag{2.47}$$

with all the previous definitions Q_{NX} can be defined. It is U_{LC} with the difference that Q_{NX} do not have the subtraction of the 'random baseline'. The range is $Q_{NX} \in [0,1]$ where 1 means perfect embedding. in terms of the coranking matrix and U_{LC}, Q_{NX} is defined as:

$$Q_{\mathrm{NX}}(K) = U_{\mathrm{P}}(K) + U_{\mathrm{N}}(K) + U_{\mathrm{X}}(K) = U_{\mathrm{LC}}(K) + \frac{K}{N-1} \tag{2.48}$$

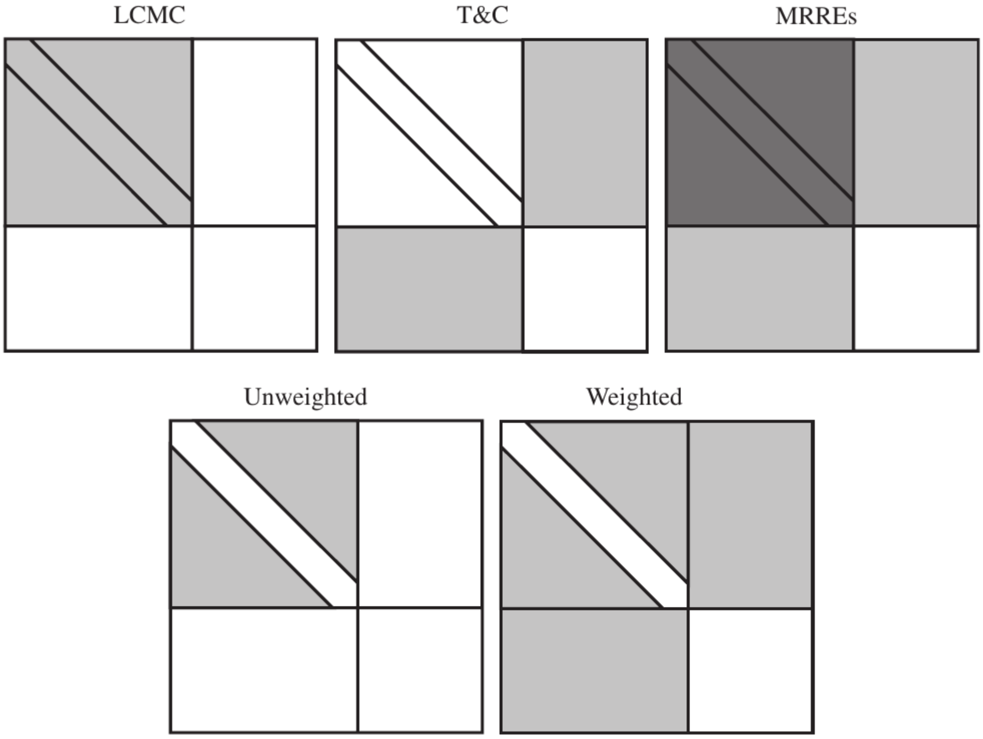


Figure 2.3: Co-ranking Matrix.

R_{NX} can be viewed as a renormalized U_{LC} which allows comparing values at different scales [54]. As other measures R_{NX} range is $[0, 1]$ where 1 represents perfect embedding. R_{NX} renormalization is given by

$$R_{NX}(K) = \frac{(N-1)Q_{NX}(K) - K}{N-1-K}. \tag{2.49}$$

Chapter 3

Methodology

Dimensionality reduction focus on providing low-dimensional representations of high-dimensional data sets, and the proposed method looks for the quality assessment of this low dimensionality representation using the R_{NX} curve based on the co-ranking matrix. In short, the low dimensional representation of high dimensional data is calculated. From this, we cal calculate the co-ranking matrix, which framework contains the RNS curve quality assessment.

3.1 Method flowchart

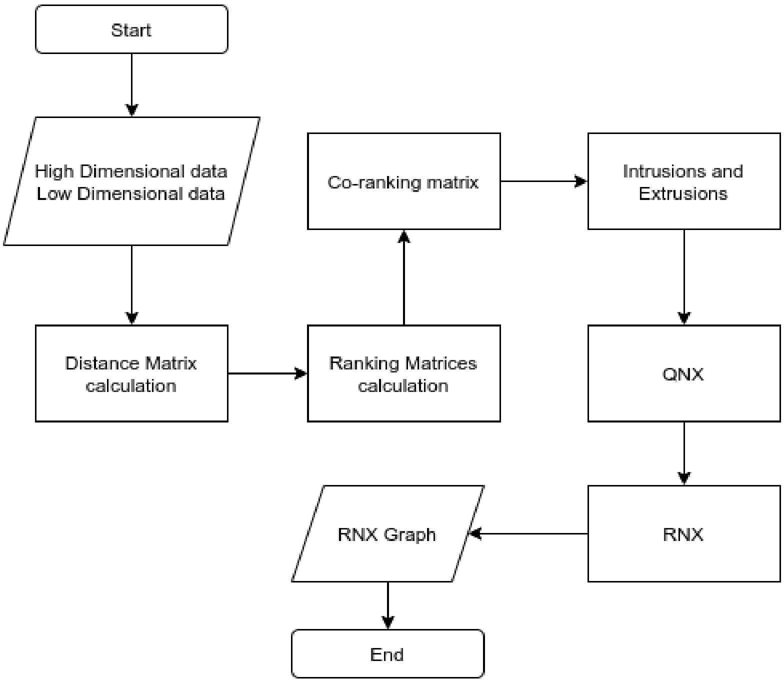


Figure 3.1: Proposed R_{NX} curve flowchart. It contains the steps for obtaining R_{NX} curve from a high dimensional and low dimensional matrix.

3.1.1 Data matrix and dimensionality reduction

The data sets **X** used in this work are specified in section 3.2. We first proceed to perform a dimensionality reduction from each data set using the methods Locally Linear Embedding, Laplacian eigenmaps, Multidimensional Scaling, and Kernel Locally linear embedding. Once we have the low-dimensional representation **Y** of the high-dimensional data set **X** lets feed the algorithm with both matrices.

3.1.2 Distance matrix calculation

At this point, we count with a high dimensional matrix and a low dimensional matrix. From here, we get the pairwise distances matrix D_X and D_Y. Then, the indices that would sort these matrices are obtained as matrices D_{Xs} and D_{Ys}.

3.1.3 Ranking matrices calculation

From D_{Xs} and D_{Ys}, using equations 2.34 and 2.35 respectively we proceed to calculate the ranking matrices R_X and R_Y. The implemented algorithm is the following

```
input: Dys,Dxs
ldrank[Rows,Cols]
hdrank[Rows,Cols]

For j=1:Rows
    For i=1:Cols
        ldrank[Dys[i,j],j]=i
        hdrank[Dxs[i,j],j]=i
output: ldrank, hdprank
```

Where ldrank (R_Y) and hdrank (R_X) are the low dimensional rank matrix and high dimensional rank matrix, respectively.

3.1.4 Co-ranking matrix

Using R_X and R_Y we proceed to calculate the co-ranking matrix following the equation 2.36. Translated to an algorithm we have

```
input: ldrank,hdrank
For j=1:Rows
    For i=1:Cols
        k = hdrank[i,j]
        l = ldrank[i,j]
        c[k,l] = c[k,l]+1
Remove first row and column from c
output: c
```

3.1.5 Intrusions and extrusions

With the co-ranking matrix we can calculate $Q_{NX}(K)$. But, before calculating $Q_{NX}(K)$ we need to calculate the intrusions and extrusions. In this algorithm, we take into account the mild K-intrusions U_N and mild K-extrusions U_K with the equations 2.46, the fraction vector U_P from equation 2.47 and the 'random baseline' The algorithm is as follows:

```
input: c
v1[Rows+1]
v2[Rows+1]
For i=1:Rows+1
    v2=v1[i]*Rows+1

For k=1:Rows
    n[k]=sum[c[k,0:k]]
    x[k]=sum[c[0:k,k]]

n = accumulativesum(n)/v2
x = accumulativesum(x)/v2

d=diagonalfrom(c)
p=accumulativesum(c)

b=v1/(1/Rows)
output: n,x,p,b
```

Where c is the co-ranking matrix and n, x, p, b are U_N, U_X, U_P, 'random baseline' respectively. Figure 3.2 shows from a graphical manner the types of intrusions and extrusions calculated in this algorithm [7].

3.1.6 QNX

With the outputs from the last algorithm we can calculate Q_{NX} as $qnx = n+x+p$ following the equation 2.48.

3.1.7 RNX

The normalization of Q_{NX} to obtain R_{NX} is given by equation 2.49 and the respective algorithm implementation is

```
input: qnx
lcmc = qnx - b
tmp = 1 - b
rnx = lcmc/tmp
output:rnx
```

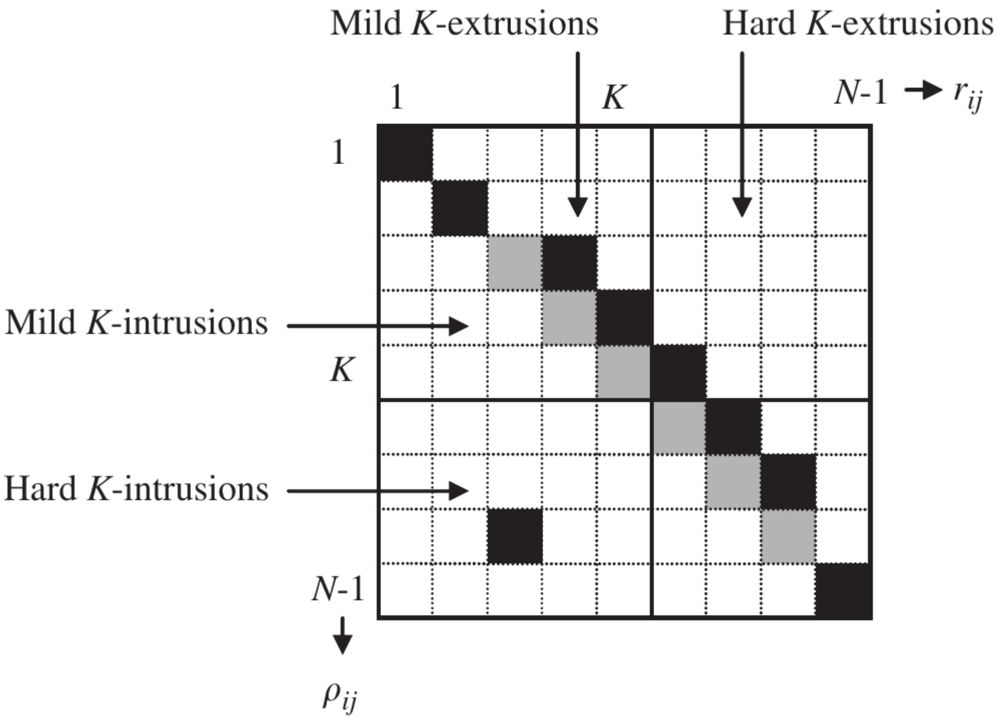


Figure 3.2: Different types of intrusions and extrusions.

With the rnx vector we can draw the $R_{NX}(K)$. Fort he graph we multiply the values from rnxby 100 in order to get a percentage representation. From this vector we can obtain the area under the curve with the following algorithm

```
input: rnx
wgh = 1/[1...len(rnx)]
s = sum(wgh)
wgh = wgh/s
rnx_auc = wgh . rnx
output: rnx_auc
```

where rnx_auc is the area under the curve of R_{NX} and it is also multiplied by 100 in the graphs shown.

3.2 Experimental setup

This section aims at evaluating through experimentation the validity of the R_{NX} curve developed in python. For this purpose, four data sets are employed (section 3.2.1), and two experiments are set. The first experiment compares the two versions of the R_{NX} curve: the recently created Python implementation against the original developed in Matlab. For this first experiment, seven DR techniques were applied to each data set using the two implementations. On the other hand, the second experiment is designed to evaluate the kernel approximation quality using the python implementation. Here three DR methods were used along with their Kernel trick implementation.

3.2.1 Databases

The experiments involve four data sets. The first one contains 1000 uniformly sampled points from the surface of a sphere. Here, the colors are constant along the longitudes of the spherical shell. The second data set includes 1000 uniformly distributed sampled points from the surface of a Swiss-roll. The Columbia Object Image Library (COIL-20) is the third data set [55]. It consists of 72 gray-level normalized images of 20 various objects. Each one represents a 4-degree rotation around every object. Some of these images can be seen in Fig. 3.3 The fourth data set is a random subset of the MNIST database of handwritten digits [56]. It contains 1000 gray-level images of scanned handwritten digits (out of 60 000). Each 28 by 28 image is vectorized in order to be fed to various NLDR algorithms without any other preprocessing.

3.2.2 Python implementation vs MAT-LAB implementation

This first experiment compares the R_{NX} quality curves obtained with Matlab against the Python implementation ones. For this purpose, we use the data sets described before in section 3.2.1 and seven DR methods: LLE, Isomap, MDS, PCA, LE, and t-SNE. Every method is applied to each data set. Each of the low-dimensional representations Y was put in a file alongside its original representation X. Next, we proceeded to read the files with Matlab and Python so that their respective R_{NX} implementation can process the matrices. Both implementations output a vector containing the curve to be drawn and the area under it. This vector and area under the curve are again saved in files and drawn in pairs (Matlab and Python implementation) for a better graphical comparison of both implementations. With the resulting vectors, the mean squared error between them is calculated, and if the Python implementation is correct, the mean square error should approximate to zero. A flowchart of the procedure is presented in Fig 3.4.

3.2.3 Evaluating kernel approximation

A conventional spectral DR method transforms high-dimensional data into a low-dimensional representation. The Kernel approximation of these methods performs the same task with similar results. In this experiment, three spectral DR methods alongside their Kernel approximations are used: Locally Linear Embedding, Laplacian eigenmaps, and Isomap. The latter mentioned should perform (approximate) the R_{NX} curve obtained with the conventional methods. Every conventional method and its kernel approximation are applied to each of the last used data sets. Then, both results are drawn in one figure to visualize the similarity. Fig. 3.5 shows a flowchart of this experiment. The number of figures obtained from this experiment is twelve, each with two R_{NX} curves, one representing the conventional method and the other the kernel approximation of that particular method.

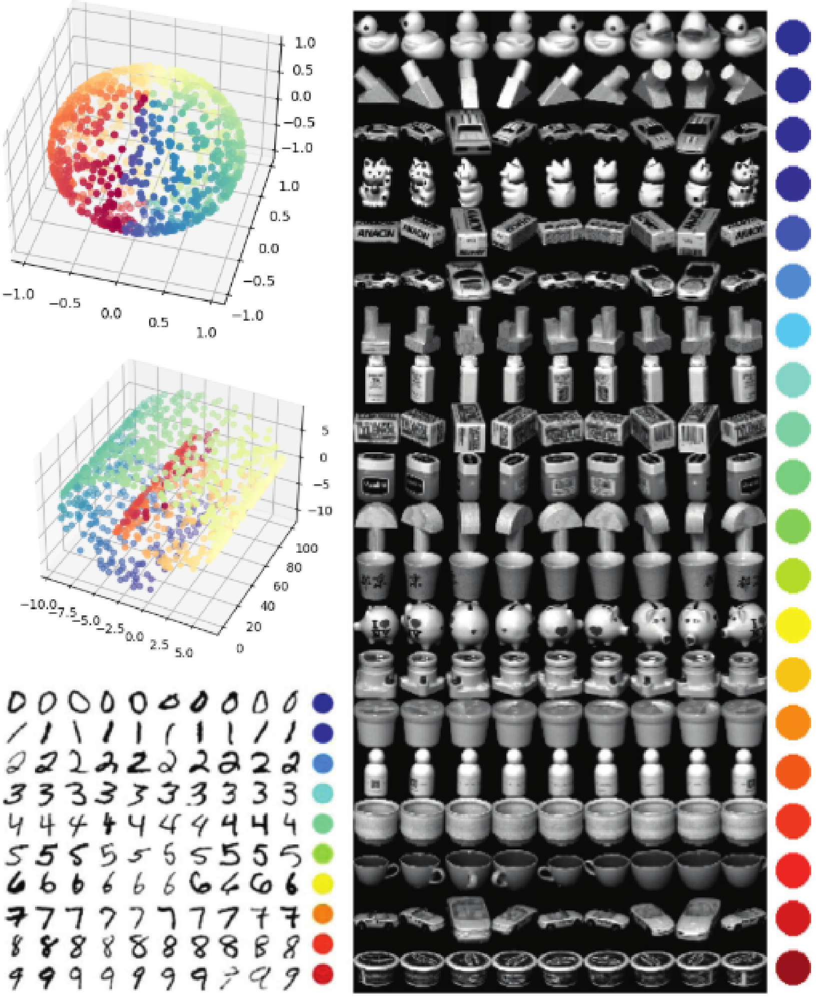


Figure 3.3: The four data sets used in the experiments are the spherical shell, the COIL-20 image bank, the Swiss Roll, and a random subset of the MNIST image bank. The COIL-20 and MNIST images are just vectorized before dimensionality reduction. In all four cases, a two-dimensional embedding is sought.

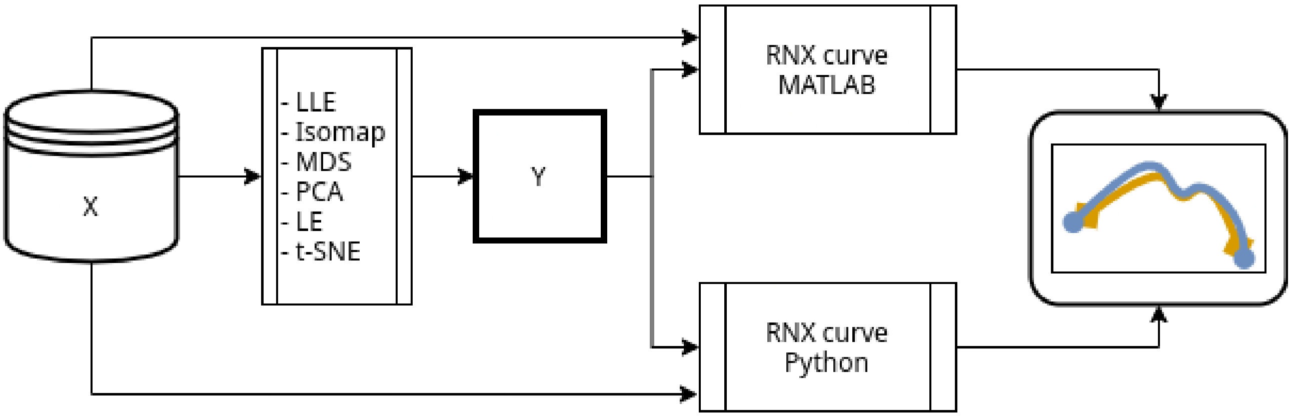


Figure 3.4: Methodology for comparing the Matlab implementation against Python implementation.

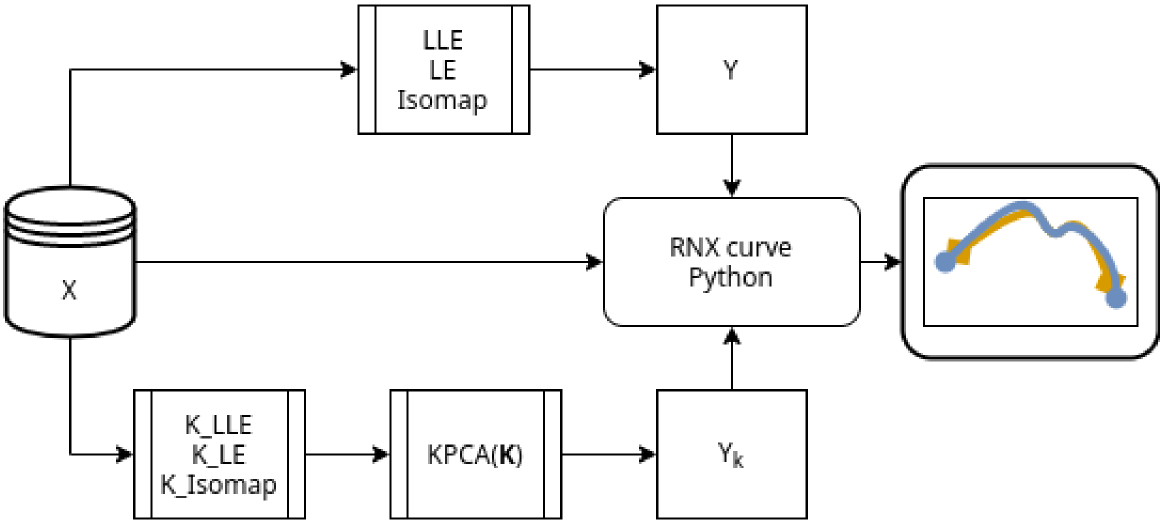


Figure 3.5: Methodology for comparing the Kernel methods KLE, KLLE, and KIsomap with their respective conventional method.

Chapter 4

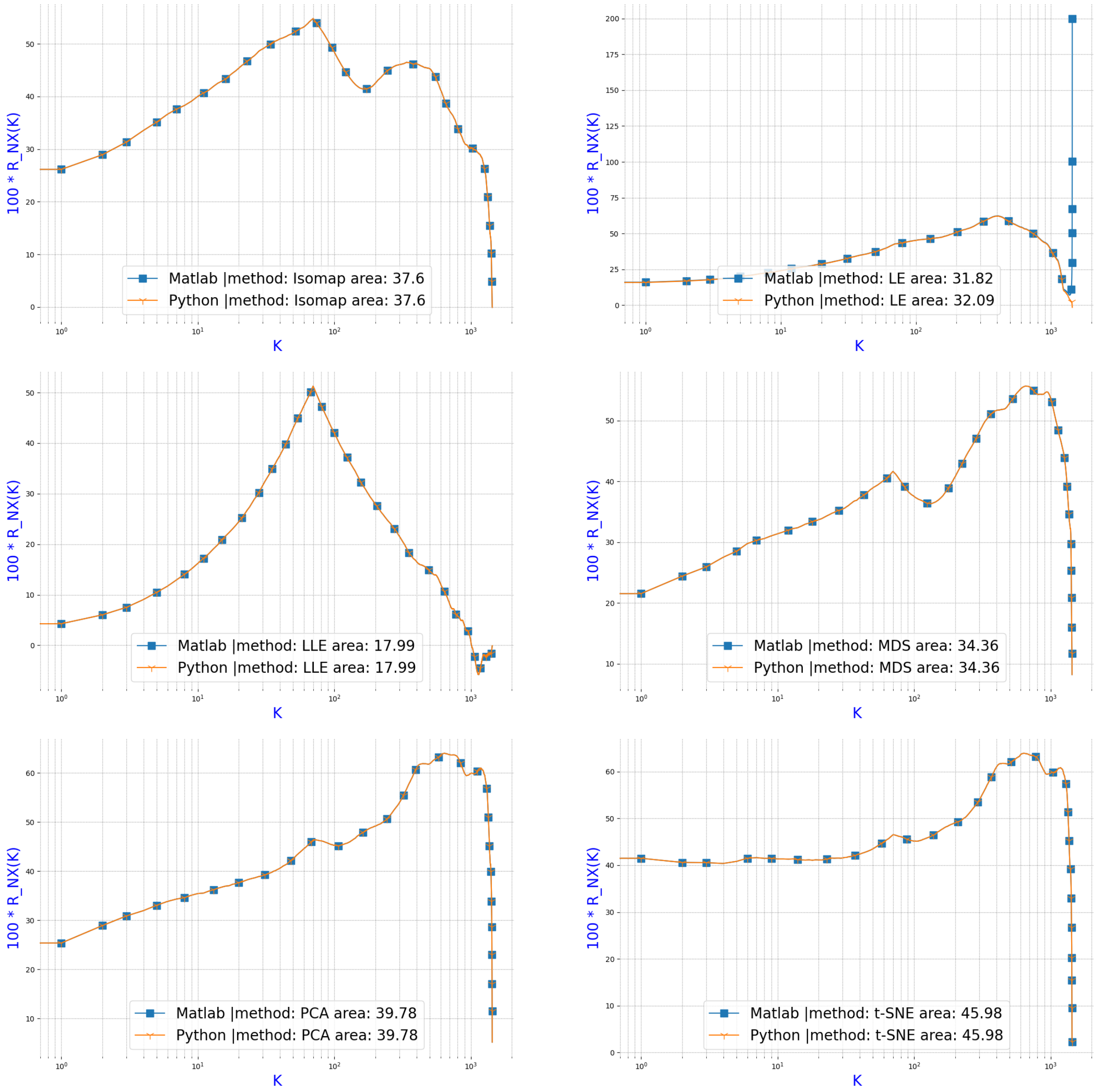
Results and Discussion

4.1 Implementation comparison results

In this section, we present the results from experiment 1. Here, we compare the R_{NX} curve implementation developed in this work against the preexisting Matlab implementation. This experiment validates the functioning of the Python R_{NX} quality curve module by creating the quality curve for different DR methods in Python and Matlab, using four different datasets with six different DR methods. Figures **??**, 4.2, 4.3, 4.4,represent the R_{NX} quality curves for different DR methods,drawn with the Matlab Implementation and the Python implementation, from the databases Coil-20, MNIST, Sphere, and Swiss Roll respectively. Graphically, an overlap on the curves is observed, which tells us that the Python implementation works like the Matlab implementation.

Another essential factor of this experiment is that it visually compares different DR methods using the R_{NX} curve, which evaluates the topology preservation of the low dimensional representations. If the curve shows asymmetric forms to the right, it means global topology preservation and if the curve is asymmetric to the left means local topology reservation. Section 2.3 mention convex DR techniques, which optimize an objective function without local optima. Similarly, section 2.3.3 talks about non-convex techniques which do contain local optima.

This experiment shows the non-local approach and the local approaches in the convex and non-convex techniques; this is achieved through the R_{NX} curve. Used techniques: Isomap, Le, LLE, MDS, and PCA are classified as convex techniques. Using the Coil-20 database, all of these techniques present a curve to the right, which points out global topology preservation. On the other hand, t-SNE is classified as a non-convex technique, and all figures from this experiment show an asymmetric curve to the left for this method in all databases which means local topology preservation. All the figures show an overlap in the Python and Matlab Implementation, but a better validation is constructed by measuring the mean squared error between the Matlab implementation vector and the Python implementation. Table 4.1 shows the AUC from Matlab and Python's curve and the mean square error for each DR method used with every dataset. As observed in this table, the AUCs for both methods are in most of the cases the same, and the mean squared error is practically zero. Thus, we have a better validation of the Python implementation.



captionR_{NX} quality curve comparison between Matlab and Python using the Coil-20 database

4.2 Evaluating kernel approximation results

This section presents the results for experiment 2. As we saw in section 2.4, the spectral DR methods can be represented with kernel matrices allowing KPCA to perform a quality reduction that approximates the traditional method. This experiment validates the functioning of the Python R_{NX} quality curve module by performing a graphical validation of kernel matrices as DR approximations from conventional spectral methods. Thus, the representation quality curve between the traditional method and the kernel method is approximately the same.

As shown in this experiment's results, the kernel approximations KLLE, KLE, and KIsomap present an almost equal quality curve compared to their equivalent conventional method. Figures 4.5, 4.6, 4.7, 4.8 present the R_{NX} quality curve comparison of DR conventional methods and their corresponding kernel representation for the databases coil20,

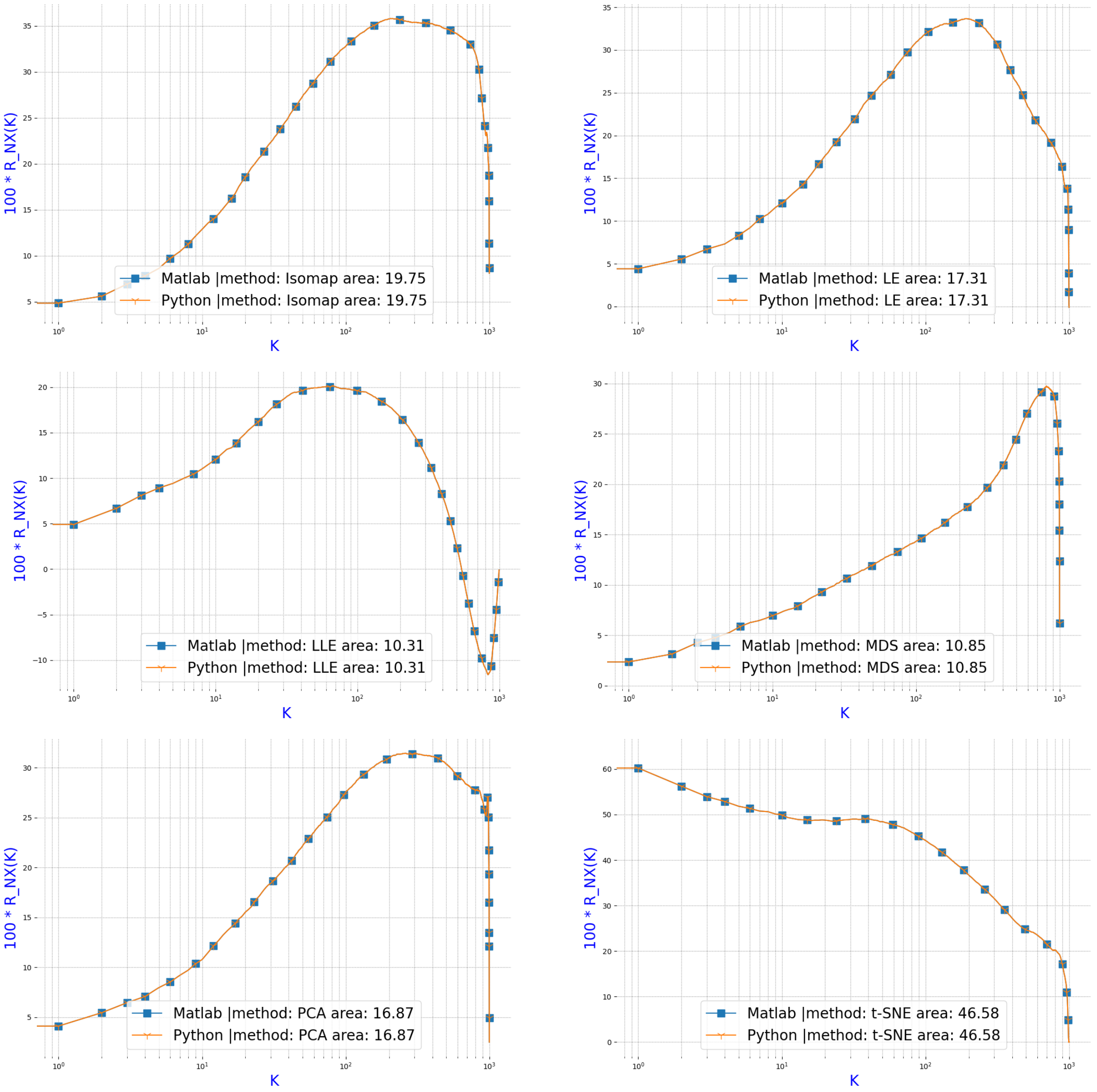


Figure 4.2: R_{NX} quality curve comparison between Matlab and Python using the MNIST database

MNIST, Sphere, and Swiss Roll, respectively. It is seen that both methods, kernel and conventional, generate a good representation of the databases, except for the KIsomap, where the curve does not approximate to the traditional method. The figures also show the area under the curve for each method. Areas from LE and LLE are very similar to the area from KLE and KLLE, respectively.

As observed from the results, the kernel matrix presents a good approximation for the traditional DR methods LE and LLE. This is seen from the point of view of the quality curve and its area under the curve. Except for Isomap, the kernel quality curve with their area under the curve approximated very accurately to the traditional method's curve.

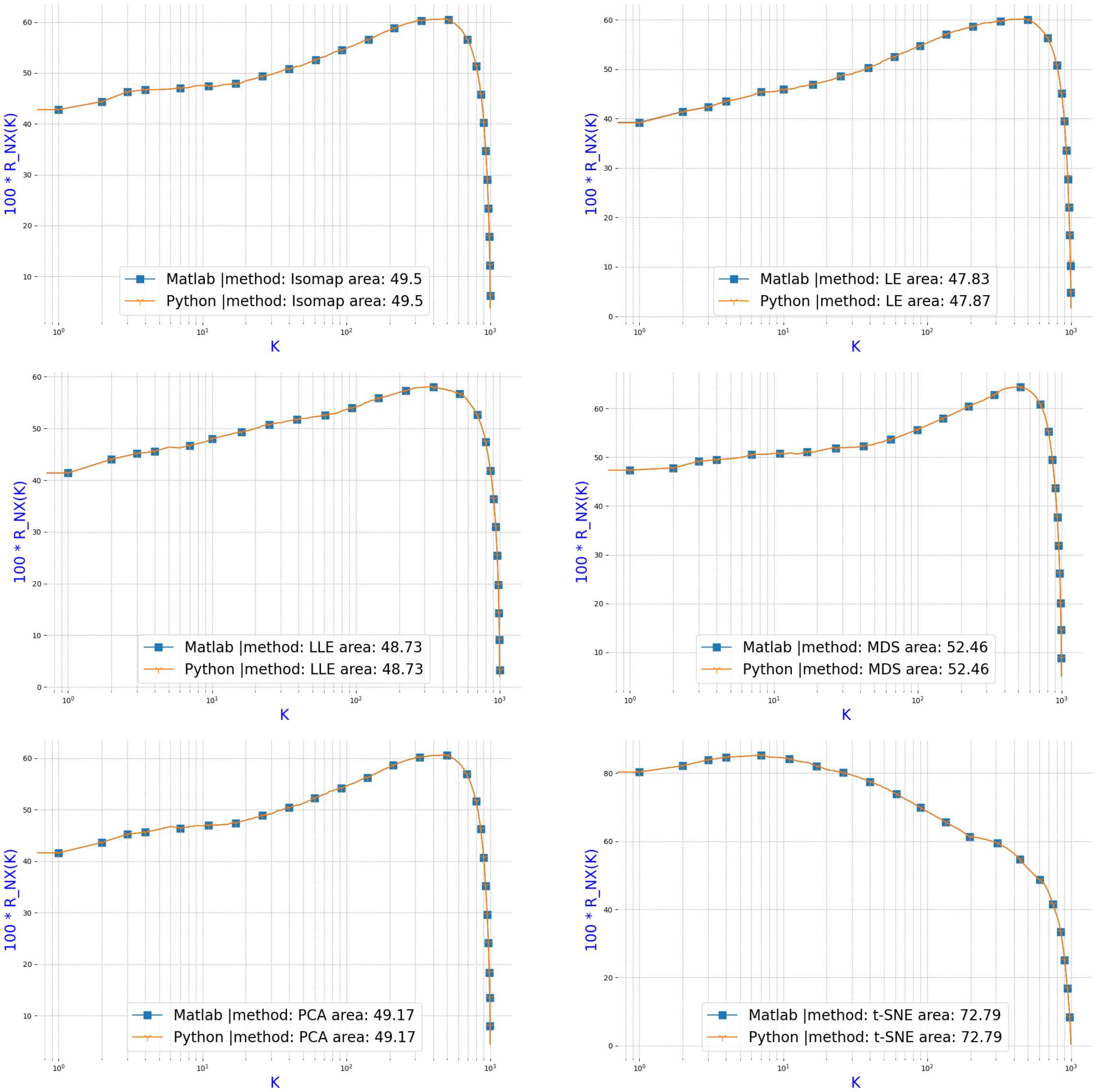


Figure 4.3: R_{NX} quality curve comparison between Matlab and Python using the Sphere database.

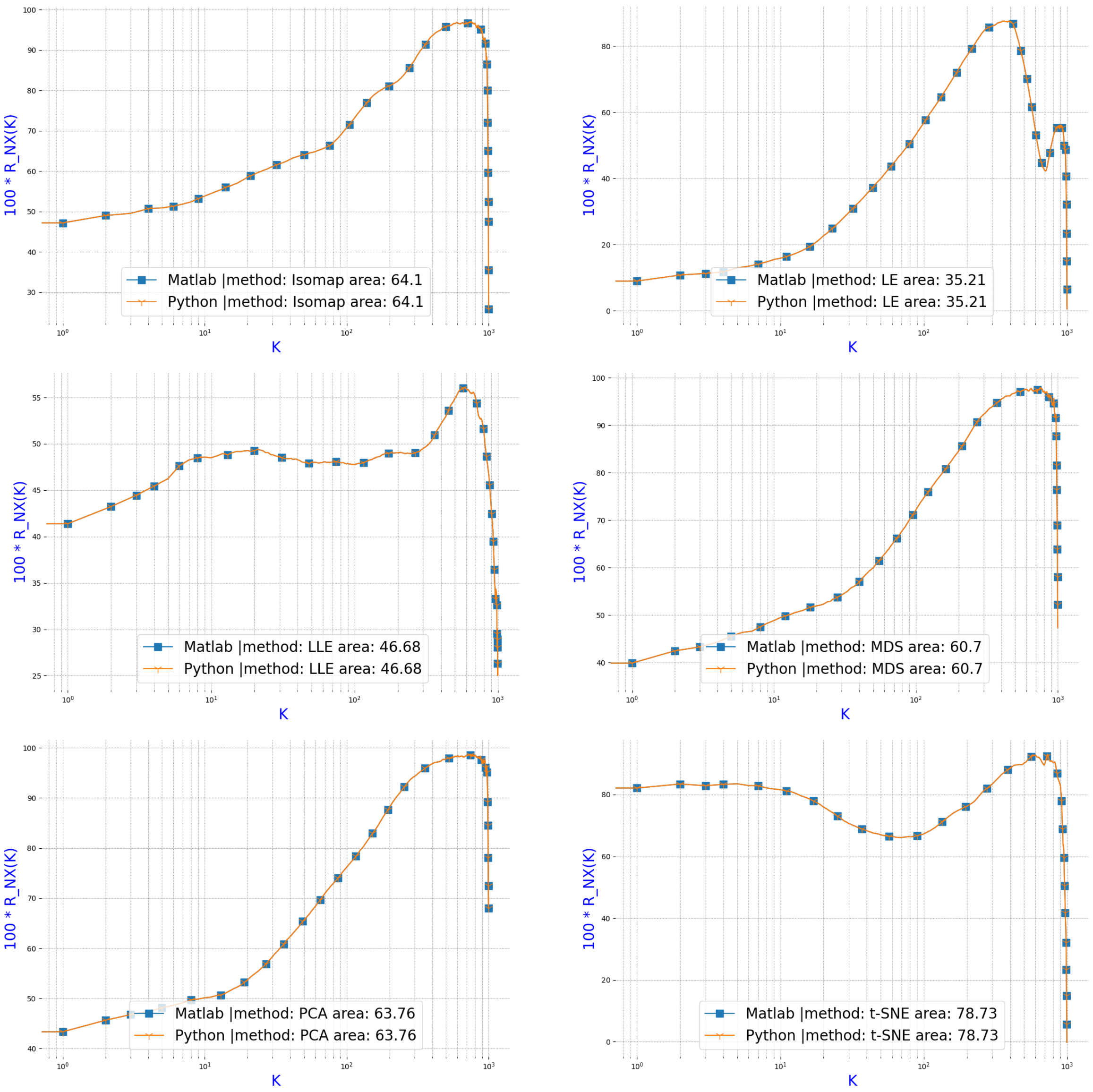


Figure 4.4: R_{NX} quality curve comparison between Matlab and Python using the Swiss Roll database.

Dataset	**DR method**	**AUC Matlab**	**AUC Python**	**RNX mse**
Coil20	LE	0.321	0.318	0.004621
Coil20	PCA	0.398	0.398	9.08E-15
Coil20	t-SNE	0.46	0.46	1.66E-13
Coil20	LLE	0.18	0.18	2.90E-15
Coil20	MDS	0.344	0.344	2.13E-14
Coil20	Isomap	0.376	0.376	5.84E-30
MNIST	LE	0.173	0.173	5.42E-13
MNIST	PCA	0.169	0.169	5.08E-13
MNIST	t-SNE	0.466	0.466	5.93E-13
MNIST	LLE	0.103	0.103	3.82E-12
MNIST	MDS	0.109	0.109	5.61E-13
MNIST	Isomap	0.198	0.198	3.63E-12
Swiss	LE	0.352	0.352	6.77E-32
Swiss	PCA	0.638	0.638	2.36E-33
Swiss	t-SNE	0.787	0.787	4.04E-13
Swiss	LLE	0.467	0.467	8.71E-32
Swiss	MDS	0.607	0.607	1.08E-32
Swiss	Isomap	0.641	0.641	1.35E-32
Sphere	LE	0.479	0.478	1.32E-08
Sphere	PCA	0.492	0.492	1.19E-31
Sphere	t-SNE	0.728	0.728	2.03E-13
Sphere	LLE	0.487	0.487	1.25E-31
Sphere	MDS	0.525	0.525	1.14E-31
Sphere	Isomap	0.495	0.495	1.19E-31

Table 4.1: AUC comparison from the Matlab and Python implementation, and their corresponding R_{NX} mean squared error.

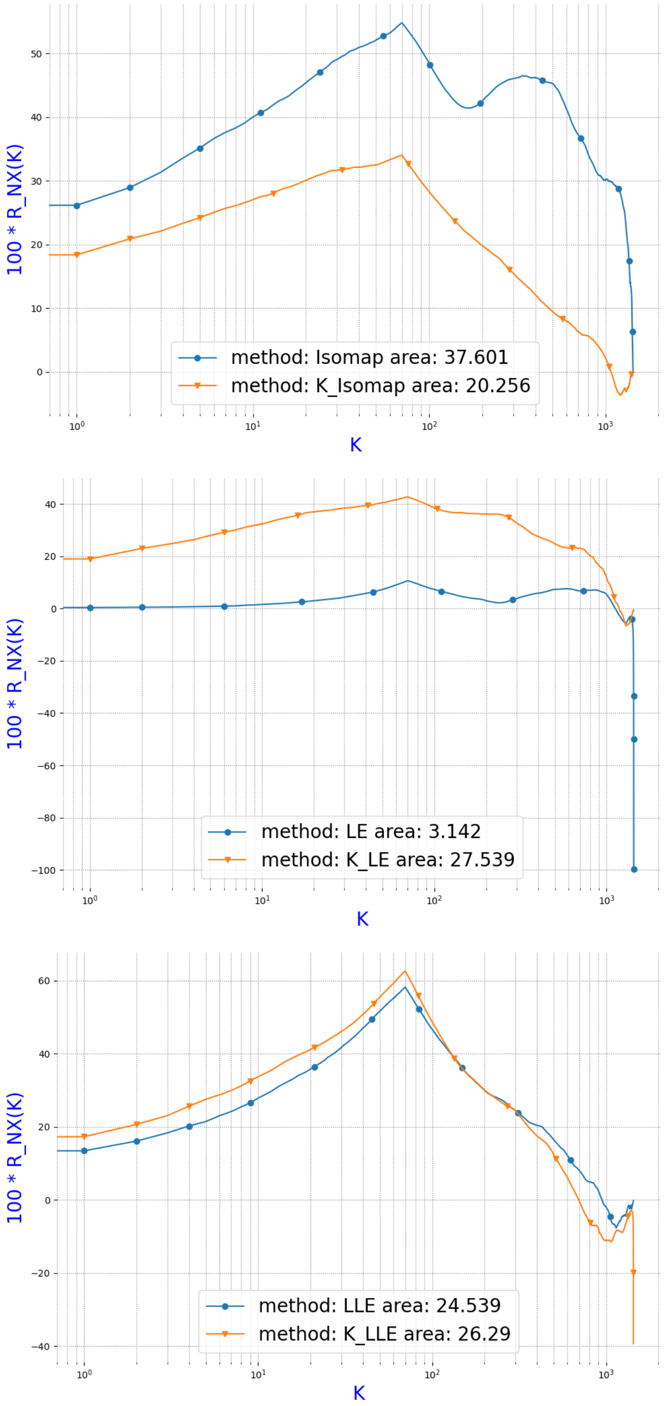


Figure 4.5: R_{NX} curve comparison from each DR spectral method and their respective kernel approximation, applied to Coil-20 database

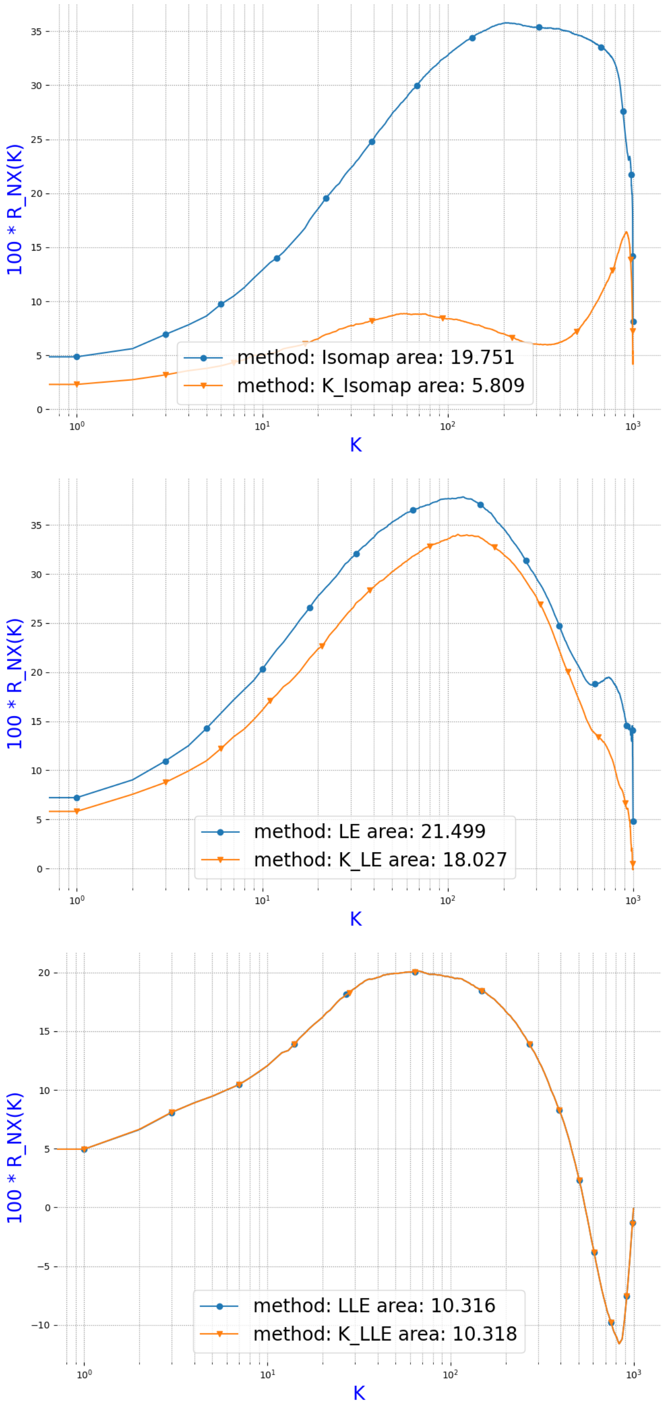


Figure 4.6: R_{NX} curve comparison from each DR spectral method and their respective kernel approximation, applied to MNIST database.

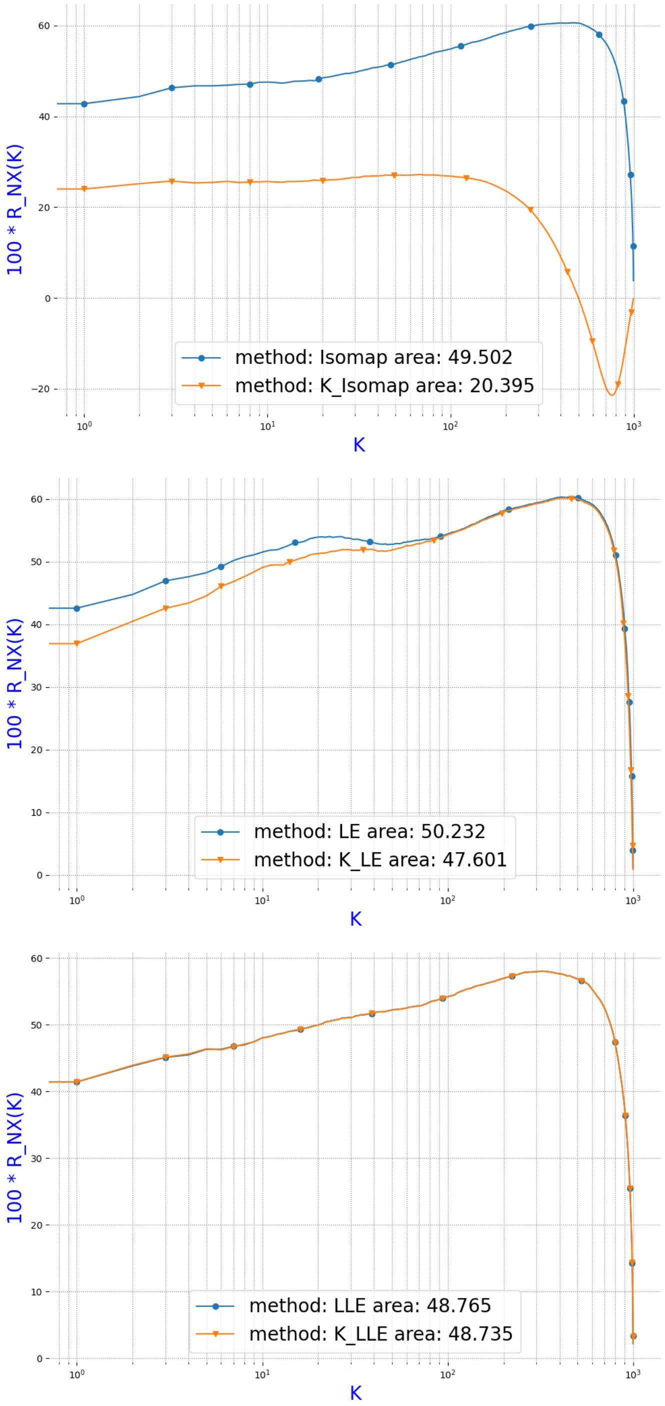


Figure 4.7: R_{NX} curve comparison from each DR spectral method and their respective kernel approximation, applied to Sphere database.

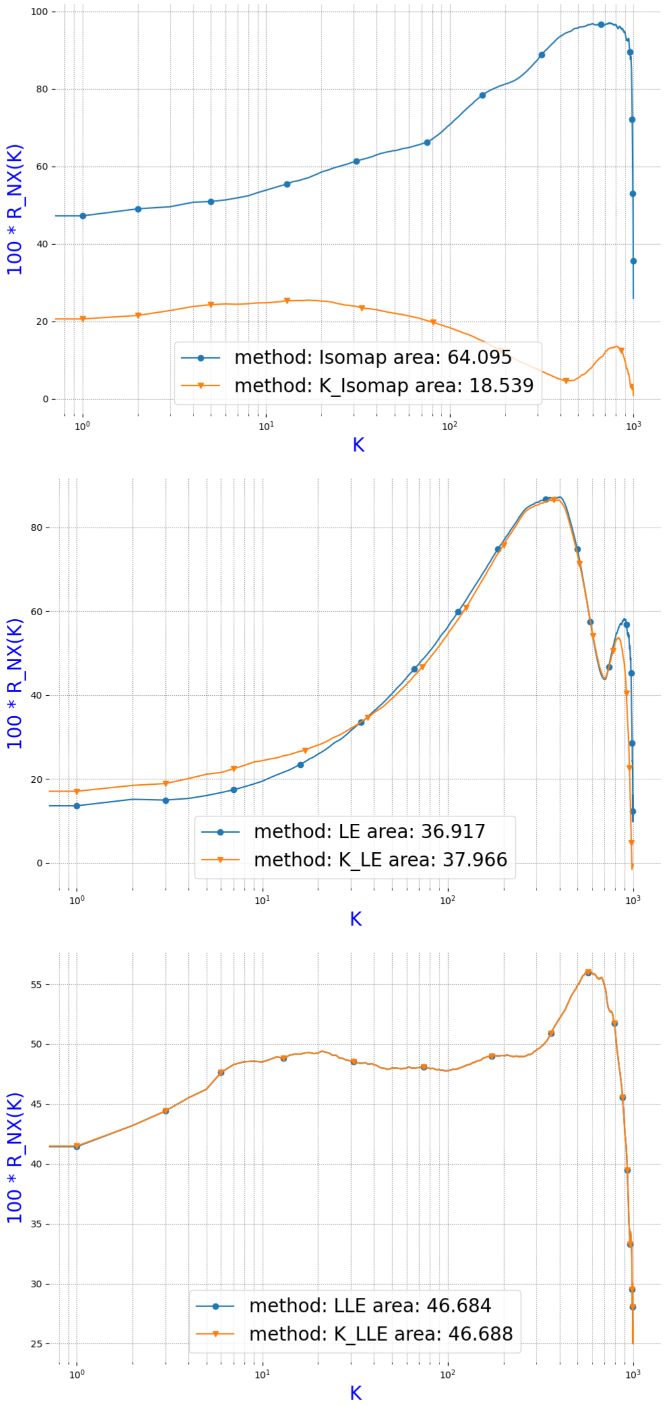


Figure 4.8: R_{NX} curve comparison from each DR spectral method and their respective kernel approximation, applied to Swiss Roll database.

Chapter 5

Conclusions

Many DR approaches seek topology preservation. These methods can be divided into convex and non-convex techniques. The former optimizes an equation that does not contain a local optima, and the latter optimizes equations that do contain local optima. Inside convex techniques, we have the subdivision of *full spectral*, which carries out a full matrix eigen-decomposition, and *sparse spectral* techniques that solve a generalized eigenproblem. Both methods obtain the covariances between dimensions or similarities between data points.

This work validates the Python R_{NX} curve implementation, comparing it with the Matlab existing one. Data sets Coil-20, MNIST, Sphere, and Swiss Roll along different DR methods were chosen for this validation. The methods Isomap, LE, LLE, MDS, PCA, and t-SNE were applied to each data set using Python, and compared with the curve from the Matlab implementation. Both methods perform similarly, and results are equivalent graphically, as well numerically. This test also allows us to appreciate the difference between different methods. Consequently, convex methods showed a curve to the right, meaning non-local optima, and t-SNE, which is not convex, showed a curve to the left, meaning local optima.

This work has mentioned Isomap, LLE, and LE, which are methods based on graphs that perform an eigendecomposition of a Laplacian matrix. These methods can be represented through a kernel matrix and KPCA. Using the developed Python implementation, a comparison between conventional methods and kernel methods was conducted, showing that in LLE and LE, the Kernel approximation is almost equivalent to the conventional method.

As the existing implementation was written in Matlab, it was limited for future development, and cannot be implemented in other technologies. With our new implementation developed in Python, we overcome that challenge. The proposed implementation can be further integrated into new frameworks of dimensionality reduction and quality assessment.

Bibliography

[1] D. Peluffo, J. Lee, and M. Verleysen, “Recent methods for dimensionality reduction: A brief comparative analysis,” 01 2014.

[2] P. Prasdika and B. Sugiantoro, “A review paper on big data and data mining concepts and techniques,” *IJID (International Journal on Informatics for Development)*, vol. 7, p. 33, 12 2018.

[3] N. Renard, S. Bourennane, and J. Blanc-Talon, “Denoising and dimensionality reduction using multilinear tools for hyperspectral images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 2, pp. 138–142, 2008.

[4] D. S Velliangiri, S. Alagumuthukrishnan, and I. T. J. Swamidason, “A review of dimensionality reduction techniques for efficient computation,” 01 2020.

[5] J. A. Lee, D. H. Peluffo-Ordóñez, and M. Verleysen, “Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure,” *Neurocomputing*, vol. 169, pp. 246–261, 2015, learning for Visual Semantic Understanding in Big Data ESANN 2014 Industrial Data Processing and Analysis. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231215003641

[6] L. Van Der Maaten, E. Postma, and J. Van den Herik, “Dimensionality reduction: a comparative review,” *J Mach Learn Res*, vol. 10, pp. 66–71, 2009.

[7] J. A. Lee and M. Verleysen, “Quality assessment of dimensionality reduction: Rank-based criteria,” *Neurocomputing*, vol. 72, no. 7, pp. 1431–1443, 2009, advances in Machine Learning and Computational Intelligence. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231209000101

[8] M. Butwall, P. Ranka, and S. Shah, “Python in field of data science: A review,” *International Journal of Computer Applications*, vol. 178, pp. 20–24, 09 2019.

[9] S. Tuffry, *Data Mining and Statistics for Decision Making*, 1st ed. Wiley Publishing, 2011.

[10] K. Fukunaga, *Introduction to Statistical Pattern Recognition (2nd Ed.)*. USA: Academic Press Professional, Inc., 1990.

[11] X. Zeng, Y.-W. Chen, and C. Tao, "Feature selection using recursive feature elimination for handwritten digit recognition," in *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2009, pp. 1205–1208.

[12] X.-w. Chen and J. C. Jeong, "Enhanced recursive feature elimination," 01 2008, pp. 429 – 435.

[13] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," *Proceedings of 2014 Science and Information Conference, SAI 2014*, pp. 372–378, 10 2014.

[14] J. Lee, *Introduction to Topological Manifolds*, ser. Graduate Texts in Mathematics. Springer New York, 2010. [Online]. Available: https://books.google.de/books?id=ZQVGAAAAQBAJ

[15] A. Gracia, S. González, V. Robles, and E. Menasalvas, "A methodology to compare dimensionality reduction algorithms in terms of loss of quality," *Information Sciences*, vol. 270, pp. 1–27, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025514001741

[16] J. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*, 01 2007, vol. 8226.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[18] P. Boileau, N. S. Hejazi, and S. Dudoit, "Exploring high-dimensional biological data with sparse contrastive principal component analysis," *Bioinformatics*, vol. 36, no. 11, pp. 3422–3430, 03 2020. [Online]. Available: https://doi.org/10.1093/bioinformatics/btaa176

[19] B. NICULESCU and G. Andrei, "Principal component analysis as a tool for enhanced well log interpretation," vol. 60, p. 49–61, 01 2016.

[20] N. Qureshi, V. Suthar, H. Magsi, M. Sheikh, M. Pathan, and B. Qureshi, "Application of principal component analysis (pca) to medical data," *Indian Journal of Science and Technology*, vol. 10, pp. 1–9, 02 2017.

[21] A. Rencher and W. Christensen, *Methods of Multivariate Analysis*, ser. Wiley Series in Probability and Statistics. Wiley, 2012. [Online]. Available: https://books.google.com.ec/books?id=fWMTV3wSpTcC

[22] S. Mishra, U. Sarkar, S. Taraphder, S. Datta, D. Swain, R. Saikhom, S. Panda, and M. Laishram, "Principal component analysis," *International Journal of Livestock Research*, p. 1, 01 2017.

[23] W. S. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, pp. 401–419, 1952.

[24] H. Strange and R. Zwiggelaar, *Open Problems in Spectral Dimensionality Reduction*, 01 2014.

[25] J. Tenenbaum, V. Silva, and J. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science (New York, N.Y.)*, vol. 290, pp. 2319–23, 01 2001.

[26] M. Balasubramanian and E. Schwartz, “Schwartz, e.l.: The isomap algorithm and topological stability. science 295, 5552,” *Science (New York, N.Y.)*, vol. 295, p. 7, 02 2002.

[27] C. Shao and H. Huang, “Improvement of data visualization based on isomap,” 11 2005, pp. 534–543.

[28] X. Liu, P. Ma, and G. Li, “Research on adaptive ISOMAP algorithm and application in intrusion detection,” *Journal of Physics: Conference Series*, vol. 1607, p. 012130, aug 2020. [Online]. Available: https://doi.org/10.1088/1742-6596/1607/1/012130

[29] Q. Wang, “Kernel principal component analysis and its applications in face recognition and active shape models,” *CoRR*, vol. abs/1207.3538, 2012. [Online]. Available: http://arxiv.org/abs/1207.3538

[30] L. van der Maaten, E. O. Postma, and H. J. van den Herik, “Dimensionality reduction: A comparative review,” 2008.

[31] C. Chatfield, *Introduction to multivariate analysis*, ser. Science paperbacks. London: Chapman and Hall, 1980.

[32] M. V. Ramana and P. M. Pardalos, *Semidefinite Programming.* Boston, MA: Springer US, 1996, pp. 369–398. [Online]. Available: https://doi.org/10.1007/978-1-4613-3449-1_9

[33] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker, “Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, pp. 7426–31, 06 2005.

[34] S. Lafon and A. Lee, “Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, pp. 1393–403, 10 2006.

[35] S. Roweis and L. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science (New York, N.Y.)*, vol. 290, pp. 2323–6, 01 2001.

[36] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, ser. NIPS’01. Cambridge, MA, USA: MIT Press, 2001, p. 585–591.

[37] W. N. A. Jr. and T. D. Morley, “Eigenvalues of the laplacian of a graph,” *Linear and Multilinear Algebra*, vol. 18, no. 2, pp. 141–145, 1985. [Online]. Available: https://doi.org/10.1080/03081088508817681

[38] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimension reduction via local tangent space alignment," *SIAM J. Sci. Comput.*, vol. 26, 01 2003.

[39] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. C-18, no. 5, pp. 401–409, 1969.

[40] M. A. A. Cox and T. F. Cox, *Multidimensional Scaling.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 315–347. [Online]. Available: https://doi.org/10.1007/978-3-540-33037-0_14

[41] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science (New York, N.Y.)*, vol. 313, pp. 504–7, 08 2006.

[42] D. Peluffo, J. Lee, and M. Verleysen, "Generalized kernel framework for unsupervised spectral methods of dimensionality reduction," 12 2014.

[43] K. Weinberger, F. Sha, and L. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," 07 2004.

[44] Q. Wang, "Kernel principal component analysis and its applications in face recognition and active shape models," *CoRR*, vol. abs/1207.3538, 2012. [Online]. Available: http://arxiv.org/abs/1207.3538

[45] L. Belanche, "Developments in kernel design," in *21st European Symposium on Artificial Neural Networks, ESANN 2013, Bruges, Belgium, April 24-26, 2013*, 2013. [Online]. Available: http://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2013-10.pdf

[46] J. Ham, D. Lee, S. Mika, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds," 07 2004.

[47] L. E. Toothaker, "Book review : Nonparametric statistics for the behavioral sciences (second edition): Sidney siegel and n. john castellan, jr. new york: Mcgraw-hill, 1988, 399 pp., approx. $47.95," *Applied Psychological Measurement*, vol. 13, no. 2, pp. 217–219, 1989. [Online]. Available: https://doi.org/10.1177/014662168901300212

[48] T. Villmann, R. Der, M. Herrmann, and T. M. Martinetz, "Topology preservation in self-organizing feature maps: exact definition and measurement," *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 256–266, 1997.

[49] A. König, "Konig, a.: Interactive visualization and analysis of hierarchical neural projections for data mining. ieee transactions on neural networks 11(3), 615-624," *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 11, pp. 615–24, 05 2000.

[50] J. Venna and S. Kaski, "Local multidimensional scaling," *Neural Networks*, vol. 19, no. 6, pp. 889–899, 2006, advances in Self Organising Maps - WSOM'05. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608006000724

[51] L. Chen and A. Buja, “Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis,” *Journal of the American Statistical Association*, vol. 104, pp. 209–219, 03 2009.

[52] R. Shepard, “The analysis of proximities: Multidimensional scaling with an unknown distance function. ii,” *Psychometrika*, vol. 27, no. 3, pp. 219–246, 1962. [Online]. Available: https://EconPapers.repec.org/RePEc:spr:psycho:v:27:y:1962:i:3:p:219-246

[53] J. Venna and S. Kaski, “Nonlinear dimensionality reduction as information retrieval,” in *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, M. Meila and X. Shen, Eds., vol. 2. San Juan, Puerto Rico: PMLR, 21–24 Mar 2007, pp. 572–579. [Online]. Available: http://proceedings.mlr.press/v2/venna07a.html

[54] J. A. Lee, E. Renard, G. Bernard, P. Dupont, and M. Verleysen, “Type 1 and 2 mixtures of kullback–leibler divergences as cost functions in dimensionality reduction based on similarity preservation,” *Neurocomputing*, vol. 112, pp. 92–108, 2013, advances in artificial neural networks, machine learning, and computational intelligence. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231213001471

[55] S. A. Nene, S. K. Nayar, and H. Murase, “Columbia Object Image Library (COIL-20),” Tech. Rep., Feb 1996.

[56] Y. LeCun and C. Cortes, “MNIST handwritten digit database,” 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

Appendices

5.1 Appendix 1

5.1.1 Python implementation of RNX curve

For the full featured nxcurve Python package you can visit the webpage https://pypi.org/project/nxcurve/.