London South Bank University
Division of Computer Science and Informatics
School of Engineering

Effective and Trustworthy Dimensionality Reduction Approaches for High Dimensional Data Understanding and Visualization

Laureta Hajderanj

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computer Science and Informatics at
London South Bank University
January 2022

Abstract

In recent years, the huge expansion of digital technologies has vastly increased the volume of data to be explored. Reducing the dimensionality of data is an essential step in data exploration and visualisation. The integrity of a dimensionality reduction technique relates to the goodness of maintaining the data structure. The visualisation of a low dimensional data that has not captured the high dimensional space data structure is untrustworthy. The scale of maintained data structure by a method depends on several factors, such as the type of data considered and tuning parameters. The type of the data includes linear and nonlinear data, and the tuning parameters include the number of neighbours and perplexity. In reality, most of the data under consideration are nonlinear, and the process to tune parameters could be costly since it depends on the number of data samples considered.

Currently, the existing dimensionality reduction approaches suffer from the following problems: 1) Only work well with linear data, 2) The scale of maintained data structure is related to the number of data samples considered, and/or 3) Tear problem and false neighbours problem.To deal with all the above-mentioned problems, this research has developed Same Degree Distribution (SDD), multi-SDD (MSDD) and parameter-free SDD approaches , that 1) Saves computational time because its tuning parameter does not 2) Produces more trustworthy visualisation by using degree-distribution that is smooth enough to capture local and global data structure, and 3) Does not suffer from tear and false neighbours problems due to using the same degree-distribution in the high and low dimensional spaces to calculate the similarities between data samples. The developed dimensionality reduction methods are tested with several popular synthetics and real datasets. The scale of the maintained data structure is evaluated using different quality metrics, i.e., Kendall's Tau coefficient, Trustworthiness, Continuity, LCMC, and Co-ranking matrix.

Also, the theoretical analysis of the impact of dissimilarity measure in structure capturing has been supported by simulations results conducted in two different datasets evaluated by Kendall's Tau and Co-ranking matrix.

The SDD, MSDD, and parameter-free SDD methods do not outperform other global methods such as Isomap in data with a large fraction of large pairwise distances, and it remains a further work task. Reducing the computational complexity is another objective for further work.

Acknowledgements

I would like to express my gratitude to:

My supervisor Daqing Chen for his huge and consistent support throughout this research project.

My second supervisor Sandra Dudley who has given a lot of assistance and support.

My colleagues, Souhel Fenghour and Isakh Weheliye, who has given a lot of advice and assistance during my project.

My family, especially my husband and my little son, for being my motivation through this project.

And finally, Acctive Systems Ltd and London South Bank University, for sponsoring me financially to undertake this research.

Acronyms

DR	Dimensionality Reduction
CCA	Curvlinear Commponent Analysis
CDA	Curvelinear Distance Analysis
DD-HDS	Data-Driven High-Dimensional Scaling
DP	Distance Preservation
ESLLE	Enhanced Supervised Locally Linear Embedding
GTE	Generative Topographic Embedding
HLLE	Hessian Locally-Linear Embedding
Isomap	Isometric Maps
KM	Konig's Measures
KPCA	Kernel PCA
KSM	Kruskal Stress Measure
LCMC	Local Continuity Meta-Criterion
LDA	Linear Discriminant Analysis
LE	Laplacian Eigenmaps
LLE	Locally Linear Embedding
LTSA	Local Tangent Space Alignment
MDS	Multidimensional Scaling
MLLE	Modified Locally Linear Embedding
MPV	Maximal Preserved Variance
MRE	Minimal Reconstruction Error
MRRE	Mean Relative Rank Error
MSDD	Multi Same Degree Distribution
MVU	Maximum Variance Unifolding
PCA	Principal Component Analysis
SDD	Same Degree Distribution
SLLE	Supervised Locally Linear Embedding
S_R	Spearman's Rho
T& C	Trustworthiness & Continuity
TCIE	Topologically Constrained Isometric Embedding
TP	Topology Preservation
T_{Pr}	Topological Product
t-SNE	t-Stochastic Neighbor Embedding
UMAP	Uniform Manifold Approximation and Projection

Symbols

N	Number of samples
n	Number of degree-distributions
D	Number of dimensions of the original data
d	Number of dimensions in low dimensions space data
$X^{N\times D}$	Original space dataset
$Y^{N\times d}$	Low dimensional space dataset
k	Number of neighbours
pr	Perplexity
$dis(a,b)$	Euclidean distance between data samples a and b
τ	Kendall's Tau
λ	User-defined parameter
λ_1	User-defined parameter in DD-HDS method
$K(.,.)$	Kernel function
σ	Density in Gaussian distribution
t	Timestep in Diffusion Maps

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, the huge expansion of digital technologies has vastly increased the volume of data to be explored. There are problems relating with high-dimensional data such as *the curse of dimensionality* and *concentration phenomenon*. Reducing the data dimensionality is an essential step in data analytics to overcome the problems relating to high dimensional data. Although many dimensionality reduction methods have been proposed, their performance in maintaining the high dimensional space data structure is related to the type of data, similarity function (i.e., Euclidean distances) used, and some tuning parameters[1] such as the number of neighbours, and perplexity. In other words, some dimensionality reduction methods do not require tuning parameters; however, they only perform well in linear data[2]. On the other hand, some other methods can perform well in nonlinear data[3], but their performance relates to parameter tuning that makes them costly methods. Also, the impact of a similarity functions[4] used in a dimensionality reduction method in structure maintaining requires further investigation.

This research aims to develop nonlinear dimensionality reduction methods that work well in any data (linear or complex nonlinear data) and do not require parameter tuning (i.e., the number

[1]Parameters include the parameters of a model such as the number of neighbours, perplexity, and does not include optimisation parameters such as learning rate.

[2]The low dimensional representation lies on linear manifolds.

[3]The low dimensional representation lies on nonlinear manifolds.

[4]A similarity function measures the similarity between data samples, i.e., Euclidean distance.

of neighbours or perplexity), making a dimensionality reduction method costly in terms of computational time and resources. Also, this research aims to investigate the impact of a similarity function on the scale of the maintained structure of high dimensional space data during the dimensionality reduction process.

This Chapter will introduce the research by first providing the technical background of the study, background information in the high dimensional space data and problems relating to their analysis, and presenting the problems of current dimensionality reduction methods in maintaining high dimensional space data structure. Also, this Chapter will provide research aims, objectives and questions, and finally, will demonstrate the main contributions of this research.

1.1 Research Background

1.1.1 Terminology and Definitions

Terminology and definitions used through the thesis are presented in this Section. This includes definitions of topological space, manifold, intrinsic dimensionality, linear or nonlinear dimensionality reduction methods, and order sets.

A topological space is a set for which topology is defined [1]. Suppose that Y is a set, and T is defined as a collection of subset Y that obey the following properties:

1. $\emptyset \in T$, $and Y \in T$.
2. If $A, B \in T$ then $A \cap B \in T$.
3. If $A, B \in T$ then $A \cup B \in T$.

From a geometrical viewpoint, a topological space can be defined using neighbourhoods and Haussdrof's axioms as follows:

1. To each point y there corresponds at least one neighbourhood $U(y)$, and $U(y)$ contains y.

2. If $U(y)$ and $V(y)$ are neighborhoods of the same point y, then a neighborhood $W(y)$ exists such that $W(y) \subset U(y) \cup V(y)$.

3. If $z \in U(y)$, then a neighborhood $V(z)$ of z exists such that $V(z) \subset U(y)$.

4. For two distinct points, two disjoint neighbourhoods of these points exist.

A manifold M^d, also known as topological manifold, is a topological space that is locally a Euclidean space, and it has to be a second countable space. A Euclidean space is a space with a finite number of dimensions, where coordinates present each sample (one per dimension). The distance between any two samples is calculated using the Pythagorean theorem, where the distance between the data sample a with n coordinates $(a_1, ..., a_n)$ and data sample b with n coordinates $(b_1, ..., b_n)$ is calculated using $\sqrt{(a_1 - b_1)^2 + ... + (a_n - b_n)^2}$, which corresponds to an Euclidean distance. According to [2], the Hausdorff and the second-countability conditions ensure that distinct samples on the manifold can be separated by their neighbourhoods and not large manifold, respectively. Let assume that a data set $\mathcal{X}$ resides on a d-manifold M that is embedded in $\mathcal{R}^D$ dimensional space. From statistics prospective, $\mathcal{X}$ can be considered as a sample set in $\mathcal{R}^D$. A random vector $Y \in \mathcal{R}^d$ and a function $f : \mathcal{R}^d \to \mathcal{R}^D$ such that $f(Y) = X$, then the random vector X is said to have intrinsic dimensions d.

The terms intrinsic dimensions refer to the number of latent variables. It reveals topological structure in data when the intrinsic dimension $d < D$, data points are constrained to lie in a well-delimited subspace. A low intrinsic dimension indicates that a topological object or structure underlines the data set.

Linear dimensionality reduction methods are based on the assumption that the observed dataset resides on linear manifolds. On nonlinear manifolds, a tangent space exists at each point of a nonlinear manifold, which locally approximates the manifold [3]. The tangent space is a linear manifold. A smooth manifold, also called an infinitely differentiable manifold, is a manifold with its functional structure (parametric equations), and it differs from standard topological manifold differentiability. A smooth manifold with boundaries are named submanifolds if the

following conditions are met: All points $y \in M$ there exist two open sets $U, V \subset M$ with $y \in U$, and a diffeomorphism $h : U \to V, y \mapsto x = h(y)$ such that $h(U \cap M) = V \cap (R^P \times 0)$. As can be seen,$x$ can be reduced to d- dimensional coordinates.

Order sets (or partially order sets) is a set S and a relation $\leq$ on the set S if the following properties are satisified:

1. Reflexivity: for all $a \in S$, $a \leq a$.

2. Antisymmetry: $a, b \in S$, $a \leq b$ and $b \leq a$, then $a = b$.

3. Transitivity: $a, b, c \in S$, $a \leq b$ and $b \leq c$, then $a \leq c$

1.1.2 Dimensionality Reduction

Fields of technologies or sciences where high dimensional space data are typically encountered include: processing of sensor data such as anomaly detection [4, 5, 6], fingerprint identification [7, 8], face recognition [9, 10], hyperspectral image analyses [11, 12, 13], text document classification [14, 15], speech recognition [16, 17], computer vision [18], neuroinformatics [19, 20], bioinformatics [21, 22], social media [23, 24], and telecoms [25]. High dimensional space data can be generated by arrays of antennas, electrodes recording time signals at different places on the chest or the scalp for biomedical applications, several stations or satellites deliver data weather forecasting, fingerprints, and hyperspectral image. An example of high dimensional data can be a digital fingerprint with a resolution of 64×64 that can be converted into a raw data sample with a dimension of $4,696$. Also, a face image resolution is 256×256, and a text document with a pre-defined dictionary of keywords consisting of $10,000$ words can be converted in raws with $65,536$ and $10,000$ dimensions, respectively.

From a practical viewpoint, high dimensional data is not always high dimensional, as the data analysis community has agreed that high dimensional data points reside on low-dimensional manifolds [3]. In other words, a large set of variables can be represented by a smaller one, with

no or less redundancy[5]. From a theoretical perspective, all difficulties that occurred when dealing with high dimensional data are related to the *curse of dimensionality* [26] and *concentration* phenomenons [27]. The term curse of dimensionality refers to the situation that to estimate a function with a certain accuracy, the number of data samples considered has to be exponentially higher than the number of variables accuracy [3]. The concentration phenomenon refers to the weak discrimination power of a metric that measures similarity between data samples. As the dimensionality of data increases, the similarity generated by the metric becomes less discriminant. In practice, the concentration phenomenon makes the nearest-neighbour search problem challenging in high dimensional space data. To avoid the problems mentioned above, embedding high dimensional space data into a lower dimensional space has been considered significantly recently. Ideally, an embedding method should be able to:

1. Estimate the intrinsic dimensionality,
2. Embed data to reduce their dimensionality, or
3. Embed data to recover latent variable.

Estimating the intrinsic dimensionality means counting the number of variables that describe data given a few samples. Intrinsic dimensionality is closely related to having a topological structure of data. If $d = D$, there is no data structure, whereas there is a data structure if $d < D$. Having that data structure means that the d-dimensional space data can represent the D- dimensional space data. Embedding data to reduce dimensionality refers to re-embedding D dimensional space data in d dimensional space with the aim to preserve the manifold structure of the high dimensional space data. The embedding techniques that aim to reduce the dimensionality of data to capture the structure of manifolds that contain the low dimensional representation are also called dimensionality reduction techniques. The fundamental assumption of a dimensionality reduction technique is that the high dimensional space data lies approximately on a manifold (often nonlinear) with lower dimensions than the original space data. Overall, a dimensionality reduction technique aims to find a representation of that manifold

[5]Redundancy means variables that are not independent of each other.

(a coordinate system) that will allow the high dimensional space data to be projected on a lower-dimensional representation, where the manifold structure has been preserved as good as possible. Some embedding methods aim to decrease the number of variables by keeping those statistically independent and removing the others dependent on each other.

Methods that aim to estimate the intrinsic dimensionalities and latent variables will be behind the scope of this research, and this research focuses on dimensionality reduction methods and, more specifically, on the visualisation of the high dimensional space data. The dimensionality reduction methods focusing on visualisation seek that the two- or three-dimensional space data represents the high dimensional space data by reviling their data structure.

To analyse a dimensionality reduction method, four following characteristics should be considered:

1. Type of data,
2. Criterion,
3. Similarity function, and
4. Parameters.

The type of data relates to the shape of the manifold, where the low dimensional representation of the high dimensional data lies. If the shape of a manifold is linear, then the data is known as *linear data*; otherwise, it is known as *nonliear data*. Furthermore, nonlinear data are also categorised in smooth and heavy-curved manifolds. Usually, a dimensionality reduction method assumes the data type. Some dimensionality reduction methods assume that data is linear, and some assume it is nonlinear but in smooth manifolds.

Criterion is also known as the cost function and relates to the purpose of using the embedding method. If the purpose of an embedding method is to preserve the data structure, it means maintaining the pairwise distances between data spaces. Then the criterion could be the mean square error between high and low dimensional spaces pairwise distances [3].

The similarity function calculates how similar or dissimilar two data samples are. Usually, standard dimensionality reduction methods use Euclidean distances to calculate the similarity between data samples. However, the change of the similarity function in standard dimensionality reduction methods leads to generating their supervised or nonlinear versions.

Parameters include the parameters involved in the dimensionality reduction model, i.e., in the similarity functions (number of neighbours, perplexity), and as a consequence, their structure depends on their value.

Although many dimensionality reduction techniques have been proposed, most of them have limitations in the type of data they can successfully be applied. Also, some dimensionality reduction methods assume that the low dimensional representation lies on linear or smooth manifolds, which makes them limited to application in real-world, complex nonlinear data[6]. On the other hand, some dimensionality reduction methods such as SNE, t-SNE [28], Umap [29] and Trimap [30] and DD-HDS [31] are able to maintain the data structure of complex nonlinear data due to calculating the similarity between data samples using Gaussian distribution. The key success of employing the Gaussian distribution is prioritising the maintenance of the close data samples than the faraway data samples, which is a limitation of other dimensionality reduction methods such as PCA, MDS, and Isomap. However, employing Gaussian distribution as a similarity measure leads to further problems: 1) methods that employ Gaussian distribution neglects the global data structure capturing, 2) their scale of maintained data structure is closely related to tuning the number of neighbours or perplexity, making them costly methods, especially when the number of data samples is considerably high, and 3) the Gaussian distribution is an expensive distribution due to using the *exp* function. Note that tuning the number of neighbours or perplexity ranges from 1 to the number of samples -1, and if the number of data samples considered is 10,000, then the number of neighbours or perplexity has to be tuned from 1 to 9,999. To deal with the costly procedure of tuning parameters, multiscale approaches such as multiscale-SNE have been proposed; however, this approach remains a costly method due to both the multiscale calculations and the utilisation of Gaussian distribution.

[6]High dimensional data, where their low dimensional representation lies on heavily curved manifolds.

In addition, this research has identified two common problems in maintaining the high dimensional space data structure, *tear*[7] and *false neighbours*[8] [32]. However, there is a lack of information on the leading causes of those problems.

In addition to dimensionality reduction techniques, this research has also been focused on their supervised versions. A supervised dimensionality reduction technique differs from its unsupervised version by using a dissimilarity measure instead of its standard metric. i.e., Euclidean distance, to calculate similarities between data samples. Supervised dimensionality reduction techniques have been proposed to increase the classification accuracy and capture better the structure of high dimensional space data. However, all work done in this field has been experimental-based, and there is a lack of theoretical foundation on the impact of dissimilarity measures on structure capturing and classification accuracy.

This research has defined two main research problems in the current literature related to 1) dimensionality reduction techniques, and 2) supervised dimensionality reduction techniques presented as follows:

1. Although there are many dimensionality reduction methods, their performances in terms of maintained data structure and computational time and resources is related to some other factors. The dimensionality reduction methods either work well in linear data and do not require parameter tuning (PCA and MDS) or perform well in nonlinear data but require parameter tuning that is a costly process. Since the data generated nowadays are mostly nonlinear, only nonlinear dimensionality reduction techniques (manifold learning techniques), can maintain the data structure. However, the scale of the maintained structure of high dimensional space data from nonlinear dimensionality methods relates to parameters, i.e., the number of neighbours (number of data samples -1), perplexity, or other parameters that require a lot of computational time and resources to be tunned. Consequently, applying the current nonlinear dimensionality reduction techniques to visualise high dimensional space data is significantly costly. Also, nonlinear dimensionality

[7] Close data samples in the original space embed far away in the low dimensional space.
[8] Faraway data samples in the high dimensional space embed close in the low dimensional space.

reduction methods that use distributions (i.e., Gaussian or Student-t) as similarity functions focus more on preserving the local than the global structure of the high dimensional space data. Note that visualising high dimensional data without capturing the local data structure is nonsense [33]. Also, two problems, such as tear and false neighbours, usually occur in dimensionality reduction methods; however, there lacks an investigation on the leading causes of those problems.

(a) Overall, this research has identified several problems with the current dimensionality reduction techniques, and it aims to develop a dimensionality reduction technique that:

i. works well in heavily curved manifolds,

ii. does not require parameter tuning,

iii. does not include a distribution that has exponential function (exp), and

iv. is not prone to tears and false neighbours problems.

2. Dissimilarity measures instead of Euclidean distance has been used in supervised dimensionality reduction methods to improve classification accuracy and visualise high dimensional data better in terms of data structure-preserving. However, there lacks theoretical analysis on the impact of dissimilarity measures on the scale of maintained data structure and classification error.

As a result, the literature review has identified that there lacks:

1. A dimensionality reduction is more trusty and less expensive in time and resources than the current dimensionality reduction methods, and

2. Theoretical foundation on whether the unsupervised or supervised dimensionality reduction methods can better maintain the high dimensional space data structure and generate a better classification-accuracy model.

1.2 Research Aims, Objectives, and Questions

Given the research problems, this research aims

1. To propose a dimensionality reduction technique that:
   (a) captures the structure of heavily curved manifold data,
   (b) does not require tuning number of neighbours, perplexity or other time-consuming parameters, and
   (c) does not suffer from tears and false neighbours problems.
2. To provide a theoretical and practical study on the impact of dissimilarity measures employed in a dimensionality reduction have on maintaining the high dimensional space data structure and generating a better classification-accuracy model.

To support the research aims, the main objective that needs to be achieved are:

1. To investigate the impact of the type of similarity function employed in a dimensionality reduction method to calculate the similarity between data samples and the scale of the maintained structure of the high dimensional space data structure.
2. To investigate the leading causes of the tear and false neighbours problems in dimensionality reduction methods.
3. To investigate the impact of parameter tuning on the scale of the maintained structure of the high dimensional space data.
4. To theoretically prove the impact of dissimilarity measures on the scale of the maintained structure of the high dimensional space data.
5. To theoretically prove the impact of dissimilarity measures on the classification error generated by a classification model in the low dimensional space produced by a supervised dimensionality reduction method.

6. To practically demonstrates the impact dissimilarity measures have on visualising interpretability and trustworthiness.

Based on the research aims and objectives, the main identified research questions are:

1. Does the similarity measure employed to calculate the similarities between high dimensional space data samples impact the maintained structure of the high dimensional data?
2. Which are the main leading causes of the problems of tear and false neighbours in dimensionality reduction techniques?
3. Is it possible that a nonlinear dimensionality reduction technique that does not require tuning the number of neighbours, perplexity or other time-consuming parameters captures the structure of nonlinear data (heavily curved manifold(s)) usefully?
4. Does dissimilarity measure employed to calculate the similarities between high dimensional space data samples impact the maintained structure of the high dimensional data?
5. Does dissimilarity measure used to calculate the similarities between high dimensional space data samples affect the classification model performances?

1.3 Research Contributions

The main contributions of this research are:

1. Developed a nonlinear dimensionality reduction technique named Same Degree Distribution (SDD) that captures the data structures better than other current dimensionality reduction methods in less computational time. SDD employs degree-distribution, which is the same as Student-t with degree 1, and for higher degrees, it has longer tails than Student-t. By using the degree-distribution, which has longer tails than Gaussian distribution, SDD ensures that it captures the global data structure better than other Gaussian-based methods and uses the same degree-distribution in high low dimensional

spaces; SDD ensures that tears and false neighbours problems are prevented. Also, SDD does not require tuning the number of neighbours, perplexity or other expensive parameters but instead, it requires tuning the degree, which usually ranges from 1 to 15.

2. Developed an extension of SDD named Multi Same Degree Distribution (MSDD) method to capture better the high dimensional space data structure than SDD. MSDD ensures that both the local and global structure of the data has been preserved by employing more than one degree-distribution and correspondingly more than one objective function that is optimised by a multi-objective optimisation method.

3. Developed a parameter-free same degree-distribution (parameter-free SDD) dimensionality reduction method that captures the same scale of data structure with SDD but does not require tuning the degree of distribution or any other parameter that makes parameter-free SDD a significantly low costly method.

4. This research has also analysed theoretically and practically the impact of dissimilarity measures on structure capturing and classification accuracy. This research generates two beneficial findings that clarify the true impact of dissimilarity measures on structure capturing of the high dimensional space data and classification model performance. Note that one of the conclusions of this research is completely different from what previous researchers in the field had claimed.

1.4 Publications

Most of the achieved contributions have been presented in publications listed as below:

- Peer reviewed journal papers

    1. Hajderanj, L., Chen, D., Dudley, S., Gilloppe, G., and Sivy, B., Novel Parameter-Free and Parametric Same Degree Distribution-based Dimensionality Reduction Algorithms for Trustworthy Data Structure Preserving.
    Status: Under Reviewing

2. Hajderanj, L., Chen, D. and Weheliye, I., 2021. The Impact of Supervised Manifold Learning on Structure Preserving and Classification Error: A Theoretical Study. IEEE Access, 9, pp.43909-43922.

3. Hajderanj, L., Chen, D., Grisan, E. and Dudley, S., 2020. Single-and multi-distribution dimensionality reduction approaches for a better data structure capturing. IEEE Access, 8, pp.207141-207155.

- Conference papers:

1. Fenghour, S.,Chen, D., Hajderanj, L., Weheliye, I., and P. Xiao, P., 2021. A novel Supervised t-SNE based approach of viseme classification for automated lip reading. IEEE International Conference on Electrical Computer and Energy Technologies. To appear in 2021 Proceedings of the IEEE International Conference on Electrical Computer and Energy.

2. Chen, D., Hajderanj, L., Mallet, S., Camenen, P., Li, B., Ren, H. and Zhao, E., 2021. Deep Learning Causal Attributions of Breast Cancer. In Intelligent Computing (pp. 124-135). Springer, Cham.

3. Chen, D., Hajderanj, L. and Fiske, J., 2019, July. Towards automated cost analysis, benchmarking and estimating in construction: a machine learning approach. In 13th Multi Conference on Computer Science and Information Systems (MCCSIS) (pp. 85-91).

4. Hajderanj, L., Weheliye, I. and Chen, D., 2019, April. A new supervised t-SNE with dissimilarity measure for effective data visualization and classification. In Proceedings of the 2019 8th International Conference on Software and Information Engineering (pp. 232-236).

1.5 Thesis Organisation

The thesis is organised into seven chapters.

Chapter 1, *Introduction* introduces the research, briefly identifying problems with high dimensional space data, identifying the research questions, aims and objectives, and providing the research contributions.

Chapter 2, *Literature Review* presents a comprehensive review of current dimensionality reduction techniques, emphasising the strengths and limitations of most dimensionality reduction techniques and identifying the main problems and gaps in the current literature. Chapter 2 also reviews the supervised dimensionality reduction techniques and the quality assessments of dimensionality reduction techniques and concludes by identifying the literature gap.

Chapter 3, *Methodology* presents the developed dimensionality reduction approaches by providing pseudocode and complexity analyses, followed by Chapter 4, *Experiments and Discussions on Developed Approaches* showing experimental results and discussions on the developed approaches.

The Impact of Dissimilarity Measures on Visualization and Classification Error has been presented in Chapter 5, followed by *Experiments on the Impact of Dissimilarity Measures on Structure Capturing* in Chapter 6.

A summary and conclusions of the thesis achievements are given in Chapter 7, *Conclusions*.

Chapter 2

Literature Review

The literature review provides a broader view of the existing embedding techniques that reduce data dimensionality, focusing on maintaining manifold data structure, and supervised dimensionality reduction techniques, focusing on the similarity measure employed (dissimilarity measure) and the main differences to their unsupervised versions. The literature review also includes methods that measure the quality of a dimensionality reduction method in terms of maintained data structure. This Chapter will be summarised by emphasising what is missing in the literature and why addressing them is essential.

2.1 Dimensionality Reduction Techniques

In this Section, dimensionality reduction techniques shown in Table 2.1 will be described more deeply in technical details. Dimensionality reduction methods considered will be compared based on the two main criteria:

1. *Parameters*[1] refers to the parameters that impact the performance type of the dimensionality reduction method.

[1]Parameter include all parameters that each technique has to tune to achieve the best by excluding optimisation parameters.

Table 2.1: DIMENSIONALITY REDUCTION METHODS

Year	DR algorithm	Parameters	Type of Data	References
1901	PCA	none	Linear	[34]
1962	MDS	none	Linear	[35]
1969	Sammon Mapping	none	nonlinear	[36]
1997	CCA	λ	nonlinear	[37]
1997	CDA	λ	nonlinear	[37]
1997	GTM	$K(.,.)$	nonlinear	[38]
1998	KPCA	$K(.,.)$	nonlinear	[39]
1998	SOM	σ ,v_λ	nonlinear	[40]
2000	Isomap	k	nonlinear	[41]
2000	LLE	k	nonlinear	[42]
2001	LE	k, σ	nonlinear	[43]
2003	HLLE	k	nonlinear	[44]
2004	MVU	k	nonlinear	[45]
2005	nonlinear PCA	$NetSize$	nonlinear	[46]
2005	LTSA	k	nonlinear	[47]
2006	Diffusion Maps	σ, t	nonlinear	[48]
2006	Autoencoders	$NetSize$	nonlinear	[49]
2007	MLLE	k	nonlinear	[50]
2007	DD-HDS	$\lambda 1$	nonlinear	[31]
2008	t-SNE	pr	nonlinear	[28]
2008	Manifold Sculpting	k	nonlinear	[51]
2009	RankVisu	k	nonlinear	[52]
2010	TCIE	k	nonlinear	[53]
2018	Trimap	k	nonlinear	[30]
2018	UMAP	k	nonlinear	[29]

2. *Type of data* refers to the shape of a manifold (linear or nonlinear) that contains the low dimensional representation of the original data.

2.1.1 Principal Component Analysis (PCA)

PCA is the most popular dimensionality reduction technique [3]. The low dimensional data $Y^{N\times d}$ is generated using the linear transformation $M^{D\times d}$, which is an orthogonal matrix: such that $M^T M = I_d$ and $MM^T = I_D$, where I_d and I_D are the identity matrix. A constrain of PCA is that both variables in high and low dimensional space are centred such that $E_y y = 0_d$ and $E_x x = 0_D$. In case where data $x \in X$ are not centred, then they can be centred by removing the expectation of x from each observation x as $x_i = x_i - E_x x$, where $E_x x = \frac{1}{N}\sum_{i=1}^{N} x(i)$.

This approach was proposed by Pearson [54], and it is composed of two stages, *coding* and *decoding*, where the coding and decoding functions are as follows:

$$cod : R^D \longrightarrow R^d, x \longrightarrow y = cod(x) = M^+x, \tag{2.1}$$

$$dec : R^d \longrightarrow R^D, y \longrightarrow x = cod(y) = My, \tag{2.2}$$

$$W^+ = W^T \tag{2.3}$$

where W^+ is the left pseudo-inverse of W.

The reconstruction error between original data and the linear transformation of the original data can be defined as:

$$E_{codec} = E_x\{\left\|x - MM^Tx\right\|_2^2\} \tag{2.4}$$

In an ideal situation, if the original data x has been generated ideally using PCA, and if $x = My$, then $MM^Tx = MM^TMy = MI_dy = x$, and the reconstruction error is zero. However, most of the data are nonlinear, such that M cannot be identified ideally, which then results in a nonzero reconstruction error. To find the best linear transformation of the original data is to minimize the reconstruction error:

$$E_{codec} = E_y\{\left\|x - MM^Tx\right\|_2^2\} = E_x\{x^Tx\} - E_x\{x^TMM^Tx\} \tag{2.5}$$

where $E_x\{x^Tx\}$ is constant and the minimization of E_{codec} end up to maximize the terms

$$E_x\{x^TMM^Tx\} \tag{2.6}$$

$$E_x\{x^TMM^Tx\} \approx \frac{1}{N}\sum_{i=1}^{N}(x(i))^TMM^T(x(i)) \approx \frac{1}{N}tr(X^TMM^TX) \tag{2.7}$$

where $tr(L)$ denostes the trace of a matrix L.

$$X = V\Sigma U^T, E_x\{x^T M M^T x\} \approx \frac{1}{N} tr(U\Sigma^T V^T M M^T V \Sigma U^T) \tag{2.8}$$

$$arg\max_M tr(U\Sigma^T V^T M M^T V \Sigma U^T) = V I_{D\times d} \tag{2.9}$$

since V, U are unitary matrixes and are orthonormal vectors by construction. Maximum of the expression shown in E.q (2.9) can be achieved when d columns of the matrix M are colinear with columns of the matrix V that are associated with the d largest singular values of Σ.

And finally, the low dimensional data can be generated using:

$$y = I_{d\times D} V^T x \tag{2.10}$$

.

PCA has many advantages as a dimensionality reduction method, such as it is a low computational time method and does not require tuning parameters. However, PCA applies a linear transformation of the data, assuming that high dimensional space data has a low dimensional representation in a linear manifold. As a result, PCA fails to maintain the data structure of high dimensional data if having the low dimensional representation in nonlinear manifolds. Also, suppose a given data has a high number of dimensions. In that case, PCA needs relatively many (probably more than three) latent variables to capture the variances of high dimensional space data. Consequently, PCA is not feasible for visualising high dimensional space data having considerably high dimensions.

2.1.2 Multidimensional Scaling (MDS)

Classical metric MDS preserves pairwise scalar products and achieves dimensionality reduction linearly, and it has been considered a motivation for all the nonlinear methods further

considered. The observed variables x can be calculated using:

$$x = My \tag{2.11}$$

, where y are independent and uncorrelated variables, and M is a $D \times d$ matrix such that $M^T M = I_d$. As mentioned above, classical metric MDS calculates the scalar product of S known as Gram metric, defined as:

$$S = Y^T Y \tag{2.12}$$

The low dimensional space variables can be obtained by calculating eigenvalue decomposition of S as:

$$S = U\Lambda U^T \tag{2.13}$$

$$= (U\Lambda^{\frac{1}{2}})(\Lambda^{\frac{1}{2}}U^T) \tag{2.14}$$

$$= (\Lambda^{\frac{1}{2}}U^T)^T(\Lambda^{\frac{1}{2}}U^T) \tag{2.15}$$

where U is $N \times N$ orthonormal matrix and Λ is a $N \times N$ diagonal matrix containing eigenvalues. The low dimensional variables are calculated as the product:

$$Y = I_{d\times N}\Lambda^{\frac{1}{2}}U^T \tag{2.16}$$

Metric MDS

Clasical metric MDS has been further replaced with metric MDS, which uses Euclidean pairwise distances as criterion instead of scalar production

$$E_{metricMDS} = \frac{1}{2}\sum_{i,j=1}^{N}(dis(x_i, x_j) - dis(y_i, y_j))^2 \tag{2.17}$$

Advantages and Limitations

MDS is equivalent to PCA [3] in terms of simplicity and robustness, and it does not require parameter tuning. Also, like PCA, MDS successfully applies to linear data, but it is a useless method in high dimensional data with low dimensional representation located in nonlinear manifolds. Because the metric MDS uses Euclidean pairwise distances and uses the reconstruction error in Eq. (2.17) to measure the reconstruction error between high and low dimensional space pairwise distances, it favours the maintenance of large distances since the cost function in Eq. (2.17) is impacted more by changes on the large distances than shorter ones. Consequently, MDS is less effective in cases when short distance maintenance is essential.

2.1.3 Sammon's mapping

Sammon's mapping is based on metric MDS, but it minimize the following criterion (cost function):

$$E_{Sammon'sMapping} = \frac{1}{\sum_{i,j=1}^{N} dis(x_i, x_j)} \sqrt{\frac{\sum_{i,j=1}^{N}((dis(x_i, x_j) - dis(y_i, y_j))^2}{\sum_{i,j=1}^{N}(dis(x_i, x_j))}} \tag{2.18}$$

Sammon's mapping uses $\frac{1}{dis(x_i,x_j)}$ to reduce the influence of the errors generated by large distances, which is a problem that occurred in MDS.

Advantages and Limitations

Sammon's mapping can effectively handle nonlinear manifolds, especially if they are not too heavy manifold(s) [3]. And it requires a lot of resources due to building the complete distance matrix. Furthermore, Sammon's Mapping addresses the problem of MDS by adapting the weight scaling to reduce the impact of large distances and increasing the effect of short distances in the cost function. Consequently, boosting the contribution of very close points in the cost function (Eq. (2.18)) is the main weakness of Sammon's mapping.

2.1.4 Curvilinear component analysis (CCA)

CCA is similar to Sammon's mapping; however, it minimises the cost function in Eq. (2.19).

$$E_{CCA} = \frac{1}{2}\sum_{i,j=1}^{N}(dis(x_i, x_j) - dis(y_i, y_j))^2 F_\lambda(dis(y_i, y_j)) \tag{2.19}$$

CCA and Sammon's mapping changes in two ways:

1. no scaling factor, and
2. $\frac{1}{dis(x_i,x_j)}$ is replaced with $F_\lambda(dis(y_i, y_j))$.

The use of F_λ is to prioritise the preservation of short distances over larger ones. The main focus of CCA is to unfold the manifold, such that large distances have to be stretched, and their contribution in the stress function is low, thanks to using the function F_λ. Usually, the effect of the large distances in the cost function is huge, and it can be minimised by employing F_λ. However, F_λ depends on distances of the low dimensional space data, which are usually very small. When pairwise distances of high and low dimensional spaces are equal, CCA and Sammon's mapping behaves the same. However, in other scenarios, when $dis(y_i, y_j) \ll dis(x_i, x_j)$ and $dis(y_i, y_j) \gg dis(x_i, x_j)$, the two reduction methods behaves totally different from each other. If $dis(y_i, y_j) \ll dis(x_i, x_j)$, the manifold has been highly unfolded and F_λ helps to correct the flaw. In another scenario, $(dis(y_i, y_j) \gg dis(x_i, x_j))$, than the contribution of F_λ will be decreased, meaning that stretching on large and short distances will be occurred.

$$F_\lambda = H(\lambda - dis(y, y)) \tag{2.20}$$

where $H(u)$ has been defined as:

$$H(u) = \begin{cases} 0 \;\; if\, u \leq 0 \\ 1 \;\; if\, u \geq 0 \end{cases} \tag{2.21}$$

Advantages and Limitations

The main limitation of CCA is the choice of λ, which affects the method's performance. F_λ depends on distances of the embedded space, then in some cases, it allows tearing some regions of the manifold, but it can be better than Sammon's mapping [3].

In general, Sammon's mapping and CCA are variants of MDS, which makes modifications in the cost function to maintain a nonlinear data structure. However, a measure that better represents the true structure of high dimensional space data has been considered instead of modifying the cost function. Geodesic distance has further been considered a better measure than Euclidean distance for calculating the similarity of nonlinear high dimensional space data. Isomap and Curvilinear Distance Analysis (CDA) are the two methods that employ Geodesic

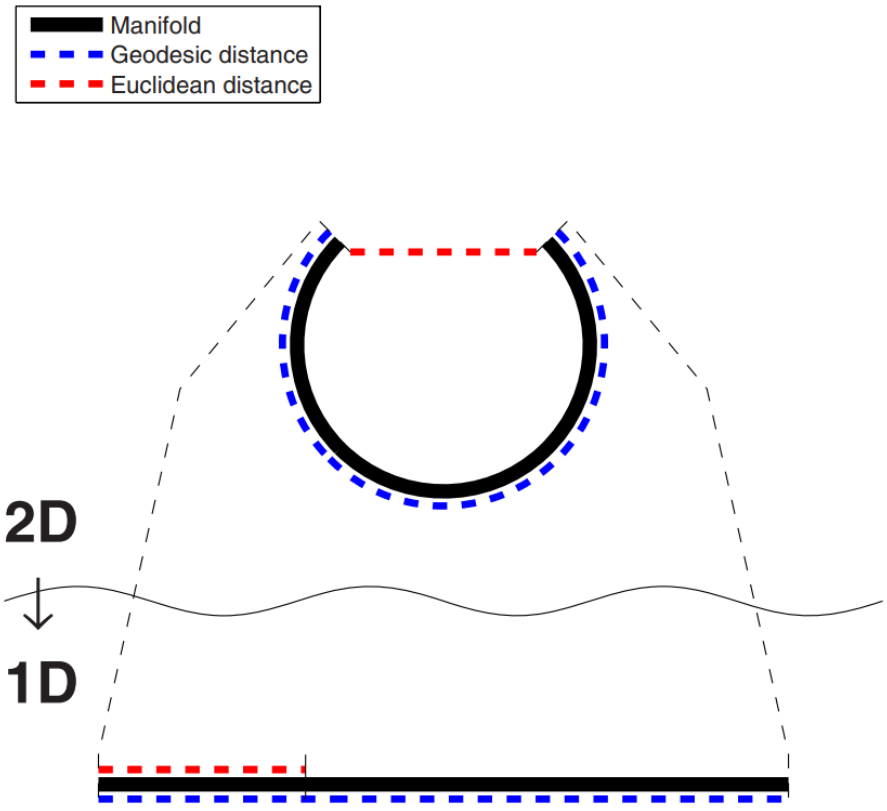


Figure 2.1: Manifold and distances [3].

distance instead of Euclidean distance to measure the similarity between data samples in high dimensional space.

2.1.5 Isomap

Isomap exploits high dimensional space nonlinear data geometry by employing the Geodesic distance. Geodesic distance is computed as the sum of the shortest path between two data samples in the neighbourhood graph. Isomap is similar to metric MDS, and the only difference between them is that metric MDS use Euclidean distance to calculate the pairwise distances matrix, whereas Isomap uses Geodesic distance. Because of using graph distances to calculate the pairwise distances, Isomap is a nonlinear dimensionality reduction method [3]. Indifference to Sammon's mapping and CCA, Isomap uses graph distance to make the technique nonlinear. In contrast, Sammon's mapping and CCA modify the optimization function, which can be more complex due to tunning parameters. It is assumed that the graph distances perfectly approximate the true Geodesic distance for theory purposes. On the other hand, it is also assumed that d-manifold is a *a developable d- manifold*[2]. As such, having a good performance of Isomap, the high dimensional space data must have the low dimensional space data lies on a *developable manifold*. To check if a manifold is developable, the Jacobian matrix of a developable manifold must be a $D \times d$ matrix whose columns are orthogonal vectors with unit norms.

$$J_X m(y) = QV(x) \tag{2.22}$$

where Q is a constant orthonormal matrix and $V(x)$ is a $D \times d$ matrix with unit-norm columns and only one nonzero entry per row. Furthermore, a d -space developable manifold embedded

[2]A manifold is developable if and only if a diffeomorphism between d- manifold and a convex subset of d -dimensional Euclidean space exist in such way that the Geodesic distances are mapped to the Euclidean distances by an identity map.

in a D dimensional space can be written as follows:

$$x = Qf(y) = \begin{bmatrix} f_1(y_1 \leq d_1 \leq d) \\ \vdots \\ f_i(y_1 \leq d_1 \leq d) \\ \vdots \\ f_D(y_1 \leq d_1 \leq d) \end{bmatrix} \tag{2.23}$$

where Q is the same as above, $J_x f(x) = V(x)$, and $f_1, f_2, ..., f_D$ are constant, linear or nonlinar functions from $\mathbb{R}$ to$\mathbb{R}$. The conditions on the functions $f_i(y_1 \leq_1 \leq d)$ are important to obtain a Jacobian matrix with orthogonal columns. Visually, a manifold is developable in three-dimensional space if it is a curved sheet of paper. A sphere or a piece of hollow is not developable, and if the manifold has a hole, it is not also developable.

Advantages and Limitations

Isomap successfully can be applied to a developable manifold. In nondevelopable manifolds, Isomap is prone to tear and false neighbours, which occurs because using two different similarity measures in the high and low dimensional space, to calculate the similarity of data samples. An identity map cannot embed Geodesic distances in a Euclidean distance. As a result, using Isomap is the same as applying linear dimensionality reduction methods such as PCA or MDS in nondevelopable manifolds. Another problem of Isomap that calculates the Geodesic distance is to approximate the graph distances. Their quality depends on the data itself and the parameters (number of neighbours k or a threshold ϵ for building the graph). Consequently, parameter selection hugely impacts the performance of a dimensionality reduction method [3].

2.1.6 Geodesic Sammon's mapping

Geodesic Sammon's mapping is like Sammon's mapping; however, it uses Geodesic distances instead of Euclidean distances to calculate the similarity between data samples in the high

dimensional space.

Advantages and Limitations

Geodesic Sammon's mapping has the same advantages and drawbacks as Sammon's mapping, but it can better deal with heavily curved manifolds [3]. On the other hand, using Geodesic distances (graph distances) requires tuning the number of neighbours, making Geodesic Sammon's mapping a more expensive method than Sammon's mapping.

2.1.7 Curvlinear distance analysis (CDA)

CDA is similar to CCA, but it calculates the similarity between data samples using Geodesic distances instead of Euclidean distances. Consequently, CDA captures the manifold shapes better than CCA and removes the shortcuts generated by Euclidean distances. Using Geodesic distances can be beneficial since the manifolds embedded in a high dimensional space data might be manifold(s) on themself, and Euclidean distances are not appropriate to describe the structure of a manifold.

Advantages and Limitations

CDA performs better in nonlinear manifolds (developable) than CCA. However, like Isomap, CDA suffers from tear and false neighbours problems since the similarity between high and low dimensional space data samples have been calculated using two different similarity measures.

2.1.8 Kernel Principal Component Analysis (KPCA)

KPCA uses kernel function over pairwise distances, and it is more similar to metric MDS than PCA [3]. The main idea of KPCA is to linearize the manifold M, and it supposes a mapping that a linear subspace can be mapped in a nonlinear subspace with dimensionality higher than

the previous one.

$$\phi : M \subset R^D \mapsto R^Q, x \mapsto z = \phi(x) \tag{2.24}$$

So, KPCA starts with increasing data dimensionality, and then compute the matrix

$$\Phi = [\langle \phi(x(i)) \cdot \phi(x(j)) \rangle]_{1 \leqslant i,j \leqslant N} \tag{2.25}$$

$$= [\langle z(i) \cdot z(j))) \rangle]_{1 \leqslant i,j \leqslant N} \tag{2.26}$$

After Φ has been defined, than the procedure is the same as for metric MDS.

$$\Phi = U \Lambda U^T \tag{2.27}$$

$$Y = I_{d \times N} \Lambda^{\frac{1}{2}} U^T \tag{2.28}$$

There exist some kernel functions:

1. Polinomial Kernel :

$$\kappa(u, v) = (\langle u \cdot v \rangle + 1)^{int} \tag{2.29}$$

2. Gaussian kernels:

$$\kappa(u, v) = exp\left(-\frac{\|u - v\|^2}{2\sigma^2}\right) \tag{2.30}$$

3. MLP Kernel :

$$\kappa(u, v) = tanh(\langle u \cdot v \rangle + b) \tag{2.31}$$

Advantages and Limitations

KPCA is considered an extension of MDS, but it better captures the nonlinear high dimensional data structure than MDS. However, choosing the proper kernel function and its parameter is the main difficulty of KPCA.

2.1.9 Maximum Variance Unfolding (MVU)

One of the main disadvantages of KPCA is choosing the right kernel function, which is solved by MVU. MVU tries to maintain high dimensional space data structure by constructing the graph, where each data sample is connected with its nearest neighbours and imposes the preservation of angles and distances for all k neighbourhoods as in Eq. (2.32):

$$\left\langle (x_i - x_j) \cdot (x_i - x_k))) \right\rangle = \left\langle (y_i - y_j) \cdot (y_i - y_k) \right\rangle \tag{2.32}$$

Let be A the $N \times N$ adjacency matrix of this graph, and then the local constraint can be expressed as:

$$\left\| y_i - y_j \right\|_2^2 = \left\| x_i - x_j \right\|_2^2 \;\; if \;\; A_{ij} = 1 \tag{2.33}$$

MVU uses the objective function:

$$\phi = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} dis_y^2(i,j) \tag{2.34}$$

All edges are subject to the local isometry constrained, it results that,

$$\phi \leq \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} graph_x^2(i,j) \tag{2.35}$$

where graph $dis_x^2(i,j)$ is the graph distance between data points x_i and x_j. The formulation of the problem can be simplified by using the dot products:

$$\phi = tr(L) \tag{2.36}$$

where

$$\phi = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} dis_y^2(i,j) \tag{2.37}$$

,and

$$\phi = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} s_x(i,j) - 2s_x(i,j) + s_x(j,j) = \sum_{i=1}^{N} s_x(i,i) = tr(L) \tag{2.38}$$

, where $tr(L)$ is the trace of L and $L = [s_x(i,j)]_{1\leq,j\leq N}$, and

$$s_x(i,j) = s_y(i,j) \;\; if \;\; a(i,j) = 1 \tag{2.39}$$

Overall, the goal of MVU consists of maximizing the trace of some $N \times N$ matrix L subject to the following constraints:

1. The matrix L is symmetric and positive semidefinite.

2. The sum of all entries of L is zero.

3. For nonzero entries of the adjacency matrix, the quality $s_x(i,j) = s_y(i,j)$ must hold.

MVU may overcome some shortcomings of Isomap, such as:

1. Isomap performs low in estimating Geodesic distances of sparse data.

2. Isomap fails to embed correctly nonconvex manifolds (manifolds with holes): in this case, graph paths are longer than necessary because they need to go around the hole.

Advantages and Limitations

MVU, like Isomap, suffers from the short-circuiting problem as it adds constraints to the optimization problem that may affect the manifold unfolding performance. Moreover, MVU is very slow due to Semidefinite Programming problem (SDP) [3]. Also, the scale of the maintained structure of the high dimensional space data depends on the number of neighbours, in which its tuning is a costly process.

All nonlinear methods mentioned above reduce the data dimensionality intending to minimize the difference between high and low dimensional space data. However, most real data have their low dimensional representation placed on curved and nonlinear manifolds. A dimensionality reduction technique requires some flexibility during the dimensionality reduction process since some regions need to be shrunk or stretched. In other words, maintaining the high dimensional data structure means maintaining the topology of the manifold (i.e., the neighbourhood relationships between subregions of a manifold) where the low dimensional representation lies. There are many dimensionality reduction methods focused on topology preservation categorized as *Predefined lattice methods* and *Data-Driven lattice methods*.

Predefined lattice methods

Predefined lattice methods are methods that define lattice in advance. In other words, the shape of the manifold is predefined. This makes the applicability of this method very limited as a few data may have the same manifold as the predefined one. Some predefined lattice methods are Self -Organizing Maps (SOM) and Generative Topographic Mapping (GTM).

2.1.10 Self-Organizing Maps (SOM)

SOM consists of the following steps:

1. A set of $c(i) \in C$ data points with dimension D

2. A function $d_g(r, s)$ which defined the neighbourhood relationship between prototipes c_i.

Prototypes have coordinates in the original and the low dimensional spaces, where the points of the low dimensional space are known in advance. Still, the corresponding data $c(i)$ in the original space are unknown and must be determined by SOM. Since $c(i)$ is determined, than, the low dimensional data samples are calculated as:

$$y(i) = g(r) \tag{2.40}$$

, where $r = arg\min_s d(y(i), c(s))$, where d is the distance function and usually is Euclidean distance. $c(s)$ can be determined iteratively by following the Robbins-Monro scheme.

$$c(s) \leftarrow c(s) + \alpha v_\lambda(r, s)(x(i) - c(s)) \tag{2.41}$$

, where α is the learning rate and $\alpha \in [0, 1]$, whereas $v_\lambda(r, s)$ can be dettermined in different forms

1. $v_\lambda(r, s) = \begin{cases} 0 \ if \ d_g(r, s) > \lambda \\ 1 \ if \ d_g(r, s) \leq \lambda \end{cases}$
2. $v_\lambda(r, s) = exp(-\frac{d_g^2(r,s)}{2\sigma^2})$
3. $v_\lambda(r, s) = \begin{cases} 0 \ if \ r \neq s \\ 1 \ if \ r = s \end{cases}$

where σ replaces the standard deviation. Data points $g(r)$ are placed on a plane in most implementations. The global shape of a lattice is often a rectangle or a hexagon (or a parallelepiped in higher dimensions). Neighbourhood shapes employed are square (eight neighbours)—Hexagonal (six neighbours), and (hyper)-cubic neighbourhoods in higher-dimensional lattices. As mentioned, the performance of SOM depends on a number of parameters such as:

1. Lattice shape (width, height, additional dimensions),
2. Neighbourhood shape (square, hexagon),

3. The neighbourhood function v_λ, and

4. Learning rate α and neighbourhood width σ.

Advantages and Limitations

The main drawback of SOM is that it predefines the shape of lattice in advance and does not capture the true structure of data. Also, SOM requires tuning parameters, which is a tedious task.

2.1.11 Generative Topographic Mapping (GTM)

GTM provides a generative model and calculates the probability of embedding at coordinates Y in the low dimensional space. GTM starts with initializing low dimensional data samples $g(r)$ and then computes the squared distances matrix

$$D = [\left\|x(i) - W\theta(g(r))\right\|^2] \tag{2.42}$$

where W is initiliazed randomly or using PCA, and Θ is the value of basis functions at the low dimensional data samples $g(r)$. The low dimensional space data are computed using d

$$y(i) = arg\max_{g(r)} p(g(r)|x(i)) \tag{2.43}$$

or

$$y(i) = \sum_{r=1}^{C} g(r)p(g(r)|x(i)) \tag{2.44}$$

where

$$p(g(r)|x(i)) = \rho_{i,r}(W_{opt}, \beta_{opt}) \tag{2.45}$$

$$\rho_{i,r}(W_{opt}, \beta_{opt}) = \frac{p(x(i)|g(r), W, \beta)}{\sum_{s=1}^{C} p(x(i)|g(s), W, \beta)} \tag{2.46}$$

$$p(x(i)|y, W, \beta) = (\frac{\beta}{2\pi})^{\frac{D}{2}} exp(-\frac{\beta}{2} \left\| x(i) - m(y, W)) \right\|^2) \tag{2.47}$$

and C is the set of representative data points with D dimensions.

Advantages and Limitations

Like SOM, GTM also predefines the shape of lattice and is limited to one or two low dimensional spaces. GTM does not perform well in data with significantly high dimensional space for the two main reasons: 1) it requires increasing the number of kernels defining m together with increasing the number of grid data points, and 2) the Gaussian kernel function employed in GTM behaves surprisingly in the high-dimensional space data [3].

Data-driven lattice methods

Data-driven lattice methods, indifferent from predefined lattice methods, do not use a predefined topology of the manifold, but they use the topology of the manifold itself and try to keep its structure accordingly. The data-driven lattice methods are LLE, LE, LTSA, Hessian LLE, Diffusion Map, t-SNE, UMAP, TriMap, DD-HDS, and RankVisu.

2.1.12 Locally Linear Embedding (LLE)

LLE is one of the methods known as data-driven lattice methods, and it starts determining which angle to consider by accounting k nearest neighbours of each data sample x_i. If the dataset is large and without noise and the manifold is well-sampled, then the manifold of k neighbourhood is approximately linear. The main stage of LLE [55] is the one that replaces

each data sample x_i with the linear combination of its neighbours, and the reconstruction error is measured using the following formula:

$$\varepsilon(W) = \sum_{i=1}^{N} \left\| x(i) - \sum_{j \in N(i)} w_{i,j} x(j) \right\| \tag{2.48}$$

where $N(i)$ is the set containing all neighbours of a data sample $x(i)$, and W is the matrix with $N \times N$ whose entries are the weight of neighbours for the reconstruction of the data sample $x(i)$. To compute the matrix W, the cost function needs to be minimized under two constrains:

1. coefficients $w_{i,j} = 0$ for points $j \notin N(i)$ and,

2. $\sum_{j=1}^{N} w_{i,j} = 1$

The weights $w_{i,j}$ reflects the intrinsic geometry of the data [3] that are invariant exactly to these transformations. So, the characterisation of the geometry in the original data space is expected to be equally valid for local patches on the manifold. Accordingly, the same weights that reconstruct data samples in the high-dimensional space D also reconstruct its manifold coordinates in a d-coordinate space. Data samples in the low dimensional space are chosen to minimise the $\Phi(Y)$ cost function.

$$\Phi(Y) = \sum_{i=1}^{N} \left\| y(i) - \sum_{j \in N(i)} w_{i,j} y(j) \right\| \tag{2.49}$$

Y is the d dimensional data that best reconstruct X given W. In practice, $w_{i,j}$ can be computed in closed form, for each data sample $x(i)$, separately.

$$\varepsilon_i(W) = \sum_{i=1}^{N} \left\| x(i) - \sum_{j \in N(i)} w_{i,j} x(j) \right\| \tag{2.50}$$

can be reformulated as:

$$\varepsilon_i(w(i)) = \left\| x(i) - \sum_{r=1}^{k} w_r(i)v(r) \right\|^2 \tag{2.51}$$

$$= \left\| \sum_{j=1}^{k} w_r(i)x(i) - v(r) \right\|^2 \tag{2.52}$$

$$= \left\| \sum_{r,s=1}^{k} w_r(i)w_s(i)g_{r,s}(i)) \right\|^2 \tag{2.53}$$

where $w(i)$ is the vector that contains the $i-th$ raw of W and $v(r)$ is the r-th neighbour of $x(i)$, corresponding to $x(j)$. There is also that $\sum_{r=1}^{K} w_r(i) = 1$ and $g_{r,s}(i) = (y_i - v(r))^T(y_i - v(s))$. The matrix $G(i)$ can be interpreted as a kind of local covariance of $x(i)$. The reconstruction error can be minimized in closed form, using Lagrange multiplier and the optimal weights.

$$w_r(i) = \frac{\sum_{s=1}^{k}(G^{-1}(i))_{r,s}}{\sum_{r,s=1}^{k}(G^{-1}(i))_{r,s}} \tag{2.54}$$

$G(i)$ is symmetric and semi-definite.

Minimisation of $\Phi(Y)$ can be done by solving an eigenproblem.

$$\Phi(Y) = \sum_{i=1}^{N} \left\| y(i) - \sum_{j \in N(i)} w_{i,j}y(j) \right\| \tag{2.55}$$

$$= \sum_{i=1}^{N} \left\| \sum_{j \in N(i)} w_{i,j}y(i) - y(j) \right\| \tag{2.56}$$

$$= \sum_{i=1}^{N} \left\| \sum_{j \in N(i)} m_{i,j}y(i)^T y(j) \right\| \tag{2.57}$$

where $m_{i,j}$ are the entries of an $N \times N$ matrix M, $M = (I - W)^T(I - W)$, which is symmetric, sparse and positive semi-definite. The optimal embedding has been found by the bottom $d+1$ eigenvectors of the matrix M, where the last eigenvectors have been discarded by keeping d eigenvectors representing the d dimensional space coordinates of Y.

Advantages and Limitations

The structure capturing of LLE is related to the number of neighbours k. If k is large, then the method approximates the manifold to be linear, which might not be a good representation for most datasets.

2.1.13 Local Tangent Space Alignment (LTSA)

LTSA is a technique that describes the local properties of high dimensional data using the local tangent of each data point. The main idea of LTSA is that if the local linearity of the manifold is assumed, there is a linear mapping from high dimensional space data to its local tangent space and a linear mapping from low-dimensional space data to the same local tangent space. LTSA simultaneously searches for the low-dimensional data representations' coordinates and the linear mappings of the low-dimensional data samples.

LTSA starts with graph $G = [X, A]$ construction considering neighbourhood size k or ϵ-ball neighbourhood is the first stage of LTSA. The second step is calculation of local coordinates. Let be $X_i = [x_{i1}...x_{ik}]$ whose columns are the k neighbours of x_i. Data are centralized by subtracting their mean: $X_i = [x_{i1} - \bar{x}...x_{ik} - \bar{x}]$. The local coordinated can be found by PCA and then the best approximation of X_i be

$$\sum_{j=1}^{d} \sigma_j u_j (v_j)' \tag{2.58}$$

Write $V = [v_1...v_d]$ and set $G_i = [\,1|V_i]$ then $W_i = I - G_i G_i'$. The third step of LTSA is kernel construction via global alignment. The LTSA kernel K is the alignment matrix of all local matrices W_i.

$$K(N(i), N(i)) = I - G_i G_i' \tag{2.59}$$

Te last step of LTSA is Eigen decomposition of kernel K.

$$K = U\Lambda U' \tag{2.60}$$

where $\Lambda = diag(\lambda_0, \lambda_1, ..., \lambda_n - 1)$ with $0 = \lambda_0 \leq \lambda_1 \leq ... \leq \lambda_{n-1}$. The low dimensional data $Y = [u_1...u_d]'$ corresponding to the $2nd - (d+1)$ smallest eigenvalues of K.

Advantages and Limitations

The main problem of LTSA is its sensitivity to noise, which impacts its performance. Neighbourhood size also affects the performance. So, LTSA usually performs well in smooth and connected manifolds. The smoothness of the manifold and selecting the neighbourhood size impacts the performance of LTSA.

2.1.14 Laplacian Eigenmaps (LE)

LE has been invented to overcome some shortcoming of Isomap by concentrating on local distances. LE also assume that data samples lie on a smooth d- manifold. For a large N, the underlined manifold can be represented very good by a graph $G = (V_N, E)$. The neighbourhood relationship can be determined using k neighbours or ϵ ball neighbourhoods. The aim of LE is to keep the neighbourhood relationship and minimize the E_{LE} as shown in Eq. (1.61).

$$E_{LE} = \frac{1}{2}\sum_{j=1}^{N} \left\|y(i) - y(j)\right\|_2^2 w_{i,j} \tag{2.61}$$

where

$$w_{ij} = \left\{ \begin{array}{c} exp(-\frac{dis(x_i,x_j)^2)}{2\sigma^2}) \; if \; x_j \in Neig_i \\ 0 \; otherwise \end{array} \right\} \tag{2.62}$$

$$E_{LE} = tr(YLY^T) \tag{2.63}$$

and L is the weighted Laplacian matrix of the graph G, defined as

$$L = W - D \tag{2.64}$$

and D is a diagonal matrix with entries $d_{i,i} = \sum_{j=1}^{N} w_{i,j}$ Minimization of E_{LE} with respect to Y under the constrain $YDY^T = I_{d\times d}$ reduces to solving the generalized eigenvalues problem $\lambda Df = Lf$ and looking for d eigenvectors of L associated with smallest eigenvalues. Then normalize the Laplacian matrix L as

$$L' = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \tag{2.65}$$

and then compute the

$$L' = U\Lambda U^T \tag{2.66}$$

The low dimensional data can be generated by multiplying eigenvectors by $D^{\frac{1}{2}}$, transposing them and keeping them associated with d eigenvalues, except the last one, which is zero.

Advantages and Limitations

LE cannot preserve the local Euclidean distances since this method is not very good in short distance preservation. LE can be flexible with moderate noise data; however, when noise is intense, then LE becomes unstable. Also, the performance of LE dramatically depends on the number of neighbours tuning and kernel function that is used for generating the weight matrix W.

2.1.15 Hessian LLE

HLLE is a variant of LLE that minimises the curviness of a high dimensional manifold when embedding it into a low dimensional space but to satisfy the constrain that the low dimensional data is locally isometric. Similar to LLE, the first step of HLLE is defining the neighbourhood system using k or ϵ -ball, where $k \geq \frac{(d+2)(d+1)}{2}$. The second step is calculating the coordinates of tangent space on its neighbourhood using PCA. Let be $LX^i = x_{j1}, ..., x_{jk}$ then apply PCA on LX^i to obtain d principal components of LX^i in the $k \times d$ matrix $V^i = [v_1 ... v_d]$, then the columns of V^i are tangent coordinate functions on LX^i. Local Hessian functional construction is the third step of HLLE. Define with:

$$V^a = [1, V^i, Q^i] \tag{2.67}$$

where $Q^i = [v_i \boxtimes v_j]_{1 \leq i \leq j \leq d}$ and $1 = [1, ..., 1]' \in R^k$. Apply the Gram-Schmidt procedure over V^a to obtain its orthonormalization $[1, V^i, \check{Q}^i]$, and then the Hessian functional is $W_i = \check{Q}^i(\check{Q}^i)'$. Initialize the kernel K to an $n \times n$ zero matrix and then update $K(N(i), N(i)) = K(N(i), N(i)) + W_i$, where $K(N(i), N(i))$ is the submatrix containing both the rows and columns with indices in $N(i)$. Finally, let be $Y^0, Y^1, ..., Y^d$ are $d+1$ eigenvectors corresponding to the $d+1$ smallest ascending eigenvalues of K and the low dimensional dataset is $Y = [Y^1, ..., Y^d]'$.

Advantages and Limitations

HLLE minimises a Hessian function, which is defined in a smooth manifold, and in nonsmoothed manifolds, HLLE performs poorly. HLLE adopts a local isometric manifold coordinate mapping, captures well the local Euclidean distances [3]. Furthermore, HLLE may not preserve the data structure very well when noise exists, as the data has a large deviation that distorts the smoothness. Another drawback of HLLE is that performance is based on the neighbourhood size.

2.1.16 Diffusion Maps

The diffusion maps aim to embed the dataset into a Euclidean distance. The Euclidean distance in the low dimensional space is equal to the diffusion distance on the data. The first step of Diffusion Maps is building the graph $G(X, A)$ neighbour giving k- or ϵ-ball.

$$g_{ij} = e^{\frac{dis(x_i,x_j)^2}{t}} \tag{2.68}$$

and

$$w_{ij} = \begin{Bmatrix} g_{ij} & g_{ij} \leq \tau \\ 0 & g_{ij} \geq \tau \end{Bmatrix} \tag{2.69}$$

The Diffusion Kernel has been constructed using $k_{ij} = \frac{w_{ij}}{v_i v_j}$, where $v_i = \sqrt{W^i 1}$ and W^i is the ith row of W. And the low dimensional representation is

$$Y = [\tilde{v_1}, ..., \tilde{v_d}]^T \tag{2.70}$$

where

$$\tilde{v_i} = \left[\frac{v_{1i}}{v_{10}}, ..., \frac{v_{ni}}{v_{n0}}\right], \;\; 1 \leq i \leq d. \tag{2.71}$$

and $v^0, ..., v^d$ are eigenvectors of K achieving the $(d+1)$ eigenvalues $1 = \lambda_0, ..., \lambda_d > 0$.

Advantages and Limitations

Like MDS, Diffusion Maps favours the preservation of large distances at the expense of neglecting small distances. Additionally, its structure capturing is related to tuning the parameter t [3]. Diffusion maps are sensitive to the neighbourhood size as well. Diffusion maps employ diffusion processing, making them insensitive to noise [3].

2.1.17 Stochastic Neighbour Embedding (SNE)

What happens if the low dimensional representation is placed in different manifolds? All mentioned methods can unfold submanifolds located in one place. This can be done by proposing probable neighbours employed to SNE [56] to preserve the neighbourhood identity, even when the low-dimensional representation is placed in different manifolds.

SNE, for each data sample $x(i)$ and a potential neighbour $x(j)$ calculates asymmetric probability p_{ij}

$$p_{ij} = \frac{exp(-dis(x_i, x_j)^2)}{\sum_{k \neq i} exp(-dis(x_i, x_k)^2)} \tag{2.72}$$

$$dis(x_i, x_j) = \frac{\left\| x_i - x_j \right\|^2}{2\sigma_i^2} \tag{2.73}$$

where σ_i is the value that makes the entropy of the distribution over neighbours equal to $\log k$, and k is the number of neighbour (k) or *perplexity* (pr) chosen by the user. In the low dimensional space, the neighbourhoods have been calculated using Gaussian distribution with a fixed variance:

$$q_{ij} = \frac{exp(-\left\| y_i - y_j \right\|^2)}{\sum_{k \neq i} exp(-\left\| y_i - y_k \right\|^2)} \tag{2.74}$$

The aim of embedding is to match these distributions as much as possible, and this has been done using Kullback-Leibler divergences as:

$$C = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{2.75}$$

Advantages and Limitations

The performance of SNE is closely related to pr or k. In addition, i-th Gaussian distributions in the high dimensional space have been converted in a Gaussian distribution with a standard

distribution, which may cause *tears* or *false neighbours* in the low dimensional space data.

2.1.18 *t*-Stochastic Neighbour Embedding (SNE)

t-SNE is a dimensionality reduction method based on SNE, but it changes from SNE in two ways: (1) it uses the symmetric version of SNE, and (2) uses a Student-t distribution instead of Gaussian distribution to compute similarity in the low-dimensional space. t-SNE is a nonlinear dimensionality reduction technique which calculates the conditional probability $p_{i|j}$ between samples x_i and x_j using the Gaussian distribution, centred at x_j with the variance σ_i as shown in Eq. (2.76).

$$p_{i|j} = \frac{exp(\frac{-dis(x_i,x_j)^2}{2\sigma_i^2})}{\sum_{k\neq i} exp(\frac{-dis(x_i,x_k)^2}{2\sigma_i^2})} \tag{2.76}$$

The high dimensional space similarity p_{ij} is calculated using Eq. (2.76): $p_{ij} = \frac{p_{i|j}+p_{j|i}}{2N}$, whereas, the low dimensional similarity is calculated as shown in Eq. (2.77).

$$q_{ij} = \frac{(1 + dis(x_i, x_j)^2)^{-1}}{\sum_{k\neq l}(1 + dis(x_i, x_j)^2)^{-1}} \tag{2.77}$$

t-SNE tries to make the low dimensional similarity q_{ij} as similar as possible to its corresponding high dimensional similarity p_{ij}. Consider Eq. (2.76), t-SNE builds N-Gaussian distributions related to density σ and the distance of each sample x_i to its neighbours. If the distance between the sample x_i and its neighbours is small, the Gaussian distribution is sharp; otherwise, it broadens.

Advantages and Limitations

t-SNE is a very popular method in visualisation due to the ability to maintain the data structure with the low dimensional representation in one (or more) curved and nonlinear manifold(s). Despite all benefits of t-SNE, the low dimensional space data produced has two problems, tear and

false neighbours. The main cause of those problems is using two different similarity functions to measure the similarity between high and low dimensional space data. In high dimensional space data, the similarity between data samples is measured by N-Gaussian distributions, whereas in the low dimensional space, the similarity between data samples has been defined using Student-t distribution (degree of freedom=1). In other words, data samples with different Euclidean distances in the high dimensional space might be mapped so that they have the same Euclidean distance in the low dimensional space, resulting in a failure with regards to data structure capturing. Besides, the goodness of the captured structure of low dimensional data generated by t-SNE relies on perplexity, making t-SNE a costly method. Also, t-SNE is considered a local dimensionality reduction technique because the Gaussian distribution measures the similarity between high dimensional space data and is a sharp distribution. As the distance between data becomes larger, the similarity produced by Gaussian distributions becomes closer to zero. As a result, Global t-SNE has been proposed to make t-SNE a better method to capture the global data structure.

2.1.19 Global *t*-SNE

Zhou and Sharpee [57] presented the Global t-SNE method to capture a more global structure of the high dimensional space data. Global t-SNE suggests using an exponential distribution in addition to Gaussian distribution. If the Gaussian distribution is sensitive to smaller distances, the exponential distribution is unstable to larger distances because of its heavy tails.

Advantages and Limitations

Global t-SNE like t-SNE is costly due to tuning perplexity. Also, Global t-SNE is prone to the two problems that occurred to t-SNE, tear and false neighbours.

2.1.20 Multiscale SNE

SNE, t-SNE and Global t-SNE are considered costly due to the number of neighbours or perplexity tunning requirements. To deal with that issue, Multiscale SNE [58] has proposed employing multi-perplexities distributions in high dimensional space to maintain small and large distances. Multiscale SNE is an extension of SNE, using Gaussian distributions in high and low dimensional spaces to keep the data structure; it defines the probabilities as follows:

$$p_{hij} = \frac{exp(\frac{-r_{hi}dis(x_i,x_j)^2}{2})}{\sum_{k\neq i} exp(\frac{-r_{hi}dis(x_i,x_j)^2}{2})} \tag{2.78}$$

$$q_{hij} = \frac{exp(\frac{-s_{hi}dis(x_i,x_j)^2}{2})}{\sum_{k\neq i} exp(\frac{-s_{hi}dis(x_i,x_j)^2}{2})} \tag{2.79}$$

$$p_{ij} = \frac{1}{L}\sum_{h=L_{min}}^{L_{max}} p_{hij} \tag{2.80}$$

$$q_{ij} = \frac{1}{L}\sum_{h=L_{min}}^{L_{max}} q_{hij} \tag{2.81}$$

where r_{hi} and s_{hi} denote precision in high and low dimensional spaces, respectively, and $1 \leq L_{min} \leq h \leq L_{max}$, where $L = L_{max} - L_{min} + 1$ is considered the number of scales (number of different perplexities employed). In [58] it is suggested using $L_{min} = 2$ and $L_{max} = \log_2 \frac{N}{2}$.

Advantages and Limitations

Multiscale SNE improves capturing a more global structure but increases the computational complexity by $\log_2 \frac{N}{2}$. Tuning the scale parameters determines the algorithm's efficiency, making multiscale SNE a complex and costly method. Also, employing Gaussian distribution in high and low dimension space to measure the similarity between data samples makes Multiscale

SNE unsuitable for preserving large distances.

2.1.21 Multiscale *t*-SNE

Multiscale t-SNE [59, 60] employs multi-perplexities Gaussian distributions in high dimensional space like Multiscale SNE [58], and it has two main drawbacks: 1) is prone to tear and false neighbour problems, which occurs by employing different distributions in high and low dimensional spaces to measure similarities between data samples, and 2) it is not suitable for large distances preservation since Gaussian distribution gives a low priority to large distances.

2.1.22 Uniform Manifold Approximation and Projection (UMAP)

UMAP, a similar method to t-SNE is a useful technique to capture the local structure of the high dimensional space data. For each data sample in the high dimensional space x_i let define ρ_i and σ_i, where

$$\rho_i = \min\left(dis(x_i, x_j), 1 \leq j \leq \ k, dis(x_i, x_j) \geq 0\right) \tag{2.82}$$

$$\sum_{j=1}^{k} exp(\frac{-\max 0, dis(x_i, x_j) - \rho_i}{\sigma_i}) = \log_2 k \tag{2.83}$$

and the similarity function is defined as in Eq. (2.84).

$$w_{ij} = exp(\frac{-\max 0, dis(x_i, x_j) - \rho_i}{\sigma_i}) \tag{2.84}$$

$$\bar{w_{i,j}} = w(x_i, x_j) + w(x_j, x_i) - w(x_i, x_j)w(x_j, x_i) \tag{2.85}$$

A be the weighted adjacency matrix of G. D degree matrix of graph A, and

$$L = D^{\frac{1}{2}}(D - A)D^{\frac{1}{2}} \tag{2.86}$$

$$evec = Eigenvectors\ of\ L \tag{2.87}$$

$$Y = evec[1...d + 1] \tag{2.88}$$

Advantages and Limitations

UMAP gives more importance to the local structure capturing than the global structure. Also, the scale of maintained data structure relates to the number of neighbours, which its tuning makes UMAP a very costly method.

2.1.23 Trimap

To capture a more global structure of the data, Amid and Warmuth [40] presented the TriMap method, which considers the similarities of three data samples (triplets) instead of a pair of data samples. TriMap defines a set of triplets $T = \{(i, j, k) : p_{ij} > p_{ik}\}$, where the satisfaction probability of the triplet (i, j, k) is defined as in Eq. (2.89).

$$Pr_{ijk} = \frac{q_{ij}}{q_{ij} + q_{ik}} = \frac{1}{1 + \frac{q_{ik}}{q_{ij}}} \tag{2.89}$$

The low dimensional representation can be calculated by minimising the cost function

$$\sum_{(i,j,k)\in T} l_{i,j,k} \tag{2.90}$$

$$l_{i,j,k} = w_{i,j,k}\frac{s(y_i, y_k)}{s(y_i, y_j)s(y_i, y_k)} \tag{2.91}$$

$$w_{i,j,k} = log(1 + 500(\frac{exp^{\frac{dis(x_i,x_k)^2}{\sigma_i\sigma_k} - \frac{dis(x_i,x_j)^2}{\sigma_i\sigma_j}}}{max_{i',j',k'\in T} exp^{\frac{dis(x_i,x_k)^2}{\sigma_i\sigma_k} - \frac{dis(x_i,x_j)^2}{\sigma_i\sigma_j}}})) \tag{2.92}$$

$$s(y_i, y_j) = (1 + \left\|y_i - y_j\right) \|^2)^{-1} \tag{2.93}$$

The inilitazion of Y has been done using PCA.

Advantages and Limitations

TriMap, like t-SNE, employs Gaussian distributions in the high dimensional space and Student-t distribution in the low dimensional space to measure similarities between data samples. As demonstrated with t-SNE, using different N- Gaussian distributions in the high dimensional space and one Student-t distribution in the low dimensional space cause tears or false neighbours, which also occurs in TriMap.

2.1.24 Autoencoders and Restricted Boltzmann Machine (RBM)

Autoencoders[3] are neural networks composed of two parts *encoder* and *decoder*. The encoder uses ϕ function (2.94) to embed the original high dimensional data X to the low dimensional data Y. In contrast, the decoder uses the function ψ (2.95) to embed the low dimensional data Y to the output data X', where X' is the reconstructed data of the original data X by minimising the cost function in (2.96).

$$\phi : X \to Y \tag{2.94}$$

[3]Autoencoders are neural networks composed of one input layer, one output layer and one hidden layer, whereas deep autoencoders are multi-layered neural networks composed of one input layer, one output layer and many hidden layers.

$$\psi : Y \to X \tag{2.95}$$

$$\phi, \psi = arg \min_{\phi,\psi} ||X - (\psi \circ \phi)X'||^2 \tag{2.96}$$

Advantages and Limitations

Deep autoencoders [49, 61, 62, 63, 64] are multi-layered neural networks, where each pair of neighbourhood layers is considered to be a Restricted Boltzmann Machine (RBM). However, like all neural networks, it isn't easy to find the optimal parameters for RBMs; and as such, their selection is heuristic or based on previous experiments [65]. Above all, most of the methods disregard the preservation of the data manifold structure [66]. Hence, to improve RBM and to capture the local data structure, neighbourhood graphs have been used [66]. However, it is complex to implement this approach since it requires tuning the number of neighbours and the number of hidden layers, the number of nodes in each hidden layer, the number of epochs, and the batch size.

2.1.25 DD-HDS

DD-HDS is a nonlinear dimensionality reduction method similar to MDS, but it favours more preservation of small distances, possibly at the price of distortions in large distances. DD-HDS uses athe weighting function

$$k(d_{ij}) = 1 - \int_{-\infty}^{d_{ij}} f(u, \mu, \sigma)du, \tag{2.97}$$

where $f(u, \mu, \sigma)$ is the probability density function of a Gaussian variable with mean μ and standard deviation σ.

$$\mu = mean_{1 \le i \le j \le N}(d_{ij} - 2(1 - \lambda)std_{1 \le i \le j \le N}(d_{ij})) \tag{2.98}$$

$$\sigma = 2\lambda std_{1\leq i\leq j\leq N}(d_{ij})) \tag{2.99}$$

where mean and standard deviation (std) are taken over the distribution distances between all pairs of data in the original space. The scale of the influence of large distances have over the small distances is controlled by λ, which is a positive user-defined parameter, and usually takes value between 0.1 and 0.9. The cost function of DD-HDS is

$$E_{DD-HDS} = \sum_{i<j} \left(\left\| d_{ij} - d'_{ij} \right\| \left(1 - \int_{-\infty}^{min(d_{ij}, d'_{ij})} f(u, \mu, \sigma) du \right) \right) \tag{2.100}$$

This method uses symmetric measures in high and low dimensional space, which helps to prevent the problems tear and false neighbours.

Advantages and Limitations

DD-HDS performance depends on tuning the parameter λ_1, and it is not good in preserving the global data structure (large distances). Furthermore, it employs Gaussian distribution, which includes exp, and it can be costly. The tails of the gaussian distribution are not long enough to capture the global structure, and that is why the authors claimed that DD-HDS did not perform well in capturing large distances.

2.1.26 RankVisu

RankVisu is a similar method to DD-HDS. However, DD-HDS aims to preserve distances, whereas RankVisu aims to preserve the neighbourhood with small neighbourhood ranks. Because it preserves the neighbourhood rank, it is similar to non-metric MDS. Neighbourhood rank can be derived directly from pairwise distances, where the closest neighbour can be allo-

cated with 1 and the less closest neighbour with 2 and as follows.

$$\zeta_1(i,j) = (k+1-min(D_{ij},\Delta_{ij})) \times \left|D_{ij}-\Delta_{ij})\right| \tag{2.101}$$

$$\zeta_1 = \sum_{i,j} \zeta_1(i,j) \tag{2.102}$$

$if \delta < k+1 \ \ and \ \ \delta_{ij} < \min(D_{ij}, D_{ji})$

$$\zeta_2(i,j) = (k+1-\delta_{ij}) \times \left|\min(D_{ij},D_{ji}) - \delta_{ij})\right| \tag{2.103}$$

$if \ else \ \ \min(D_{ij}, D_{ji}) < k+1 \ \ and \ \ \delta_{ij} > \max(D_{ij}, D_{ji})$

$$\zeta_2(i,j) = (k+1-\max(D_{ij},D_{ji}) \times \left|\min(D_{ij},D_{ji}) - \delta_{ij})\right| \tag{2.104}$$

$$else \ \ \zeta_2(i,j) = 0 \tag{2.105}$$

$$\zeta = (1-\alpha) \times \zeta_1 + \alpha \times \zeta_2 \tag{2.106}$$

where D and Δ are the Neighbourood Ranks in the high and low dimensioanl space, respectively, and α is a balancing parameter belongnig to $[0, 1]$.

Advantages and Limitations

Although RankVisu works well in heavy curved manifold data, its performance also depends on the number of neighbours, making it an expensive method.

2.1.27 Summary and Literature Gap

From the review made on the current literature, it has been identified that the performance of a dimensionality reduction method in terms of maintained data structure is related to:

1. The type of data, and

2. Parameter tuning.

PCA and MDS work well in linear data, do not require parameter tuning, and therefore can save computational time; they neglect the maintenance of local information of high dimensional space data.

On the other hand, nonlinear data has its low dimensional representation lying on :

1. Smooth manifolds

2. Non-smooth manifolds that are further categorised into:

   (a) developable curved manifolds

   (b) heavy curved manifolds

Methods that work well in smooth nonlinear data are Sammon's mapping, CCA, LLE, HLLE, MVU, LTSA, MLLE, and diffusion maps. However, in cases of having non-smooth nonlinear data, the above-mentioned methods do not perform well.

Non-smooth nonlinear data are further classified in developable manifolds and heavily curved manifolds. Dimensionality reduction techniques that can perform well in developable manifolds are Isomap and CDA. Isomap and CDA employ Geodesic distances to measure the similarity between high dimensional spaces data and Euclidean distances to measure the similarity between low dimensional space data samples.

All the above-mentioned methods work well in linear, simple smooth, or developable manifolds. However, suppose the low dimensional representation is on heavily curved manifolds. In that case, a group of methods such as SNE, LE, Autoencoders, DD-HDS, t-SNE, RankVisu, Trimap and Umap can maintain the data structure. Also, t-SNE, DD-HDS, UMAP, and TriMap have proposed using Gaussian or Student-t distributions to provide a softer border between local and global structure maintenance. However, the scale of the maintained data structure is

closely related to tuning the number of neighbours, perplexity, or λ_1, to generate the best low dimensional representation. Multiscale approaches such as Multiscale-SNE and Multiscale t-SNE attempted to overcome this shortcoming; however, it still is a costly method due to both the multiscale calculations and the utilisation of Gaussian distribution, and it is much slower than using Student-t distribution. In addition, Gaussian-based dimensionality reduction methods have two other problems:

1. They favour the local structure capturing at the expense of neglecting the global structure capturing, due to the shape of Gaussian distribution[4], and

2. Gaussian distribution is expensive due to including the exponential function.

Also, two other problems spotted in dimensionality reduction techniques, tears and false neighbours. The main cause of tears and false neighbours problems is due to employing different distributions or different similarity measures in high and low dimensional spaces. By employing different distributions in high and low dimensional spaces, the same distance is converted into different similarities, or different distances are converted into the same similarities.

In summary, the current dimensionality reduction methods, t-SNE, UMAP, Trimap and DD-HDS work well in heavily curved manifolds. However, they still require costly tuning parameters, and also, they are strictly local dimensionality reduction methods and suffers form tears and false neighbours problems. A summary of dimensionality reduction methods is presented in Table 2.2.

[4]Gaussian distribution generates large similarity values for short distances, and as distance increases, the similarity converges to zero sharply.

Table 2.2: SUMMARY OF TYPE OF DATA AND REQUIRED PARAMETERS THAT AFFECT THE PERFORMANCE OF EACH DIMENSIONALITY REDUCTION METHOD

DR algorithm	Parameters	Type of Data	Type of Manifold
PCA	None	Linear	Linear manifold
MDS	None	Linear	Linear manifold
Sammon Mapping	None	nonlinear	Smooth manifold
CCA	λ	nonlinear	Smooth manifold
CDA	λ	nonlinear	Developable Heavily curved manifold
Isomap	k	nonlinear	Developable heavily curved manifold
LLE	k	nonlinear	Smooth manifold
LE	k, σ	nonlinear	Heavy curved manifold
HLLE	k	nonlinear	Smooth manifold
MVU	k	nonlinear	Smooth manifold
LTSA	k	nonlinear	Smooth manifold
Diffusion Maps	σ, t	nonlinear	Smooth manifold
Autoencoders	$NetSize$	nonlinear	Heavily curved manifold
MLLE [33]	k	nonlinear	Smooth manifold
DDHDS	$\lambda 1$	nonlinear	Heavily curved manifold
t-SNE	pr	nonlinear	Heavily curved manifold
RankVisu	k	nonlinear	Heavily curved manifold
Trimap	k	nonlinear	Heavily curved manifold
UMAP	k	nonlinear	Heavily curved manifold
Missing	**None**	**nonlinear**	**Heavily curved manifold**

Finally, the literature misses a dimensionality reduction technique that does not require tuning parameters and maintains the structure of data with low dimensional representation on heavily curved manifold(s). To fill the literature gap, the Same Degree Distribution (SDD) method for dimensionality reduction has been proposed, together with Multi Same Degree Distributions (MSDD) and parameter-free SDD, aiming to capture the geometry of data having low dimensional representation in heavily curved manifolds, in significantly less computational time.

2.2 Supervised Dimensionality Reduction Techniques

Dimensionality reduction methods have been commonly applied in different fields, including medical images [67, 68] and financial markets [69], to visualise high dimensional data or as a pre-processing step of classification. However, the main focus of manifold learning techniques

is preserving data structure; thus, they may not be helpful in classification. In addition to the current dimensionality reduction techniques, their supervised versions have been considered to improve maintaining the data structure or improving the classification accuracy. Supervised dimensionality reduction methods use dissimilarity measures instead of Euclidean distance to define the similarity between data samples. Data samples of the same class are enforced to be close, and data samples of different classes are enforced to be located far away.

Four common used dissimilarity measures are as follows:

$$dis_1 = \begin{cases} \sqrt{1 - e^{\frac{-dis(x_i,x_j)^2)}{\beta}}} & l_i = l_j \\ \sqrt{e^{\frac{dis(x_i,x_j)^2)}{\beta}}} - \alpha & l_i \neq l_j \end{cases} \tag{2.107}$$

$$dis_2 = \begin{cases} \frac{1}{\psi} dis(x_i, x_j) & l_i = l_j \\ dis(x_i, x_j) & l_i \neq l_j \end{cases} \tag{2.108}$$

$$dis_3 = \begin{cases} dis(x_i, x_j) & l_i = l_j \\ dis(x_i, x_j) + \mu \max(dis(x_i, x_j))\lambda_{ij} & l_i \neq l_j \end{cases} \tag{2.109}$$

$$dis_4 = \begin{cases} dis(x_i, x_j)e^{v(x_i)-v(x_j)} & l_i = l_j \\ dis(x_i, x_j) & l_i \neq l_j \end{cases} \tag{2.110}$$

Dissimilarity measure dis_1 was applied in supervised t-SNE by Hajderanj et al. [70] in datasets MNIST, SEER Breast Cancer and Chest X-ray, where supervised t-SNE generated visualisation with higher class separability, and the classification error were lower than using original t-SNE. Dissimilarity measure dis_1 was also implemented to LLE to produce Enhanced Supervised Locally Linear Embedding (ESLLE) [71] to achieve a higher classification accuracy in Swiss Roll data.

WeightedIso approach [72] is a supervised manifold learning method that uses dissimilarity measure dis_2 Eq. (2.108) in Isomap implemented in Iris, Liver, Lung Sonar, Glass, and Image. Supervised Locally Linear Embedding (SLLE) [73] separates data samples of different classes and make closer data samples of the same class as in Eq. (2.109), where $\mu \in [0, 1]$ and λ_{ij} is 0 if data samples i and j are from the same class, and 1 otherwise. Yu et al. [74] and Cheng et al. [75] proposed a supervised version of t-SNE, where the distance between different classes data samples is defined in Eq. (2.110), where $v_(x_i)$ refers to the angle information [72] and the silhouette frame information [75] of the sample x_i.

Although it is not a published article, a supervised version of UMAP[5] has been proposed to capture the high dimensional data structure. However, some researchers [71, 72, 76] have proposed using dissimilarity measures to generate a better structure capturing visualisation. The main reason for using the dissimilarity measures is to apply the low dimensional space data for classification. Accordingly, Hajderanj et al. [70], Vlachos et al. [72], Geng et al. [76], and Wei et al. [77] have proposed supervised dimensionality reduction techniques that use the class information to guide the dimensionality reduction process to improve the classification accuracy. Furthermore, they have used supervised dimensionality reduction techniques to improve the data structure preservation of their unsupervised versions. The experimental findings in [70, 72, 76, 77, 78] have illustrated the effectiveness of supervised dimensionality reduction techniques in gaining a better classification model and capturing the data structure more accurately.

2.2.1 Summary and Literature Gap

There are many supervised dimensionality reduction methods. Some researchers have claimed that dissimilarity measures improve structure maintaining of the high dimensional space data. In addition, some other researchers argued that employing dissimilarity measures in a reduction method helps generate a higher classification accuracy than using standard metrics such as Euclidean distances or Geodesic distances. However, there lacks in the literature theoretical foundation on the impact of dissimilarity measures on the structure maintaining and classifi-

[5]https://umap-learn.readthedocs.io/en/latest/supervised.html

cation accuracy.

2.3 Dimensionality Reduction Quality Assessment

In general, the parameter that indicates the performance of a dimensionality reduction technique is its loss (cost) function. In each method, the cost function is formulated to achieve a specific aim, such as minimising the distances between high and low dimensional space data or maximising the covariance. Since the dimensionality reduction methods may have different aims, their cost function values can not be used to compare the performances of dimensionality reduction techniques. To compare dimensionality reduction methods should be considered quality assessment techniques that aim at a single objective, such as geometry-preservation (preserving the local geometry of data or the global geometry of data). Table 2.3 has presented some quality assessments techniques, described byS the *criterion* (Local or Global).

Table 2.3: METHODS FOR DIMENSIONALITY REDUCTION QUALITY ASSESSMENTS

Year	Name of the Measure	Criterion	References
1968	Kendall's Correlation	Local	[79]
1964	Kruskal Stress Measure	Global	[80]
1988	Spearman's Rho	Local	[81]
1992	Topological Product	Local	[82]
2000	Koing's Measure	Local	[83]
2006	Trustworthiness and Continuity	Local	[84]
2006	Local Continuity/Meta Criterion	Local	[85]
2008	Mean Relative Rank Errors	Local	[86]
2009	Co-Ranking Matrix	Local	[87]

2.3.1 Kendall's Tau (τ)

Kendall's Tau (τ) is one of the first measures to estimate rank correlation and has been successfully applied on the topology preservation after a DR process [79]. This coefficient (τ) measures

the correlation between the distance rank of the high and the low dimensional data as follows:

$$\tau = \frac{C - D}{\sqrt{((C + D + T) * (C + D + U))}} \tag{2.111}$$

where the number of concordant pairs is denoted with C, and the number of discordant pairs is denoted with D, while T and U are the numbers of ties in pairwise distance matrices of the high and the low dimensional spaces DIS and dis, respectively. If a tie occurs for the same pair in both DIS and dis, it will not be added to either T or U, and the input of the data should be in a one-dimensional array. Therefore, the pairwise distance matrixes in both the high dimensional space (DIS) and the low dimensional space (dis) will be flattened to a one-dimensional array. The value of τ ranges between -1 and 1. If τ is close to 1, ranks have a high correlation. On the other hand, if τ is close to -1 or 0, it means there is no relation or negative relation between ranks. Ranks of distances between the high and the low dimensional spaces represent the ranks of neighbours for both spaces, respectively. Consequently, a high value of τ means that the neighbour's rank is captured. In terms of comparison, the best dimensionality reduction method is the method with the highest value of τ.

2.3.2 Kruskal Stress Measure (KSM)

KSM is the residual sum of the squares between dissimilarities δ and fitted distances γ shown as in Eq.(2.112).

$$KSM = (\sum_{i \neq j = 1,\ldots,n} (\delta_{ij} - \|\gamma_{ij}\|)^2)^{\frac{1}{2}} \tag{2.112}$$

2.3.3 Spearman's Rho

Spearman's Rho measures the correlation between rank order data and assess how well the order between datapoints in high dimensional space has been kept.

$$S_R = 1 - \frac{6\sum_{i=1}^{T}(z(i) - \widehat{z}(i))^2}{T^3 - T} \tag{2.113}$$

where $z(i)$ and $\widehat{z}(i)$ for $i = 1, ..., T$ are the different ranks (order number) of pairwise distances in the original and embedded spaces, respectively. T is the total number of distances $T = n(n-1)/2$. $S_R \in [-1, 1]$

2.3.4 Topological Product (T_{Pr})

T_{Pr} measure the preservation of distances within the local neighbourhoods, and $T_{Pr} = 0$, there exist a perfect mapping.

$$T_{Pr} = \frac{1}{n(n-1)} \sum_{g=1}^{n} \sum_{f=1}^{n-1} log(\prod_{p=1}^{f} Q_1(g,p)Q_2(g,p))^{\frac{1}{2f}} \tag{2.114}$$

where $Q_1(i,j)$ and $Q_2(i,j)$ are the distances between the point i and its j^{th} nearest neighbours.

2.3.5 Konig's Measures (KM)

KM measures the local structure preservation based on the ranks order of the orignal and the embedded spaces.

$$K_M = \frac{1}{3kn} \sum_{i=1}^{n} \sum_{j=1}^{k} KM_{ij} \tag{2.115}$$

$K_M \in [0, 1]$, and where it is 1, it is a perfect embedding.

2.3.6 Trustworthiness & Continuity ($T\&C$)

$T\&C$ includes two parameters *trustworthiness* and *continuity* defined as :

$$M_T = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \in U_k(i) \notin V_k(i)} (r(i,j) - k) \tag{2.116}$$

$$M_C = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \in U_k(i) \notin V_k(i)} (\widehat{r}(i,j) - k) \tag{2.117}$$

where k is the neighbourhood size and $r(i,j)$ and $\widehat{r}(i,j)$ are the rank of $x_i \in X$ and $y_i \in Y$, whereas $U_k(i)$ and $Vk(i)$ is the set of those datasamples that are the k nearest neighbours of high and low dimensional data samples $x_i(y_i)$. In other words, M_T measures that data point that originally were farther away in the original space are puted as neighbours in the embedded space, whereas M_C that datapint that are originally close and are embedded farther away in the embedded space. $T\&C$ is measures

$$Q_t = \alpha M_T + (1 - \alpha) M_C \tag{2.118}$$

where $\alpha \in [0, 1]$.

2.3.7 Local Continuity Meta-Criterion (LCMC)

LCMC checks the performance of a dimensionality reduction method based on the degree of overlap between the neighbours sets of a data samples and their corresponding embedding as:

$$Q_k = 1 - \frac{1}{nk} \sum_{i=1}^{n} \left(\Psi_k^X(i) \cap \Psi_k^y(i) \right) - \frac{k^2}{n-1} \tag{2.119}$$

where k is the number of neighbours and $\Psi_k^X(i)$ and $\Psi_k^y(i)$ are the index sets of of $x(i)$ and $y(i)$'s k data samples. Q_k takes value between 0 and 1, where values close to 1 means a high neighbourhood overlaped between the high and low dimensional spaces.

2.3.8 Mean Relative Rank Error (MRRE)

(MRRE) is based on the ranks of pairwise Euclidean distances withing local neighbourhoods. MRRE is similar to $T\&C$ and defines two elements:

$$W_T = 1 - \frac{1}{H_k} \sum_{i=1}^{n} \sum_{j \in U_k(i)} \frac{\left| r(i,j) - \widehat{r}(i,j) \right|}{r(i,j)} \tag{2.120}$$

$$W_C = 1 - \frac{1}{H_k} \sum_{i=1}^{n} \sum_{j \in V_k(i)} \frac{\left| r(i,j) - \widehat{r}(i,j) \right|}{\widehat{r}} \tag{2.121}$$

where k is the size of the neighbourhood and

$$H_k = n \sum_{i=1}^{k} \frac{\left| n - 2i + 1 \right|}{i} \tag{2.122}$$

$$Q_M = \beta W_T + (1 - \beta) W_C \tag{2.123}$$

2.3.9 Co-ranking matrix

Co-ranking matrix is also a measure of the dimensionality reduction method quality. Let us define $DIS_{N \times N}$ and $dis_{N \times N}$ the matrixes of pairwise distances in the high and low dimensional spaces, respectively. In both spaces the rank matrices R_{NXN} and r_{NXN} of the distance matrixes DIS_{NXN} and dis_{NXN} are calculated as follows:

$$R_{ij} = \left| \{ k : DIS_{ik} < DIS_{ij} \} \right| \tag{2.124}$$

$$r_{ij} = \left| \{ k : dis_{ik} < dis_{ij} \} \right| \tag{2.125}$$

where $|\cdot|$ defines the set of cardinality. The co-ranking matrix Q is defined by

$$Q_{kl} = \left|\{(i,j) : R_{ij} = k \, and \, r_{ij} = l\}\right| \tag{2.126}$$

Errors generated by a dimensionality reduction method correspond to off-diagonal entries of the co-ranking matrix [87]. A diagonal co-ranking matrix represents a perfect dimensionality reduction method.

In addition to the above-mentioned methods, a difference matrix named *Retained-Structure* is constructed.

2.3.10 Retained-Structure

Retained-Structure is a matrix that contains the difference between the matrix that contains the neighbourhood rank of the high dimensional space data and the matrix that contains the neighbourhood rank of the low dimensional space data. In an ideal case, the *Retained-Structure* is a matrix that contains only element 0 (zero). Non-zero elements indicate a failure to retain the neighbourhood structure and are positive or negative. A positive number $P_{ij} = +in$ indicates that the method has jumped $+in$ closer the j^{th} data sample to the i^{th} data sample. By contrast, a negative number $P_{ij} = -in$ indicates that the method has enforced the i^{th} data sample to be in positions far away from the j^{th} data sample. Overall, positive values in *Retained-Structure* can cause *tear* whereas negative values can cause *false neighbours*. Also, error E has been defined as $E = sum(abs(in))$, such that the lower the E, the better the method.

As can be seen, there are many quality assessment methods, but not all of them will be used in this research. Kendall's Tau, Trustworthiness & Continuity, and Co-ranking matrix will be employed to analyse the dimensionality reduction methods in terms of the scale of structures they have captured. In addition, the Retained-Structure matrix has been proposed in this research to analyse the impact of dissimilarity measures in structure maintaining practically. However, the Retained-Structure matrix can also be used to compare the dimensionality reduction method, although it has been suggested to be used when the number of data samples

is small.

2.3.11 Summary and Literature Gap

Although there are different quality assessment methods, only a few researchers [88, 89, 90, 91] have integrated them to demonstrate the performance of the dimensionality reduction method. The Experimental Results Chapter indicates that tuning parameters has a significant impact on the maintenance of high dimensional space data structure, and measuring the quality of a dimensionality reduction is essential in the trustworthiness of the low dimensional space data.

2.4 Chapter Summary

In summary, the literature review presented a broader review on the dimensionality reduction techniques emphasising the factors that impact the scale of maintained data structure. Also, a review of supervised dimensionality reduction techniques has been provided by highlighting their applicability for visualisation and classification purposes. This research has also reviewed quality assessment techniques for dimensionality reduction techniques in structure maintenance. The review has been finalised by identifying two major gaps in the literature as follows:

1. Misses a dimensionality reduction technique that captures the best data structure of any data type without requiring tuning the time-consuming parameters, and

2. Lacks theoretical studies on the impact of dissimilarity measures on the structure maintaining and classification accuracy.

Chapter 3

Methodology: Developed Approaches

This Chapter presents four developed dimensionality reduction approaches, SDD, MSDD, parameter-free SDD and parametric SDD. SDD, MSDD, and parameter-free SDD approach to capture a better data structure in less computational time than existing dimensionality reduction approaches. Pseudocode, implementation guide and complexity analyses have been presented to describe better the developed approaches. In addition, theoretical analyses have been presented to analyse the performance of parameter-free SDD.

As concluded in the literature gap, there lacks a dimensionality reduction method that successfully maintains the structure of heavily curved manifold data types and does not require tuning the number of neighbours or any costly parameter. t-SNE, UMAP, Trimap and DD-HDS are the dimensionality reduction methods that work well in heavily curved manifolds; however, their performance in terms of the maintained data structure scale depends on tuning parameters such as the number of neighbours, perplexity, and some other parameters that their tuning makes the dimensionality reduction a very costly process. In addition, t-SNE, UMAP, Trimap and DD-HDS calculate the similarity between data samples considering the Gaussian distribution, which itself favours more short distances than larger ones. Consequently, all the mentioned methods do not work well when the maintenance of large distances is essential. Also, in some methods like t-SNE, which uses different distribution in high and low dimensional spaces, it has occurred the problems of tears and false neighbours.

To overcome all problem occurred in the methods mentioned above, Same Degree Distribution (SDD) method [92] for dimensionality reduction is proposed, together with Multi Same Degree Distributions (MSDD) and parameter-free SDD, aiming to capture the geometry of data having low dimensional representation in non-smooth and developable manifolds, in very less computational time.

3.1 Same Degree Distribution (SDD) Approach

This research aims to overcome all mentioned problems of the current methods by proposing Same Degree Distribution (SDD). SDD has been designed to:

1. Captures better the global data structure by employing the degree-distribution. Degree-distribution uses the probability density function as:
   $(p_{deg_m})_{ij} = \frac{(1+dis(x_i,x_j))^{-deg_m}}{\sum_{k\neq l}(1+dis(x_k,x_l))^{-deg_m}}$.
   Note that probability density function of Student-t distribution is:
   $(p_{deg_m})_{ij} = \frac{(1+dis(x_i,x_j))^{-\frac{deg_m+1}{2}}}{\sum_{k\neq l}(1+dis(x_k,x_l))^{-\frac{deg_m+1}{2}}}$.
   Degree-distribution ($deg = 1$) is the same as Student-t ($deg = 1$), and for greater degrees, degree-distributions ($deg > 1$) are sharper than Student-t distributions ($deg > 1$).

   On the other hand, the degree-distribution has longer tails than the Gaussian distribution (2.76), and this means the similarities generated by degree-distribution converges slower to zero than the Gaussian distribution. As a result, employing the degree-distribution instead of Gaussian distribution makes SDD a more global method than other ones that are Gaussian based.

2. SDD does not require tuning the number of neighbours, perplexity, but instead, it requires tuning the deg of degree-distribution. The range of degree in degree-distribution is from 1 to 15, where degree-distribution with $deg = 1$ is smoother, and as the degree increases, the sharper the method becomes, by favouring the short distance preservation and neglecting global structure. This makes SDD a significantly less computational method than other

methods that require the number of neighbours or perplexity, ranging from 1 to the total number of samples -1.

3. SDD employs the same degree degree-distribution, to prevent the problem of tears and false neighbours.

SDD is a nonlinear dimensionality reduction technique with pseudocode shown in Algorithm 1. It employes degree-distribution in the high (3.3) and the low (3.4) dimensional spaces to capture the local and global data structure. Degree-distribution is Student-t distribution when the degree of freedom is 1, and for greater degrees, it looks as sharper Student-ts. SDD intends to find a suitable degree to best capture the structure of the data. Degree-distributions are more sensitive to small distances, and the greater the distance, the less sensitive degree-distribution becomes. Thus, rescaling the pairwise distances of high dimensional data into the interval range between 0 and 1 would be an essential step in the performance of the proposed approach in terms of capturing the data structure. As a result, high dimensional space similarities of a degree-distribution will be calculated using the scaled Euclidean distances instead of the Euclidean distances. Kullback-Leibler is the loss function used in SDD to approximate the degree-distribution in the low dimensional space with the degree-distribution in the high dimensional space:

$$C_1 = \sum_{i \neq j} (p_{deg_m})_{ij} \log(\frac{(p_{deg_m})_{ij}}{(q_{deg_m})_{ij}} \tag{3.1}$$

where deg_m is the degree of degree-distribution m, $m = 1 : n$. SDD intends to minimize the cost function C_1 as (3.2):

$$loss_1 = \min(C_1) \tag{3.2}$$

where

$$(p_{deg_m})_{ij} = \frac{(1 + dis(x_i, x_j))^{-deg_m}}{\sum_{k \neq l} (1 + dis(x_k, x_l))^{-deg_m}} \tag{3.3}$$

$$(q_{deg_m})_{ij} = \frac{(1 + dis(y_i, y_j))^{-deg_m}}{\sum_{k \neq l}(1 + dis(y_k, y_l))^{-deg_m}} \tag{3.4}$$

Algorithm 1 SDD

Require: Input :

$X \in R^{NxD}$, number of iterations H, learning rate η, momentum α, number of degree-distributions n, degree deg_m, initial low dimensional data $Y^0 = y_1, ..., y_N \in N(0, 10^{-4}I)$, ϵ.

Step 1 :

Compute the high dimensional space similarities $(p_{deg_m})_{ij}$ using (3.3) and store in matrix P_{deg_m}.

Step 2 :

Compute the low dimensional space similarities $(q_{deg_m})_{ij}$ using (3.4) and store in matrix Q_{deg_m}.

Step 3 :

Compute the gradient $\frac{\delta C_1}{\delta\ yi}$ where C_1 is defined in (3.1).

Step 4 :

Minimize the objective function using the Gradient Descent optimisation algorithm and update the low dimensional space as: $Y^h = Y^{h-1} + \eta\frac{\delta C_1}{\delta y_i} + \alpha(Y^{h-1} - Y^{h-2})$.
The optimisation algorithm will stop either achieves the maximum number of iterations H or the Kullback-Leibler value is lower than the minimum threshhold ϵ.

Output :

Low dimensional space represenation $Y_{bestdeg_m}$.

However, the minimal loss function value of (3.2) does not reflect how well the data structure is captured. Thus, to have a better indication of the goodness of a dimensionality reduction method, we propose the use of Kendall's Tau correlation coefficient (τ). This coefficient (τ) measures the correlation between distance rank of the high and the low dimensional data as in (2.111). A high value of τ means that the neighbour's rank is captured. In terms of comparison, the best dimensionality reduction method is the method with the highest value of τ.

3.1.1 Complexity Analysis

SDD needs to create two matrixes with $N \times N$ to store distances in both high and low dimensional spaces and another matrix that stores the difference $P - Q$ with $N \times N$, where P and Q are the similarity matrixes of high and low dimensional space data. In total, the computational

and memory complexity of SDD is $O(nN^2)$, since it needs to tune the $deg : 1 : n$, and $n = 15$ [92].

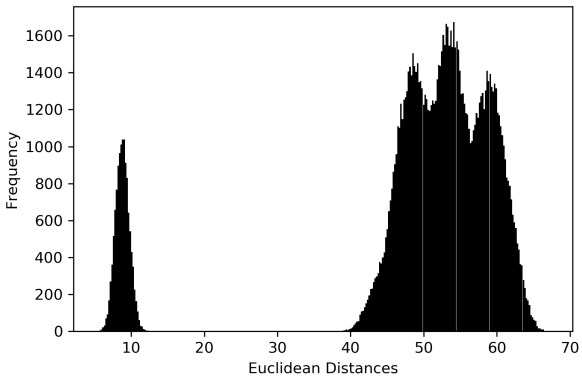
3.1.2 Implementation Guidance of SDD

The performance of SDD is related to the degree(s) of degree-distribution(s), and the selection of the degree of degree-distribution associates with 1) the high dimensional data distance distribution and 2) the dimensionality reduction purpose. In the case of data with a large fraction of large distances and small fractions of small distances, as shown in Fig. 3.1(a), employing small degree degree-distribution(s) is suggested. Degree-distributions with a small degree (i.e. *deg* 1, 2), has heavy tails, which means high sensitivity to large distances. High degree ($deg > 5$) degree-distribution(s) is suggested to be employed in datasets that have a large fraction of small distances (Fig. 3.1(c)), and medium degree degree-distribution(s) should be employed in datasets with a large fraction of medium distances (Fig. 3.1(b)). However, this is an intuitive judgement, and the simulations provided later will generate precise results. If for a user, the local structure of the data is more important than the global structure, we suggest employing high degree degree-distribution(s); otherwise, the employment of low degree degree-distribution(s) may be more beneficial. The degree of degree-distribution which captures the best structure of the data is named *best degree*.

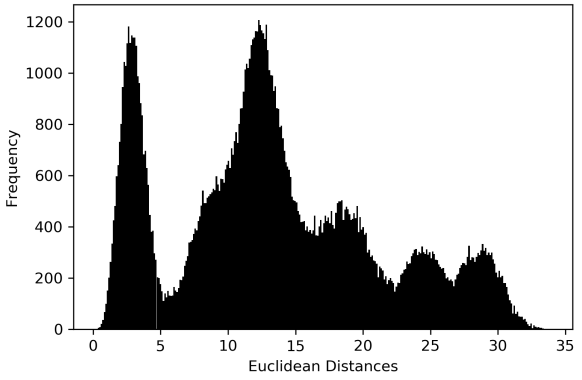
Defining the degree of a degree-distribution also depends on the distance range, and it has to be noted that degree-distributions are less sensitive to large distances. To solve this problem, it is proposed propose rescaling the distance ranges into the interval range from 0 to 1.

3.1.3 Rescaling Distance Range

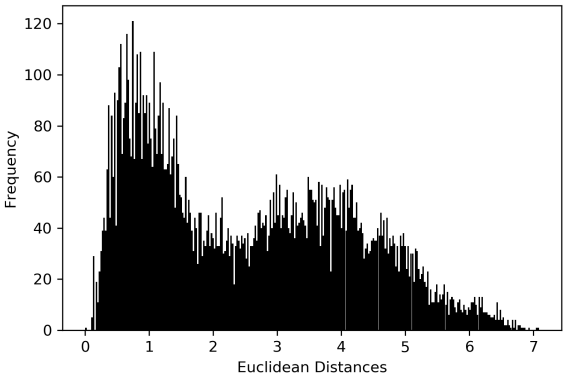
To rescale the pairwise distance range, is proposed dividing every single distance on pairwise distances with a decent positive number. The Euclidean distance between x_1, y_1 is calculated as $dis(x_1, y_1) = \sqrt{x_1^2 + y_1^2}$, whereas the Euclidean distance between $\alpha x_1, \alpha y_1$ is calculated: $dis(\alpha x_1, \alpha y_1) = \sqrt{(\alpha x_1)^2 + (\alpha y_1)^2} = \sqrt{(\alpha)^2(x_1)^2 + (\alpha)^2(y_1)^2} = \sqrt{(\alpha)^2((x_1)^2 + (y_1)^2)} =$



(a)



(b)



(c)

Figure 3.1: Three distance distributions.

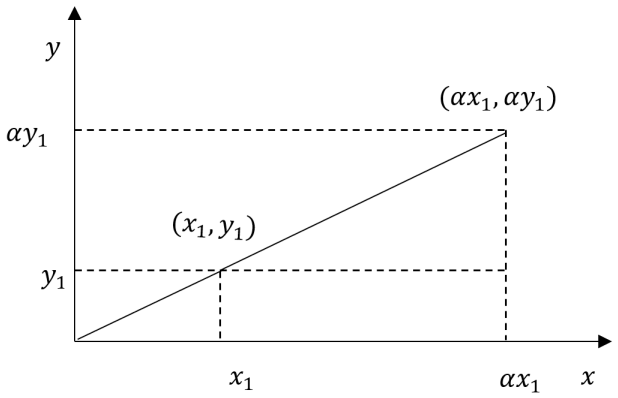
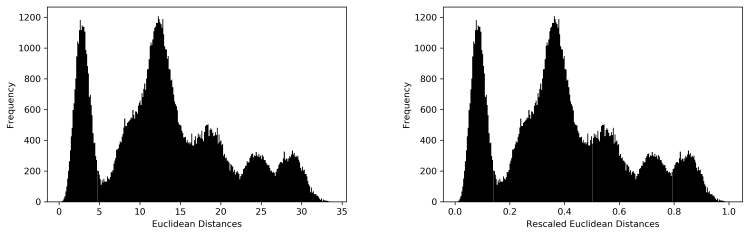


Figure 3.2: Scaled Euclidean distance.



(a) **(b)**

Figure 3.3: Distributions of Euclidean distances(a) and scaled Euclidean distances (b) of Make Blob data with 500 samples.

$\alpha\sqrt{((x_1)^2 + (y_1)^2)} = \alpha dis(x_1, y_1)$. As shown in Fig. 3.2 and proved above, if all sample values are scaled by a positive number α, the Euclidean distance calculated between the scaled samples also scales by the positive number α. Distributions of Euclidean distance and the scaled Euclidean distance can be visually seen in Fig. 3.3(a) and Fig. 3.3(b), respectively. In SDD, $\alpha = \frac{1}{\max dis(x_i, x_j)}$, due to the high sensitivity of the degree-distribution(s) in the value range between 0 and 1.

For some datasets, one degree-distribution is not sufficient to capture enough data structure, and therefore more degree-distributions are needed to be applied. To deal with this, is presented a multi-distribution-based approach Multi SDD (MSDD), as discussed in the following Section.

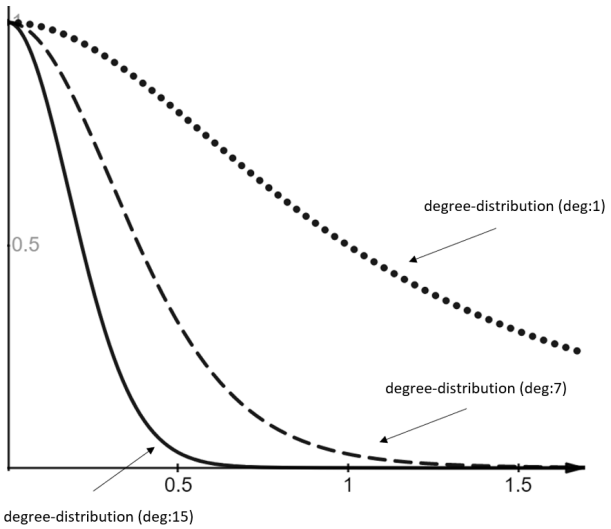


Figure 3.4: Three degree-disstributions.

3.2 Multi Same Degree Distribution (MSDD) Approach

This research proposes an extension of SDD named Multi Same Degree Distributions (MSDD). MSDD is based on SDD, but it adds more distributions to capture better the global or local data structure. In other words, SDD can be seen as the simple case of MSDD. All idea behind the MSDD is that employing one distribution may not be sufficient for capturing both local and global data structure. Using different degree-distributions may improve the scale of the maintained data structure since high-degree degree-distribution gives more priority to the short distances, and as the degree decreases to 1, more favours large distances, as also shown in Fig. 3.4, 3.5. The pseudocode of MSDD is demonstrated below in Algorithm 2.

MSDD employs n degree-distributions, as such n-objective functions must be optimised. Multi-objective optimisation problems are classically solved using scalarisation techniques [93, 94]. MSDD will be optimised using the composed Kullbak-Leibler (s) as in (3.7) via the scalarisation techniques [94]:

$$C_2 = a_1 \sum_{i \neq j} (p_1)_{ij} \log(\frac{(p_1)_{ij}}{(q_1)_{ij}} + ... + a_n \sum_{i \neq j} (p_n)_{ij} \log(\frac{(p_n)_{ij}}{(q_n)_{ij}} \tag{3.5}$$

To simplify the problem, each degree-distribution has been thought to have the same infuence (weight $a_m = 1$, $m = 1 : n$) in (3.7). So, the parameters to be tuned are the number of degree-

Algorithm 2 MSDD

Require: Input :

$X \in R^{NxD}$, calculate matrix DIS of pairwise distance of X and rescale into the range $[0, 1]$, number of iterations H, learning rate η, momentum α, number of degree-distributions n, degree deg_m, $Degrees = bestdeg_m$ from Algorithm I, $\tau_{actual} = \max(\tau)$, initial low dimensional data $Y^0 = y_1, ..., y_N \in N(0, 10^{-4}I)$, ϵ.

Step 1 :

Compute the high dimensional space similarities $(p_{deg_m})_{ij}$ using (3.3) and store in matrix P_{deg_m}.

Step 2 :

Compute the low dimensional space similarities $(q_{deg_m})_{ij}$ using (3.4) and store in matrix Q_{deg_m}.

Step 3 :

Compute the gradient $\frac{\delta C_2}{\delta\ yi}$ where C_2 defined in (3.5) is reformulated as: $C_2 = \sum_{m \notin Degrees} \sum_{i \neq j} (p_{deg_m})_{ij} \log(\frac{(p_{deg_m})_{ij}}{(q_{deg_m})_{ij}} + \sum_{m \in Degrees} \sum_{i \neq j} (p_{deg_m})_{ij} \log(\frac{(p_{deg_m})_{ij}}{(q_{deg_m})_{ij}}$.

Step 4 :

Minimize the objective function using the Gradient Descent optimisation algorithm and update the low dimensional space as: $Y^h = Y^{h-1} + \eta \frac{\delta C_2}{\delta y_i} + \alpha(Y^{h-1} - Y^{h-2})$.
The optimisation algorithm will stop either achieves the maximum number of iterations H or the Kullback-Leibler value is lower than the minimum threshhold ϵ.

Step 5 :

Add more degrees in cases $new \in \{best_{deg} - 1, best_{deg} + 1\}$: $if\ \tau_{new} < \tau_{actual}$, $Degrees = Degrees \bigcup deg_{m\ with\ \tau_{new}}$, $\tau_{actual} = \tau_{new}$.

Output :

Low dimensional space represenation $Y_{Degrees}$.

distributions n and the degree of each degree-distribution $deg_m, m = 1 : n$. The problem can be formulated as below:

$$C_2 = \sum_{m=1}^{n} \sum_{i \neq j} (p_{deg_m})_{ij} \log(\frac{(p_{deg_m})_{ij}}{(q_{deg_m})_{ij}} \tag{3.6}$$

$$loss_2 = \min(C_2) \tag{3.7}$$

3.2.1 Implementation Guidance of MSDD

MSDD has been projected on top of SDD but including more than one degree-distributions. It has been proposed at [92] using degree-distributions with one degree up or down the best degree, such as the combinations of $\{best_{deg}, best_{deg}+1\}$, $\{best_{deg}, best_{deg}-1\}$, and $\{best_{deg}, best_{deg}-1, best_{deg}+1\}$. The benefits of using one degree up or down the $best_{deg}$ is it can improve global structure by including $\{best_{deg}, best_{deg}-1\}$ or local structure by including$\{best_{deg}, best_{deg}+1\}$ or both of them by including $\{best_{deg}, best_{deg}-1, best_{deg}+1\}$. However, it is not guaranteed that adding more distributions on top of the $best_{deg}$ distribution can capture more data structure.

3.2.2 Complexity Analysis

MSDD computational complexity is higher than computational complexity of SDD and it is related to the number of degree-distributions involved. The computational and space complexity of MSDD is $O((n+3)N^2)$, where n is the number of degree-distributions, since MSDD after calculating the $best_{deg}$ (which takes $O(nN^2)$), it checks if the combinations of $\{best_{deg}, best_{deg}+1\}$, $\{best_{deg}, best_{deg}-1\}$, and $\{best_{deg}, best_{deg}-1, best_{deg}+1\}$, may generates better structure capturing. The number of degree-distributions used in MSDD will be that number that produces the highest value of the correlation coefficient Kendall's Tau (τ).

3.3 Parameter-free SDD

All benefits using the SDD comes from the degree-distribution with degree 1 ($deg = 1$), which is equivalent to Student-t distribution ($deg = 1$). Degree-distribution generates higher similarity values to short distances, and values decrease smoothly as distance increases. When distance increases infinitely, the degree-distribution with $deg = 1$ approaches to zero, making it unfeasible to maintain the large distances structures. To deal with this, Hajderanj et al. [92] proposed rescaling the pairwise distances of the original data into the range [0, 1]. It has been demonstrated that the original data X rescales by a single value (maximum value of pairwise distances), then the distribution of pairwise distances of the rescaled data is the same as the distribution of pairwise distances of the original data X, as shown in Fig 3.7. (a), (b), and (c). Rescaling the pairwise distances of the original data is the key to the success of SDD, which has been demonstrated to capture a good structure of the data; however, SDD still requires tuning the degree of degree-distribution, which normally ranges from 1 to 15. Degree-distributions with degrees $deg = 15$ and $deg = 7$ are not sensitive to distances between 0.5 and 1, and that means they can not capture the neighbourhood structure of far away data samples.

To evaluate if close (far away) neighbours in the original high-dimensional space are kept close (far away) in the embedded low dimensional space, two metrics are commonly used: *Trustworthiness* (2.116) and *Continuity* (2.117). Trustworthiness measures the far away data samples that embedd close in the low dimensional space, whereas Continuity measures close data samples that embedd far away in the low dimensional space. As shown in Fig 3.6. (a) and (b), measured by Trustworthiness and Continuity, degree-distribution with $deg = 1$, demonstrated a poorer performance in maintaining the local data structure than degree-distribution with $deg = 7$ and $deg = 15$ (where under consideration is a small number of neighbours). However, in situations where the number of neighbours under consideration is large, degree-distribution with $deg = 1$ shows a better performance in capturing the global data structure than the degree-distribution with $deg = 7$ and significantly better than degree-distribution with $deg = 15$.

Overall, using the degree-distribution with $deg = 1$ is similar to using the degree-distribution with the best degree ($deg = 7$). However, it can be shown that degree-distribution with $deg = 1$

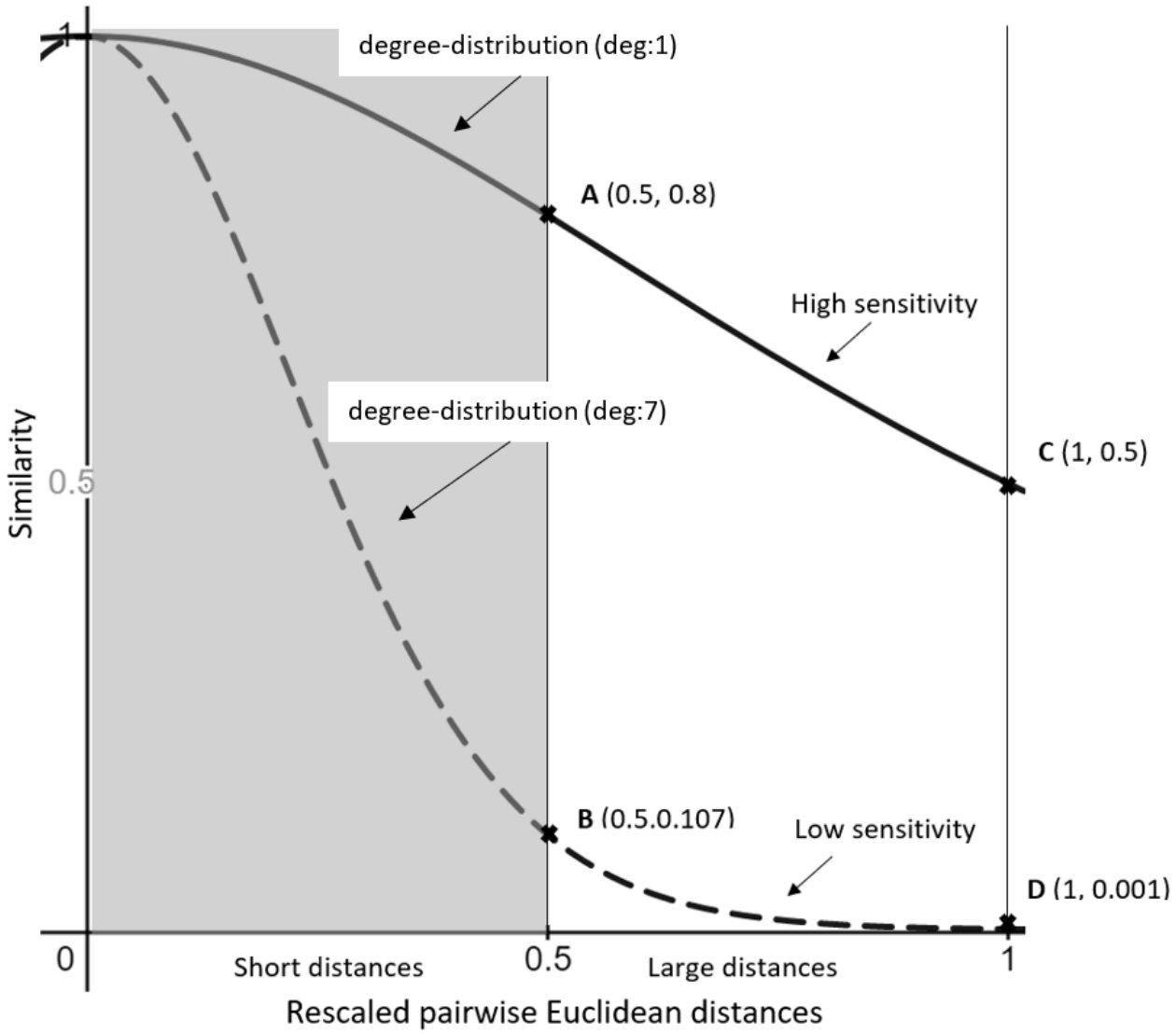
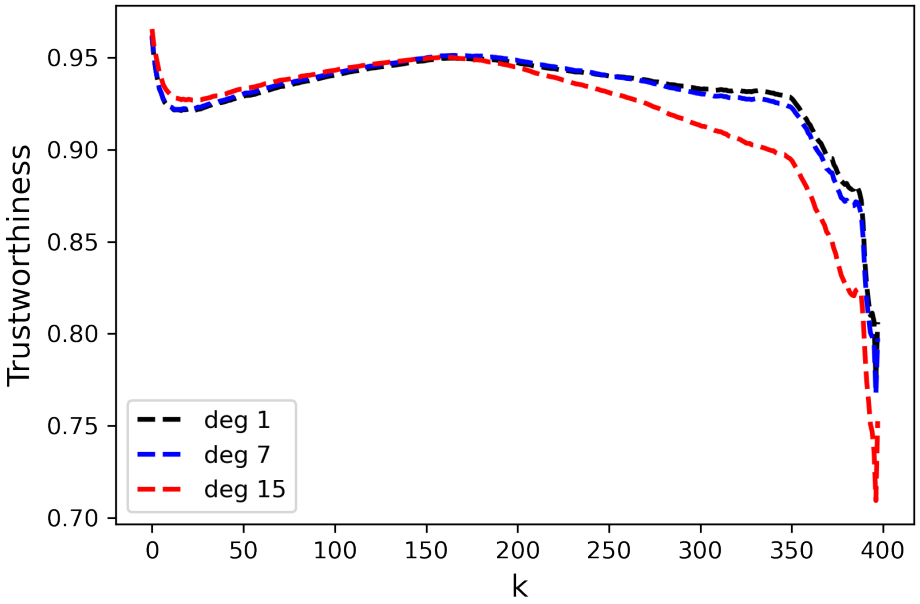
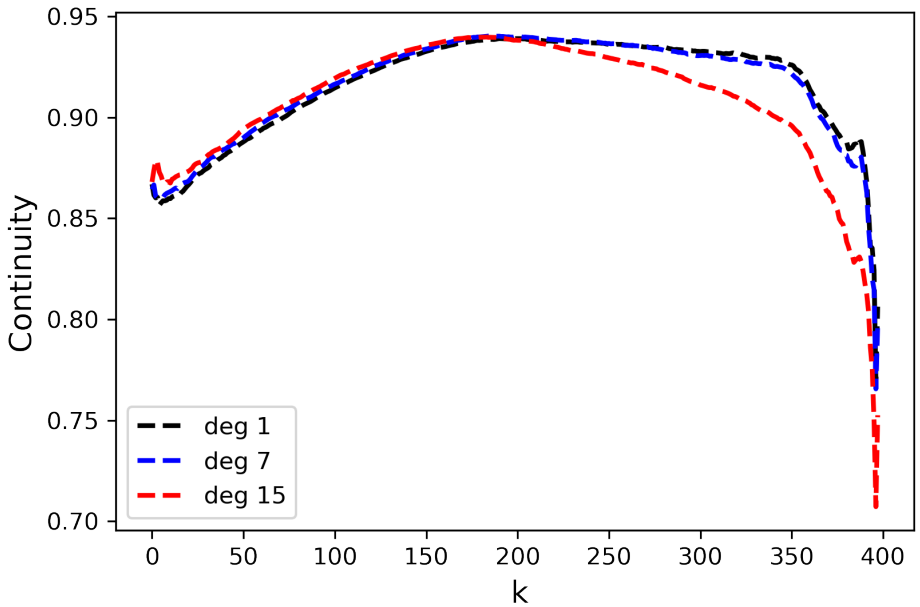


Figure 3.5: Two degree-distributions and the sensitivity to large pairwise distances.

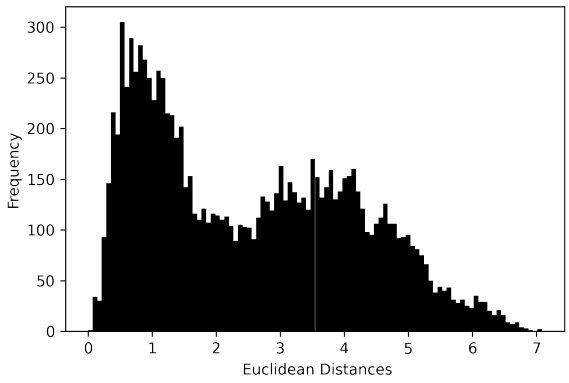


(a)

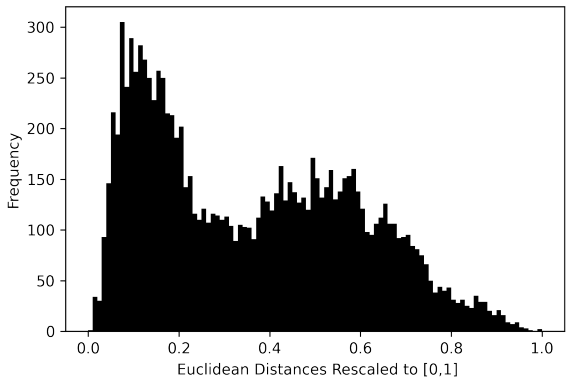


(b)

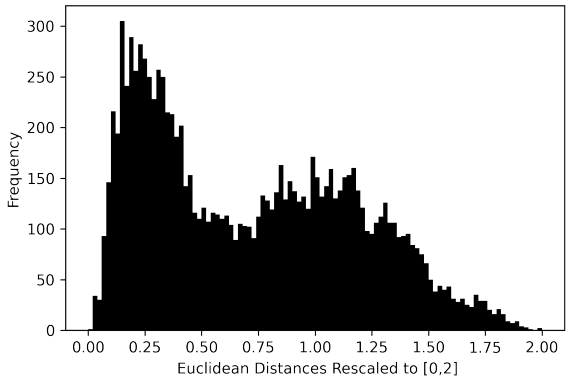
Figure 3.6: Trustworthiness (a), and Continuity (b) for SDD with degrees 1, 7 and 15 .



(a)



(b)



(c)

Figure 3.7: Euclidean Distance (a), Rescaled Euclidean distance to [0,1] (b) and Rescaled Euclidean distance to [0,2] (c).

does not perform as good as degree-distribution with $deg = 7$ in short distances (a small number of neighbours under consideration). To deal with that issue, it is proposed in the research to increase the range of pairwise distance of the original data in [0, 2].

3.3.1 Idea and Theoretical Proof

Rescaling pairwise distances in the range [0, 2] can generate a wider range of similarity, as shown in Fig. 3.8. The range of similarity generated by degree-distribution with $deg = 1$ having pairwise distances in [0, 1], ranges in the interval [1, 0.5], whereas the similarity generated by degree-distribution with $deg = 1$ having pairwise distances in [0, 2], ranges in a wider interval [1, 0.2]. Moreover, it can be theoretically proven that rescaling the pairwise distances of the original data in the range [0, 2] generates a wider range of similarity produced by degree distribution with $deg = 1$, as in Proposition 1 below.

Proposition 1 By increasing the rescaled distance range interval, the similarity range of degree-distribution $p_{ij} = \frac{(1+dis(x_i,x_j))^{-1}}{\sum_{k\neq l}(1+dis(x_k,x_l))^{-1}}$ also increases.

Proof

Let's define with $d_1(x_i, x_j)$ the rescaled distance of $dis(x_i, x_j)$ in the interval $[0, 1]$, and $d_2(x_i, x_j)$ the rescaled distance of $dis(x_i, x_j)$ in the interval $[0, 2]$ and $d_2(x_i, x_j) = 2 \times d_1(x_i, x_j)$, where $d_{10}(x_i, x_j) = 0$, $d_{11}(x_i, x_j) = 1$, $d_{20}(x_i, x_j) = 0$, and $d_{22}(x_i, x_j) = 2$. Let's also define with $[L_1, U_1]$ and $[L_2, U_2]$ the the similarity ranges of $[0, 1]$ and $[0, 2]$, respectively.

$$L_1 = \frac{(1 + d_{10}(x_i, x_j))^{-1}}{S_1} = \frac{(1 + 0)^{-1}}{S_1} = \frac{1}{S_1} = \frac{1}{\sum_{k\neq l}(1 + d_1(x_k, x_l))^{-1}} \tag{3.8}$$

where $S_1 = \sum_{k\neq l}(1 + d_1(x_k, x_l))^{-1}$

$$U_1 = \frac{(1 + d_{11}(x_i, x_j))^{-1}}{S_1} = \frac{(1 + 1)^{-1}}{S_1} = \frac{1}{2(S_1)} = \frac{1}{2\sum_{k\neq l}(1 + d_1(x_k, x_l))^{-1}} \tag{3.9}$$

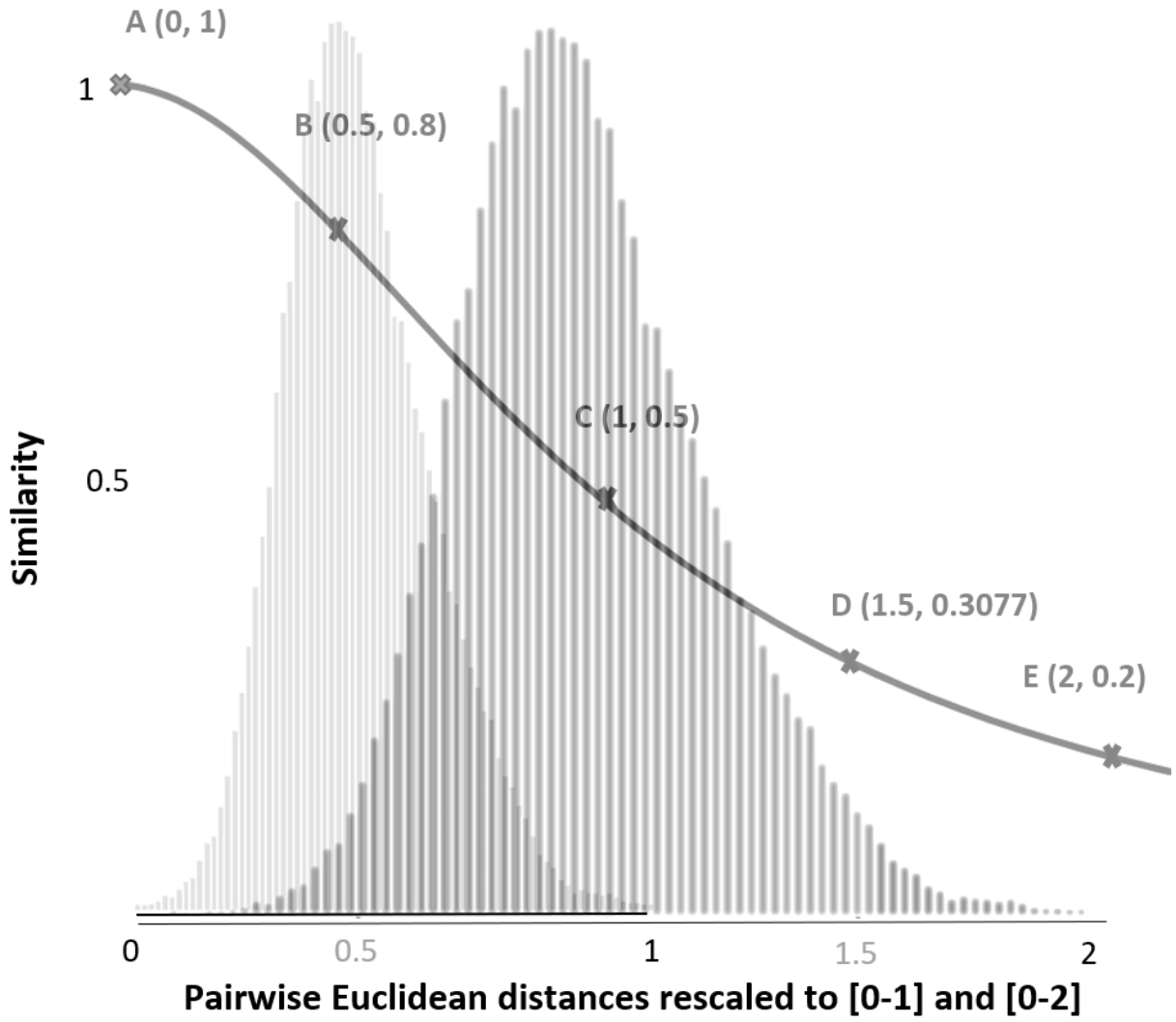


Figure 3.8: Degree-distribution ($deg = 1$) in the pairwise distances rescaled in [0-2].

$$L_2 = \frac{(1 + d_{20}(x_i, x_j))^{-1}}{S_2} = \frac{(1+0)^{-1}}{S_2} = \frac{1}{S_2} = \frac{1}{\sum\limits_{k \neq l}(1 + 2d_1(x_k, x_l))^{-1}} \tag{3.10}$$

where $S_2 = \sum\limits_{k \neq l}(1 + d_2(x_k, x_l))^{-1} = \sum\limits_{k \neq l}(1 + 2d_1(x_k, x_l))^{-1}$

$$U_2 = \frac{(1 + d_{22}(x_i, x_j))^{-1}}{S_2} = \frac{(1+2)^{-1}}{S_2} = \frac{1}{3(S_2)} = \frac{1}{3\sum\limits_{k \neq l}(1 + 2d_1(x_k, x_l))^{-1}} \tag{3.11}$$

Based on Eqs. (3.8) and (3.9), the interval of similarity is:

$[L_1, U_1] = [\sum\limits_{k \neq l}(1 + d_1(x_k, x_l)), \frac{1}{2\sum\limits_{k \neq l}(1+d_1(x_k,x_l))^{-1}}]$,
and based on Eqs. (3.10) and (3.11) the interval of similarity is:

$[L_2, U_2] = [\frac{1}{\sum\limits_{k \neq l}(1+2d_1(x_k,x_l))^{-1}}, \frac{1}{3\sum\limits_{k \neq l}(1+2d_1(x_k,x_l))^{-1}}]$.

As such, the length of the interval is

$$L_1 - U_1 = \frac{1}{\sum\limits_{k \neq l}(1 + d_1(x_k, x_l))^{-1}} - \frac{1}{2\sum\limits_{k \neq l}(1 + d_1(x_k, x_l))^{-1}} = \frac{1}{2\sum\limits_{k \neq l}(1 + d_1(x_k, x_l))^{-1}} \tag{3.12}$$

$$L_2 - U_2 = \frac{1}{\sum\limits_{k \neq l}(1 + 2d_1(x_k, x_l))^{-1}} - \frac{1}{3\sum\limits_{k \neq l}(1 + 2d_1(x_k, x_l))^{-1}} = \frac{2}{3\sum\limits_{k \neq l}(1 + 2d_1(x_k, x_l))^{-1}} \tag{3.13}$$

To proof that $[L_2, U_2]$ is wider than $[L_1, U_1]$, then based on Eqs. (3.12) and (3.13) it has to be proven that

$$\frac{2}{3\sum\limits_{k \neq l}(1 + 2d_1(x_k, x_l))^{-1}} > \frac{1}{2\sum\limits_{k \neq l}(1 + d_1(x_k, x_l))^{-1}} \Longrightarrow$$

$$\frac{2}{3\sum\limits_{k \neq l}(1 + 2d_1(x_k, x_l))^{-1}} - \frac{1}{2\sum\limits_{k \neq l}(1 + d_1(x_k, x_l))^{-1}} > 0$$

$$(L_2 - U_2) - (L_1 - U_1) =$$

$$= \frac{2}{3\sum_{k\neq l}\frac{1}{(1+2d_1)}} - \frac{1}{2\sum_{k\neq l}\frac{1}{(1+d_1)}}$$

$$= \frac{4\sum_{k\neq l}\frac{1}{(1+d_1)} - 3\Big(\sum_{k\neq l}\frac{1}{(1+2d_1(x_k,x_l))}\Big)}{6\sum_{k\neq l}\frac{1}{(1+2d_1(x_k,x_l))}\sum_{k\neq l}\frac{1}{(1+d_1(x_k,x_l))}}$$

$$= \frac{\sum_{k\neq l}\frac{4}{(1+d_1(x_k,x_l))} - \Big(\frac{3}{(1+2d_1)}\Big)}{6\sum_{k\neq l}\frac{1}{(1+2d_1)}\frac{1}{(1+d_1)}}$$

$$= \frac{\sum_{k\neq l}\frac{4((1+2d_1))-3((1+d_1))}{(1+d_1)(1+2d_1(x_k,x_l))}}{6\sum_{k\neq l}\frac{1}{(1+2d_1(x_k,x_l))(1+d_1(x_k,x_l))}}$$

$$= \frac{\sum_{k\neq l}\frac{1}{(1+d_1(x_k,x_l))(1+2d_1(x_k,x_l))}\sum_{k\neq l}4((1+2d_1(x_k,x_l))) - 3((1+d_1(x_k,x_l)))}{6\sum_{k\neq l}\frac{1}{(1+2d_1(x_k,x_l))(1+d_1(x_k,x_l))}}$$

$$= \frac{\sum_{k\neq l}4(1+2d_1(x_k,x_l)) - 3((1+d_1(x_k,x_l)))}{6}$$

$$= \frac{\sum_{k\neq l}(1+5d_1(x_k,x_l))}{6}$$

Since $\frac{\sum_{k\neq l}(1+5d_1(x_k,x_l))}{6}$, then the similarity range provided by pairwise distances rescaled in the range [0, 2] is wider than the similarity range provided by pairwise distances rescaled in the range [0, 1]. ■

However, a further question arises: are the similarity ranges of both short and large distances expanded the same as the range of pairwise distances increases? To examine the question, consider short distances $d_{1\,short} \in [L_1, H_1]$, $d_{1\,large} \in]H_1, U_1]$, $d_{2\,short} \in [L_2, H_2]$, and $d_{1\,large} \in]H_2, U_2]$ as in Fig. 3.9.

To evaluate whether short and large distances have been affected mainly by increasing the range of pairwise distances, has defined and proven in Proposition 2.

Proposition 2 By increasing the range of rescaled pairwise distances, the similarity range of

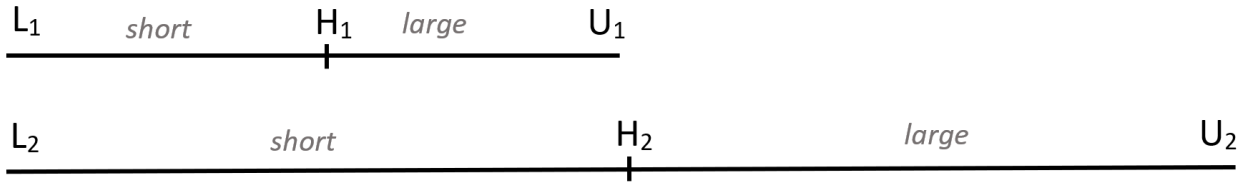


Figure 3.9: Two data segements.

short distances increases more than the similarity range of large distances.

Proof

Since L_1, U_1, L_2, U_2 have been calculated in the Proposition 1, then let calculates the H_1 and H_2 which are the middle samples of $[L_1, U_1]$ and $[L_2, U_2]$, respectively.

$$H_1 = \frac{(1 + \frac{(d_{10}(x_i,x_j))+(d_{11}(x_i,x_j))}{2})^{-1}}{S_1} = \frac{2}{3(S_1)} = \frac{2}{3\sum\limits_{k\neq l}(1 + d_1(x_k, x_l))^{-1}} \tag{3.14}$$

$$H_2 = \frac{(1 + \frac{(d_{20}(x_i,x_j))+(d_{22}(x_i,x_j))}{2})^{-1}}{S_2} = \frac{1}{2(S_2)} = \frac{1}{2\sum\limits_{k\neq l}(1 + 2d_1(x_k, x_l))^{-1}} \tag{3.15}$$

Finally,

$\Big[L_1, H_1\Big] = \Big[\frac{1}{\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}}, \frac{2}{3\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}}\Big]$, and

$\Big[L_2, H_2\Big] = \Big[\frac{1}{\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}}, \frac{1}{2\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}}\Big]$

Then,

$$L_1 - H_1 = \frac{1}{\sum\limits_{k\neq l}(1 + d_1(x_k, x_l))^{-1}} - \frac{2}{3\sum\limits_{k\neq l}(1 + d_1(x_k, x_l))^{-1}} = \frac{1}{3\sum\limits_{k\neq l}(1 + d_1(x_k, x_l))^{-1}} \tag{3.16}$$

$$L_2 - H_2 = \frac{1}{\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} - \frac{1}{2\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} = \frac{1}{2\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \quad (3.17)$$

To proof that $[L_2, H_2]$ is wider than $[L_1, H_1]$, then based on Eqs. (3.16) and (3.17) have to be checked if $\frac{1}{2\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} > \frac{1}{3\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}}$ which is equivalent with $\frac{1}{2\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} - \frac{1}{3\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}} > 0$

$$\begin{aligned}
&(L_2 - H_2) - (L_1 - H_1) \\
&= \frac{1}{2\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} - \frac{1}{3\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}} \\
&= \frac{3\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1} - 2(\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1})}{6\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&= \frac{3\frac{1}{\sum\limits_{k\neq l}(1+d_1(x_k,x_l))} - 2\frac{1}{\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))}}{6\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&= \frac{\frac{3\sum\limits_{k\neq l}(1+2d_1(x_k,x_l)) - 2\sum\limits_{k\neq l}(1+d_1(x_k,x_l))}{\sum\limits_{k\neq l}(1+d_1(x_k,x_l))\sum\limits_{k\neq l}(d_1(x_k,x_l))}}{6\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&= \frac{\frac{\sum\limits_{k\neq l}(1) + 4\sum\limits_{k\neq l}(d_1(x_k,x_l))}{\sum\limits_{k\neq l}(1+d_1(x_k,x_l))\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))}}{6\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&= \frac{\sum\limits_{k\neq l}(1) + 4\sum\limits_{k\neq l}(d_1(x_k,x_l))\frac{1}{\sum\limits_{k\neq l}(1+d_1(x_k,x_l))\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))}}{6\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&= \frac{\sum\limits_{k\neq l}(1) + 4\sum\limits_{k\neq l}(d_1(x_k,x_l))}{6}
\end{aligned}$$

Also, based on Eqs. (3.14) and (3.9) $\Big[H_1, U_1\Big] = \Big[\frac{2}{3\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}}, \frac{1}{2\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}}\Big]$, and based

on Eqs. (3.15) and (3.11) $\Big[H_2, U_2\Big] = \Big[\frac{1}{2\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}}, \frac{1}{3\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}}\Big]$ then,

$$H_1 - U_1 = \frac{2}{3\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}} - \frac{1}{2\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}} = \frac{1}{6\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}} \quad (3.18)$$

$$H_2 - U_2 = \frac{1}{2\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} - \frac{1}{3\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} = \frac{1}{6\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \quad (3.19)$$

To proof that $[H_2, U_2]$ is wider than $[H_1, U_1]$, then based on Eqs. (3.18) and (3.19) has to be checked if $\frac{1}{2\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} > \frac{1}{3\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}}$ which is equivalent with $\frac{1}{6\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} - \frac{1}{6\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}} > 0$

$$\begin{aligned}
(H_2 - U_2) - (H_1 - U_1) = & \\
&= \frac{1}{6\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} - \frac{1}{6\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}} \\
&= \frac{\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1} - (\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1})}{6\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&= \frac{\frac{1}{\sum\limits_{k\neq l}(1+d_1(x_k,x_l))} - \frac{1}{\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))}}{6\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&= \frac{\frac{\sum\limits_{k\neq l}(1+2d_1(x_k,x_l)) - \sum\limits_{k\neq l}(1+d_1(x_k,x_l))}{\sum\limits_{k\neq l}(1+d_1(x_k,x_l))\sum\limits_{k\neq l}(d_1(x_k,x_l))}}{6\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&= \frac{\sum\limits_{k\neq l}(d_1(x_k,x_l))\frac{1}{\sum\limits_{k\neq l}(1+d_1(x_k,x_l))\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))}}{6\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&= \frac{\sum\limits_{k\neq l}(d_1(x_k,x_l))}{6}
\end{aligned}$$

As proved above, increasing the pairwise distances ranges from [0, 1] to [0, 2], the similarity range of short distances is increased by $\frac{\sum\limits_{k\neq l}(1)+4\sum\limits_{k\neq l}(d_1(x_k,x_l))}{6}$ and similarity range of large distances increased by $\frac{\sum\limits_{k\neq l}(d_1(x_k,x_l))}{6}$. Having a wider interval of similarity means a small change in distance

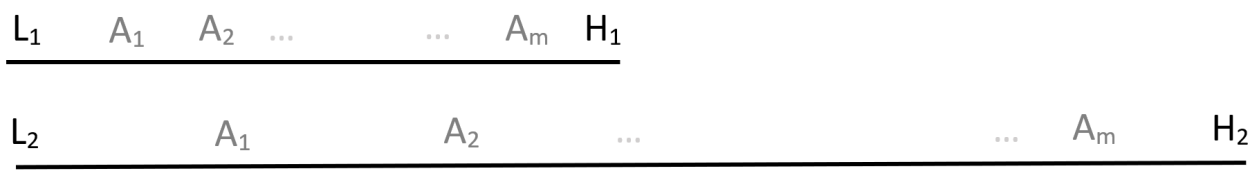


Figure 3.10: Data samples $A_1, A_2, ..., A_m$ whose distances is in in the range $[L_1, H_1]$ and $[L_2, H_2]$.

derives a bigger change in similarity. As such, data samples $A_1, A_2, ..., A_m$ have pairwise distances in the interval $[L_1 = 0, H_1 = 0.5]$ if pairwise distances are rescaled in range $[0, 1]$ and in and in interval $[L_0 = 0, H_2 = 1]$ if pairwise distances are rescaled in range $[0, 2]$. ■

Based on the proof of Proposition 2, the interval of similarity between data samples $A_1, A_2, ..., A_m$ with pairwise distances rescaled in the interval $[L_0 = 0, H_2 = 1]$ will be $\frac{\sum_{k \neq l}(1) + 4 \sum_{k \neq l}(d_1(x_k, x_l))}{6}$ wider than the interval of similarity between data samples $A_1, A_2, ..., A_m$ with pairwise distances rescaled in the interval $[L_1 = 0, H_1 = 0.5]$. So, if data samples A_1 and A_2 are close and A_1 and A_3 are far away, with pairwise distances $d_1(A_1, A_2) = 0$, $d_1(A_1, A_3) = 5$ and $d_2(A_1, A_2) = 0$, $d_2(A_1, A_3) = 1$, then is more possible that datapoints A_1, A_2 and A_3 will maintain their structure using pairwise distances rescaled to the interval [0, 2] rather than scaled in the interval [0, 1], due to the huge difference provided between small distances similarities and large distance similarities.

In other words, a wider similarity interval means a better-maintained structure. It is also visible that increasing the pairwise distance range has an impact more on short distances than on large distances. Overall, the increase of the rescaled range has a negative impact on capturing local data structure, which is one of the disadvantages of using degree-distribution with $deg = 1$ in the rescaled distance range [0, 1].

Additionally, based on Proposition 2, if the range of rescaled pairwise distances increases to [0, 2], the global structure destroys. Consequently, rescaling the pairwise distances in the interval [0, 3] or [0, 4] may improve the local structure maintenance. However, it destroys the maintenance of the global data structure because degree-distribution converges to zero when the pairwise distances increase. In conclusion, this research proposes rescaling original data in the interval [0, 2] due to the sensitivity that degree-distribution with $deg = 1$ has in this

interval.

Algorithm 3 Parameter-free SDD

Require: Input :

$X \in R^{NxD}$, calculate matrix DIS of pairwise distance of X and rescale into the range $[0, 2]$, number of iterations H, learning rate η, momentum α, initial low dimensional data $Y^0 = y_1, ..., y_N \in N(0, 10^{-4}I)$, ϵ.

Step 1 :

Compute the high dimensional space similarities (p_{ij}) using (3.22) and store them in P.

Step 2 :

Compute the low dimensional space similarities (q_{ij}) using (3.23) and store them in Q.

Step 3 :

Compute the gradient $\frac{\delta C_1}{\delta\ yi}$ where C_1 is defined in (3.20).

Step 4 :

Minimize the objective function using the Gradient Descent optimisation algorithm: $Y^h = Y^{h-1} + \eta\frac{\delta C_2}{\delta y_i} + \alpha(Y^{h-1} - Y^{h-2})$.

The optimisation algorithm will stop either achieves the maximum number of iterations H or the Kullback-Leibler value is lower than the minimum threshhold ϵ.

Output :

Low dimensional space represenation Y.

As such, it is proposed to use SDD with degree $(deg = 1)$ in the rescaled pairwise range in [0, 2]. Using SDD with $deg = 1$ in the rescaled range [0, 2] is named as parameter-free SDD as shown in Algorithm 3, and like SDD, it uses Kullback-Leibler to approximate the degree-distribution in the low dimensional space with the degree-distribution in the high dimensional space:

$$C_1 = \sum_{i \neq j} p_{ij} \log(\frac{p_{ij}}{q_{ij}}) \tag{3.20}$$

Parameter-free SDD intends to minimize the cost function C_1 as :

$$loss_1 = \min(C_1) \tag{3.21}$$

where

$$p_{ij} = \frac{(1 + dis(x_i, x_j))^{-1}}{\sum_{k \neq l}(1 + dis(x_k, x_l))^{-1}} \tag{3.22}$$

$$q_{ij} = \frac{(1 + dis(y_i, y_j))^{-1}}{\sum_{k \neq l}(1 + dis(y_k, y_l))^{-1}} \tag{3.23}$$

3.3.2 Complexity Analysis

Parameter-free SDD needs to create two matrixes with $N \times N$ to store distances in both high and low dimensional spaces and another matrix that stores the difference $P - Q$ with $N \times N$. In total, the computational and space complexity of parameter-free SDD is $O(N^2)$ and is significantly less than the computational and space complexity of SDD and MSDD.

3.4 Parametric SDD

Although SDD is an excellent structure capturing method, it is still not feasible for the out-of-the sample data. Producing a parametric SDD is beneficial in terms of saving computational time and resources. Most of the dimensionality reduction techniques are non-parametric methods, and the parametric methods are PCA and RBM. PCA is one of the most famous parametric dimensionality reduction methods; however, it is a linear method that favours preserving global data structure at the expense of neglecting local data structure. Also, RBMs, a parametric method, favours capturing the global data structure, and it is a more complicated method due to the number of parameters required to tune. RBMs were proposed [95] to make t-SNE a parametric method. Parametric t-SNE is intended to maintain the local data structure, whereas RBMs maximises the data covariance, and that it means it captures the global data structure. And as a result, parametric t-SNE is ineffective in preserving well-separated clusters. It contrasts with RBM's objective function that maximises the variance. To deal

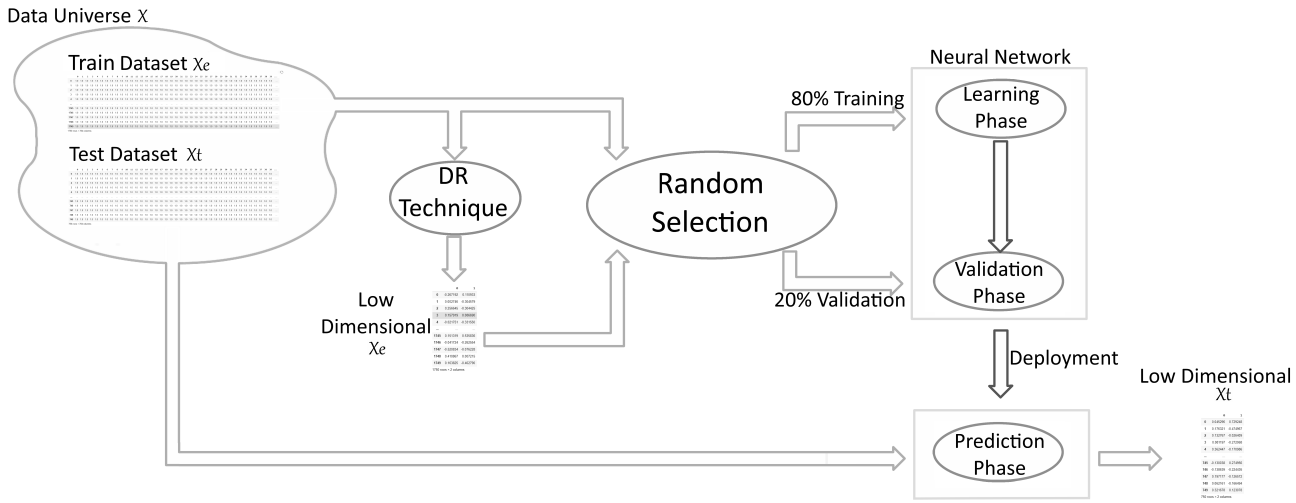


Figure 3.11: Framework of Learning Projection.

with this problem, using supervised learning with neural networks was proposed [96] to make t-SNE a parametric method. A neural network was trained to learn the two-dimensional data generated by a given dimensionality reduction method (t-SNE). This approach has been very effective in using high dimensional data in data structure capturing, scalability, and simplicity of genericity [95]. It minimises the distance between the two-dimensional data generated by the standard method and the two-dimensional data generated by the trained neural network.

Motivated by the effectiveness of using neural networks (ANN) to learn how to transform the original high-dimensional data to its corresponding low-dimensional data, parametric SDD is designed to use ANN to mimic the low-dimensional data generated from SDD in out of sample data. Note that SDD is an excellent method for capturing the structure of complex, heavily folded data, and the parametric method will mimic the results generated by the SDD. As a result, the low dimensional data generated by parametric SDD will have captured better the data structure than any other parametric method. The logical flowchart of the project for training an NN to learn an embedding is shown in Fig. 3.11. A successfully trained NN can be used to embed any new data, and therefore, this makes SDD a parametric method. In other words, the trained NN provides an explicit model to estimate the implicant embedded formed by SDD.

To testify the neural network, the dataset X will be split into training set X_e and testing set X_t, where X_e will be used to train the neural network, whereas X_t to test the neural network. The

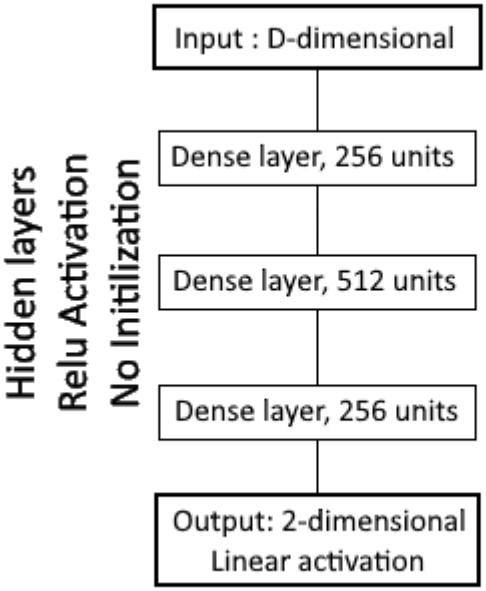


Figure 3.12: Network architecture employed.

neural network uses X_e and the two-dimensional data generated by dimensionality reduction techniques $DR(X_e)$. Each datasets used is randomly split into 80% training the network and 20% to validate the network. After the ANN has achieved satisfactory results in terms of classification accuracy, the network be used to embedd will predict X_t.

The architecture of the ANN employed is shown in Fig. 3.12, and it has three fully connected hidden layers with 256, 512 and 256 units, respectively, using the ReLU activation function. The last layer has two elements and uses a sigmoid function to encode the two-dimensional projection, scaled to the interval [0,1]. The optimisation method used for the training is the ADAM optimiser, a derivate of the stochastic gradient descent. The training will last up to 80 epochs and stop when there is no significant change to the loss in three successive epochs. The cost function used is the Mean Squared Error (MSE), can be expressed as following:

$$MSE = \frac{1}{N}\sum_{i=1}^{N} \|DR(X_e)_i - NN(X_e)_i\|^2 \tag{3.24}$$

where $DR(X_e)_i$, $NN(X_e)_i$ are the ground truth of two dimensional data generated by the dimensionality reduction method (DR) and the network (NN), respectively.

3.5 Chapter Summary

This Chapter has been proposed three nonlinear dimensionality reduction techniques, SDD, MSDD and parameter-free SDD. SDD employs the same degree-distribution in high and low dimensional spaces and tunes degree-distributions to generate the highes Kendall's Tau (the representation of structure capturing). It can unfold heavily curved manifold data very well, improves the global data structures compared with other methods such as t-SNE, Umap, Trimap, DD-HDS. Also, SDD saves computational time since it does not require tuning the number of neighbours, perplexity, but instead, it requires tuning the degree, which makes SDD a significantly less costly process than other methods. MSDD is s SDD based method, such that it has all benefits of SDD. However, it changes from SDD in two aspects: 1) it captures better the local and global data structure, and 2) it is more expensive than SDD as it employs more than one distribution. However, in the previous section of MSDD, an implementation guide clarifies how many distributions and which distribution to add on top of the distribution employed by SDD. However, there is no sure if adding distribution improves the structure maintenance. In the worse example, MSDD will generate the same low dimensioanl data as SDD in high computational time. The third approach is parameter-free SDD, which is similar to SDD, but it does not require tuning the degree, and consequently, it saves computational time. Also, parameter-free SDD changes from SDD in the way it rescales the similarity of high dimensional data in the range [0, 2] instead of the range [0, 1] that SDD does. Also, here is the proposed parametric SDD, which can generalise the low dimensional data produced by SDD for out of sample data.

All the proposed approaches performances will be demonstrated and compared with other methods in the next Chapter.

Chapter 4

Experiments and Discussions on Developed Approaches

This Chapter is organised into five Sections, experimental results of SDD, experimental results of MSDD, experimental results of parameter-free SDD, experimental results parametric SDD, and Chapter Summary. The four first Sections show the experimental results of each of the proposed approaches, evaluated by the scale of the maintained data structure and computational time. The maintained data structure has been measured considering Kendall's Tau, Co-ranking matrix, Trustworthiness, Continuity, and LCMC. The last Section summarises the main findings of experimental results.

4.1 Experimental Results of SDD

In this Section, the proposed method SDD is tested and compared with several benchmark dimensionality reduction techniques: PCA, MDS, Isomap, LLE, t-SNE, UMAP, and Trimap, using several typical benchmark datasets including Iris, Breast Cancer, Swiss roll, and MNIST. The fourth considered datasets represent different fractions of pairwise distances, such as Breast Cancer and Iris datasets have a large fraction with small distances. In contrast, Swiss Roll and MNIST datasets have the most significant fraction of medium to large distances. Thus, by

considering these fourth datasets, we demonstrate the strengths and weaknesses of the proposed algorithms since their performances closely depend on the data distance distribution. Also, the considered datasets are small (with a small number of data samples) because considering large datasets is very time-consuming for methods such as t-SNE, UMAP, Isomap due to the requirenment of parameter tuning.

All algorithms were implemented in Python with the same number of iterations of 2000. PCA, MDS, Isomap, LLE, LE, and t-SNE, were implemented using their Sklearn versions, and for UMAP[1], Trimap[2], SDD (MSDD and parameter-free SDD) [3], their GitHub versions were applied. For Isomap, LE, UMAP, and t-SNE, the parameter k (pr for t-SNE) was tuned in the range $(1, N-1)$, to find an appropriate number of neighbours which could produce the best low dimensional representation in relation to the structure capturing as shown in Figs. 4.2, 4.7, 4.12, and 4.17, where has been demonstrated that tuning those parameters has a huge impact of the scale of the maintained data structure for each considered dataset, respectively. For LLE, in the MNIST dataset, the number of neighbours k was tuned up to 1000 due to the memory problem. TriMap also failed to obtain a number of neighbours of more than 199, so the number of neighbours was tuned up to 198.

The effectiveness of each of the methods was evaluated using the Co-ranking matrix and τ (Kendall's Tau). The Co-ranking matrix indicates a perfect mapping if the matrix is diagonal, and the off-diagonal entries are the errors. τ takes values between -1 and 1, and when τ is 1, there exists a perfect correlation between ranks corresponding to an ideal mapping. The performance of each method in terms of Kendall's Tau τ along with the computational time t (in seconds) and the number of neighbours k (perplexity pr for t-SNE) presented in Tables 4.1, 4.2, 4.3 and 4.4, respectively for four considered datasets Iris, Breast Cancer, Swiss Roll and MNIST. Visualizations of the four considered datasets have been presented in Figs. 4.3, 4.8, 4.13, and 4.18, respectively.

[1]https://github.com/lmcinnes/umap
[2]https://github.com/eamid/trimap
[3]thttps://github.com/hajderal/SDD

4.1.1 Iris data

The first dataset considered is the Iris dataset, which contains 150 flowers and 4 attributes for each (length and width of petal and sepal). There are three different types of flowers and fifty samples per each. The distribution of pairwise distances for Iris dataset are presented in Fig. 4.1, where is presented by a balance on short, medium and large distances. PCA,

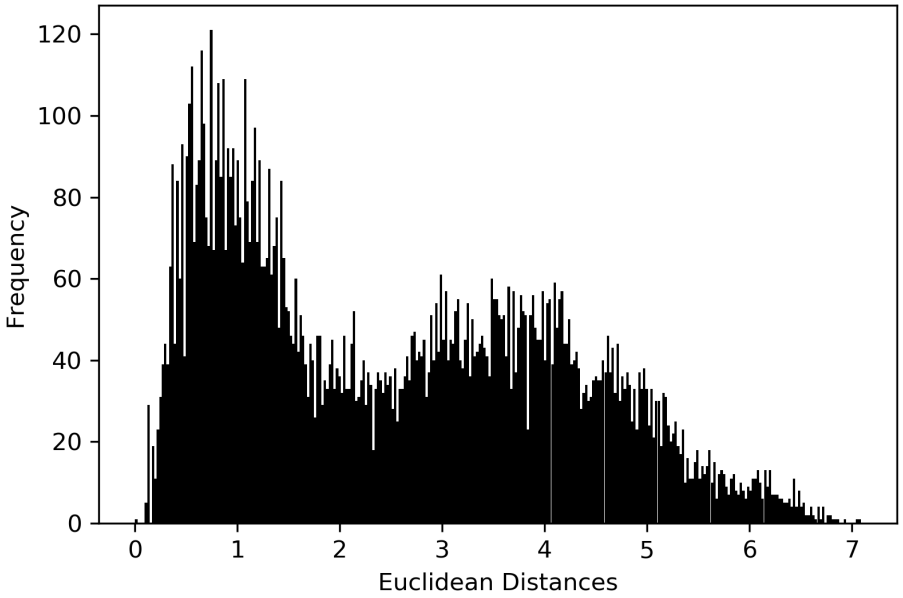


Figure 4.1: Euclidean distance distribution of Iris dataset.

MDS, Isomap, LLE, LE, t-SNE, UMAP, Trimap and SDD have been implemented to the Iris dataset. The parameters ($k : 1, N-1$, $pr : 1, N-1$ and $deg : 1, 15$) have tunned for Isomap, LLE, LE, t-SNE, UMAP, Trimap and SDD, to reveal the best low dimensional representation in terms of structure capturing, evaluated by Kendall's Tau coefficient as shown in Fig. 4.2. As shown in Fig. 4.2, tuning parameters such as k, pr or deg is crucial in the performance of each method, demonstrated by fluctuations in the τ values per each method. The best performances of considered methods in terms of τ are presented in Table 4.1, also described by the computational time.

From the simulation results, the best method with the highest τ of 0.9673 (Table 4.1) was SDD (deg: 8) with two-dimensional visualisation in Fig 4.3 (a). The highest performance of SDD is also confirmed by the Co-ranking matrix, shown in Fig 4.4(a), with fewer off-diagonal entries

in the top-centre Sections, which indicates good short and medium distance preservations. However, the Co-ranking matrix of SDD (*deg*: 8) has more off-diagonal entries than the Co-ranking matrixes of Isomap shown in Fig. 4.4 (p) and PCA shown in Fig. 4.4 (k), in the bottom right Sections. Thus, for the Iris dataset, SDD (*deg*: 8) performed better than the other methods of local structure capturing, and it performed similarly with Isomap and PCA for global structure-preserving.

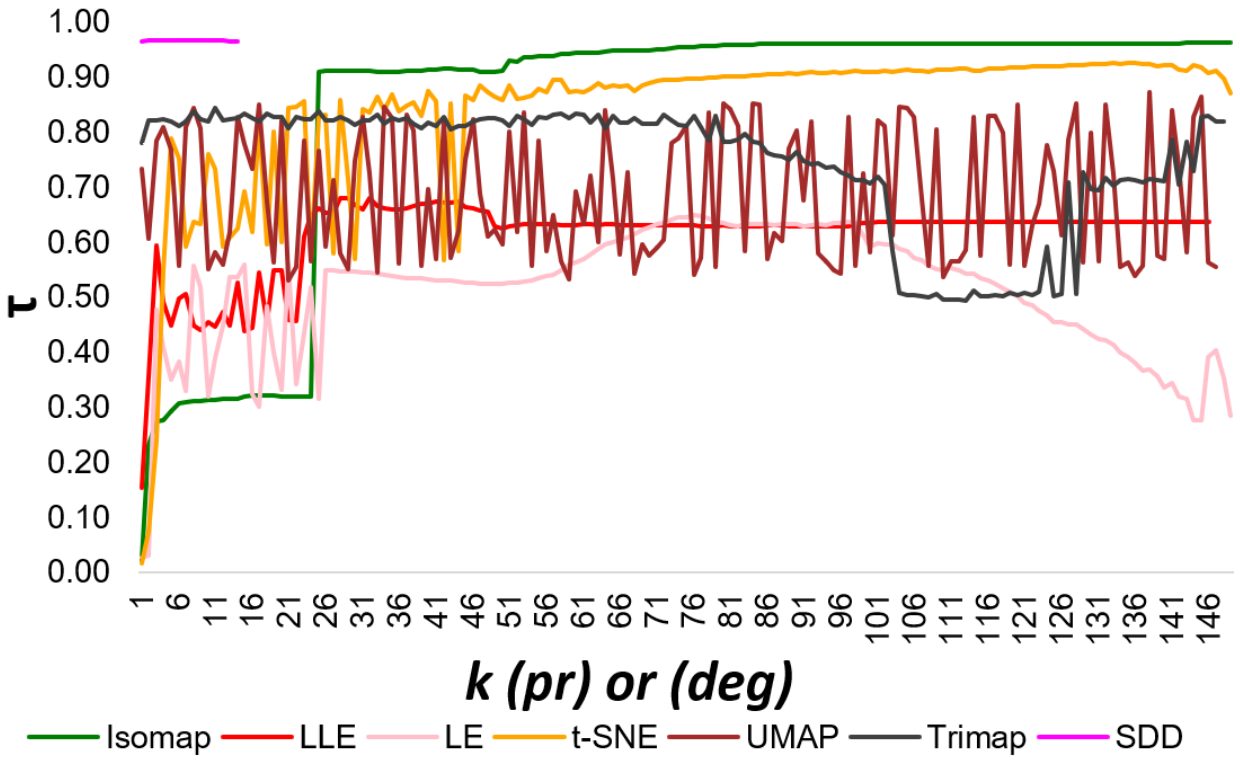


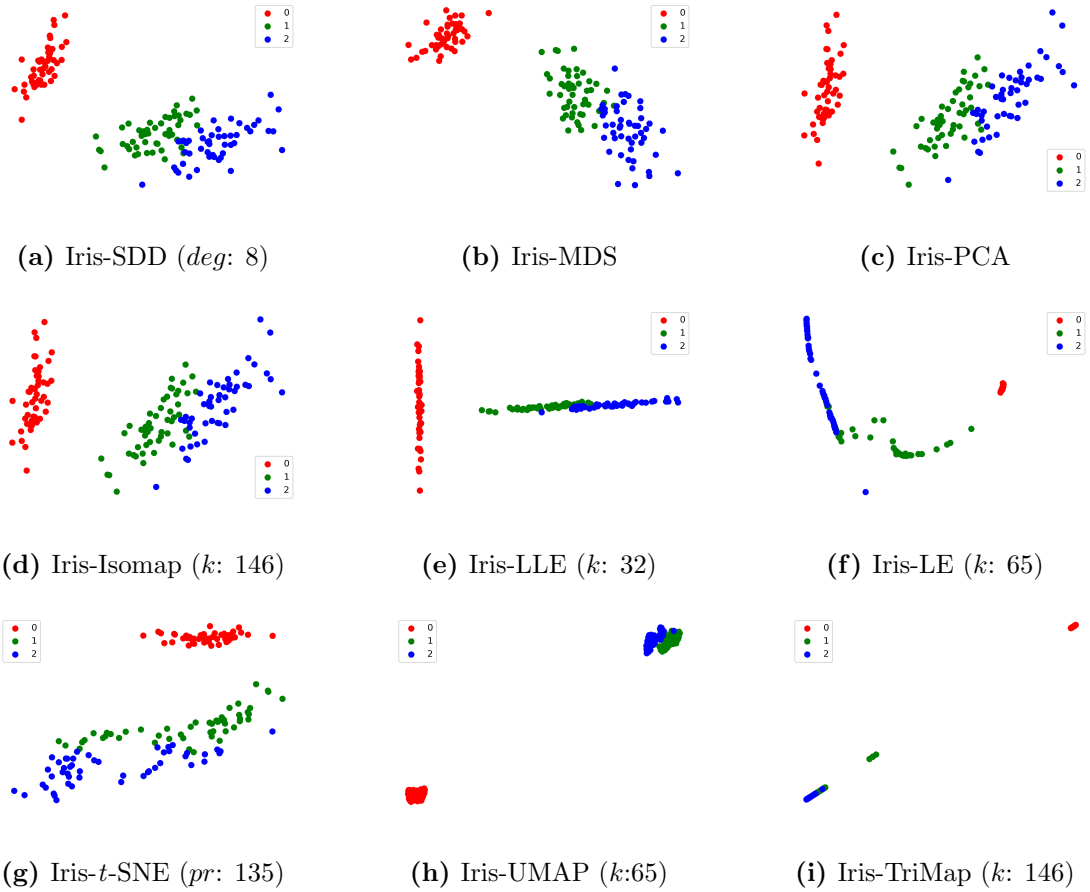
Figure 4.2: Kendall's Tau values based on the number of neighbours k (perplexity (pr) or degree of freedom (deg)).

To better investigate the local and global data structure, three other metrics, Trustworthiness, Continuity, and LCMC have been used as shown in Fig. 4.5. Trustworthiness has been used to measures the far away data samples that become close in the low dimensional space, Continuity measures close data samples that embed far away in the low dimensional space, and LCMC measures the degree of overlap between neighbours sets ain the original and embedding data. Based on Fig. 4.5, it can be seen that the best method that performs the best in terms of Trustworthiness, Continuity, and LCMC is SDD, in both local and global data structures.

Table 4.1: THE KENDALL'S TAU COEFFICIENTS FOR IRIS DATA

Parameters		
Method (Parameter)	τ	Time (*Seconds*)
SDD ($deg : 8$)	0.9673	14.23
MDS	0.9569	1.26
PCA	0.9626	0.35
Isomap ($k : 146$)	0.9627	2.38
LLE ($k : 32$)	0.6819	7
LE ($k : 65$)	0.6460	6.72
t-SNE ($pr : 135$)	0.9252	263
Umap ($k : 65$)	0.5772	554
Trimap ($k : 146$)	0.8238	992

Although, considering the computational time, SDD was more expensive than PCA, MDS, Isomap, LLE and LE; however, it outperformed t-SNE, UMAP, and TriMap.

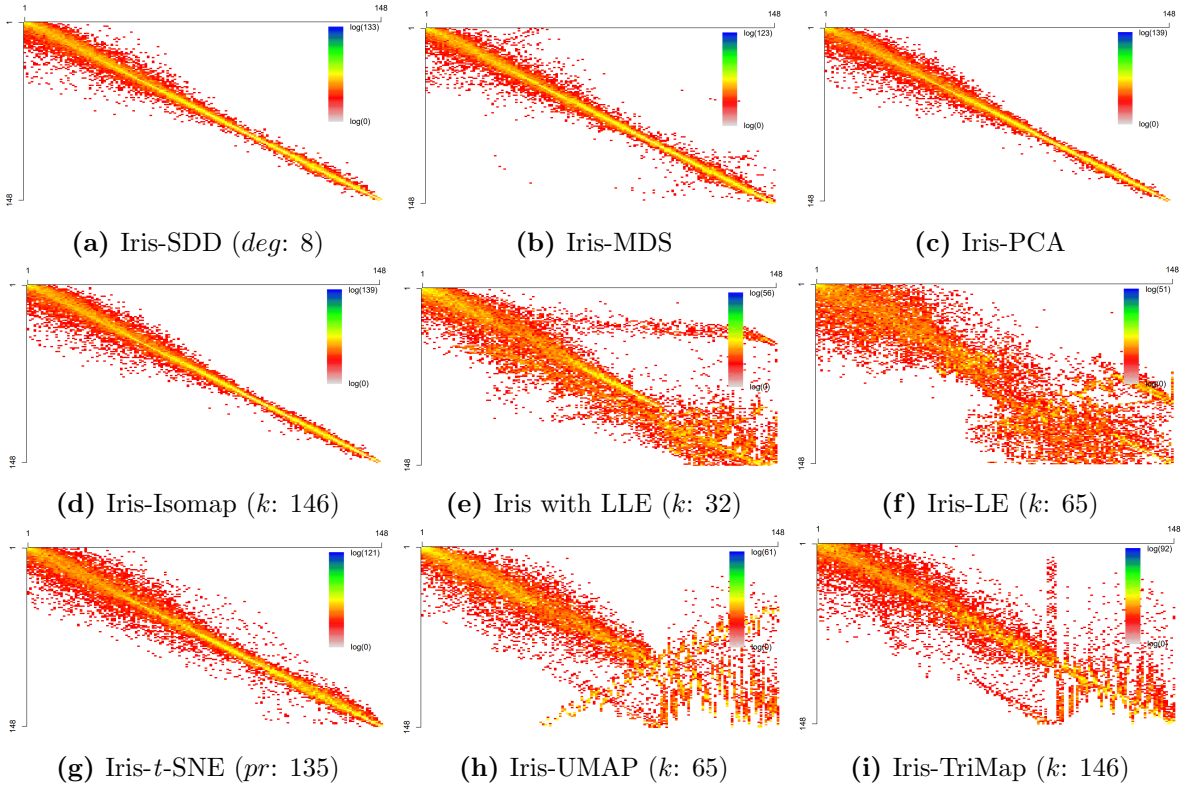


(a) Iris-SDD (*deg*: 8) **(b)** Iris-MDS **(c)** Iris-PCA

(d) Iris-Isomap (k: 146) **(e)** Iris-LLE (k: 32) **(f)** Iris-LE (k: 65)

(g) Iris-t-SNE (pr: 135) **(h)** Iris-UMAP (k:65) **(i)** Iris-TriMap (k: 146)

Figure 4.3: The visualisation of the then random samples of two-dimensional representation of the Iris (4 attributes) generated by SDD, MDS, PCA, Isomap, LLE, LE,t-SNE, UMAP and Trimap.

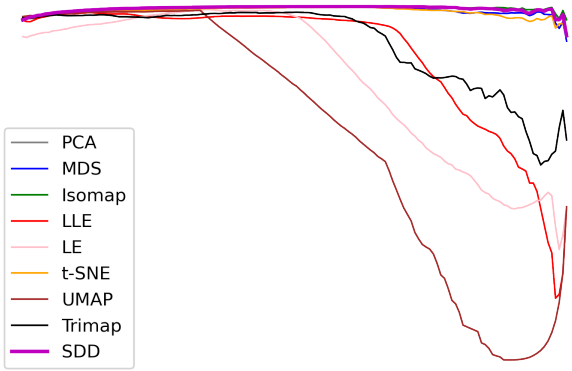


(a) Iris-SDD (*deg*: 8) **(b)** Iris-MDS **(c)** Iris-PCA

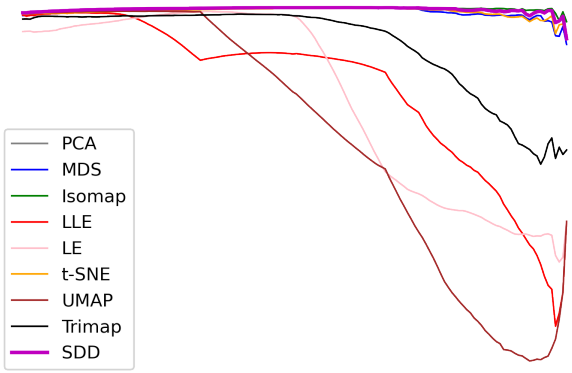
(d) Iris-Isomap (k: 146) **(e)** Iris with LLE (k: 32) **(f)** Iris-LE (k: 65)

(g) Iris-t-SNE (pr: 135) **(h)** Iris-UMAP (k: 65) **(i)** Iris-TriMap (k: 146)

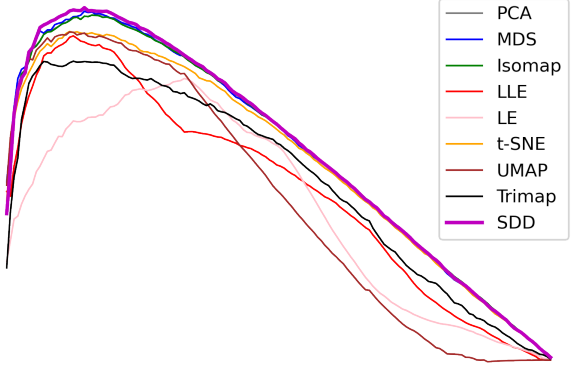
Figure 4.4: The Co-ranking matrixes of the Iris (4 attributes) by SDD, MDS, PCA, Isomap, LLE, LE, t-SNE, UMAP, and TriMap.



(a) Trustworthiness



(b) Continuity



(c) LCMC

Figure 4.5: Trustworthiness (a), Continuilty (b), and LCMC (c) for Iris data.

4.1.2 Breast cancer

The Breast Cancer dataset[4] with 30 attributes is the second dataset considered. The distance distribution of breast cancer data is shown in Fig. 4.6, where most samples have relatively short distances, in which SDD is expected to maintain the data structure better.

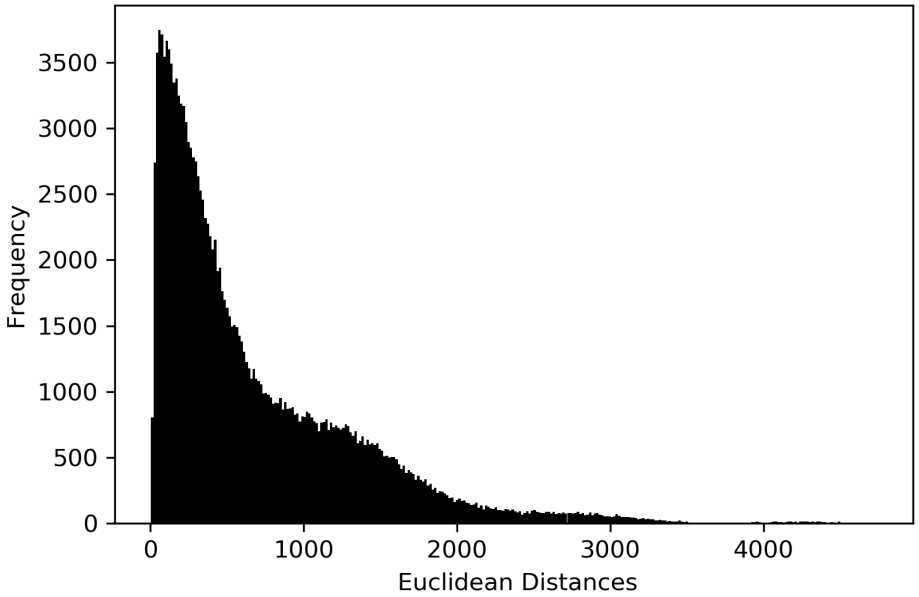


Figure 4.6: Euclidean distance distribution of Breast Cancer dataset.

Performances of nonlinear methods are related to tuning parameters such as k, pr or deg, as shown in Fig 4.7. The best two-dimensional representation in terms of structure maintenance is the one generated by SDD shown in Fig 4.8 (a) because it has the highest Kendall's Tau with 0.9981, as shown in Table 4.2. Analyzing the Co-ranking matrixes of the Breast Cancer dataset in Fig. 4.9, there is visible that SDD (deg: 10) with the Co-ranking matrix presented in Fig 4.9 (a) performed better than other methods in maintaining the short, medium and considerable distance. Also, the best performance of SDD in Breast Cancer data has been confirmed by the highest performance of three other metrics such as Trustworthiness, Continuity, and LCMC, as in Fig. 4.10. SDD is more expensive than PCA, MDS, LE; however, it was more helpful than t-SNE, LLE, and UMAP, TriMap, with higher structure maintenance and less computational time.

[4]Load breast cancer from sklearn, Python.

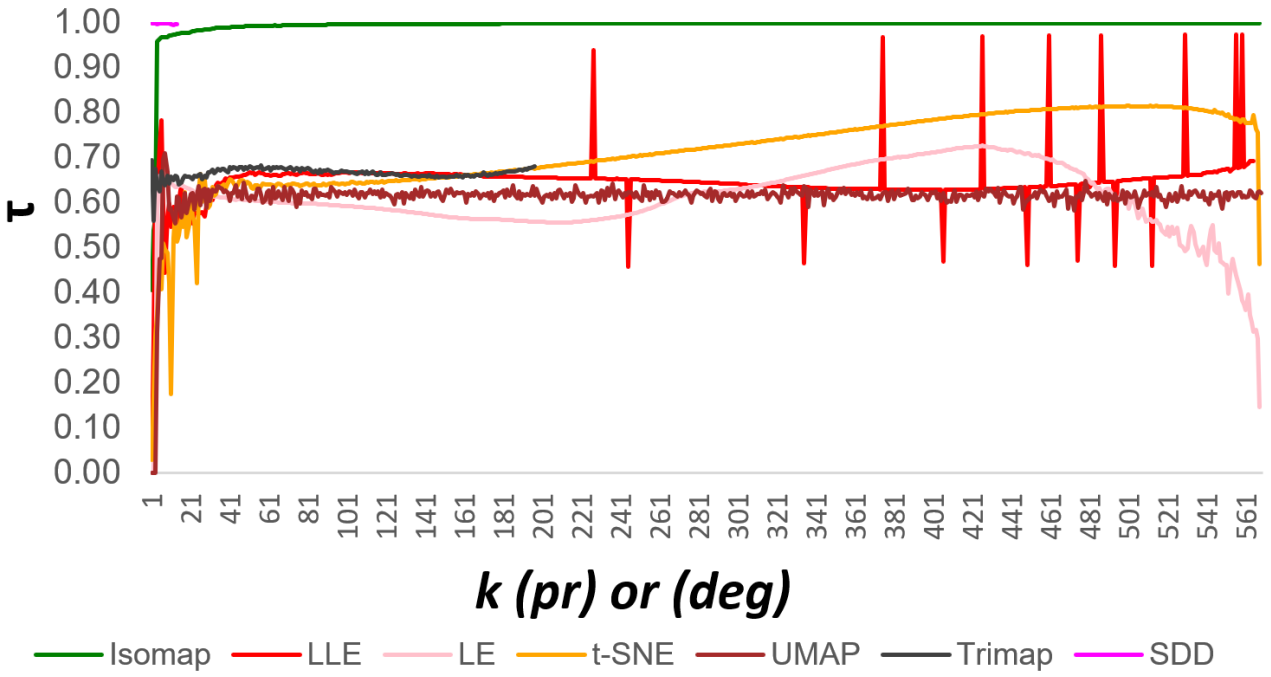
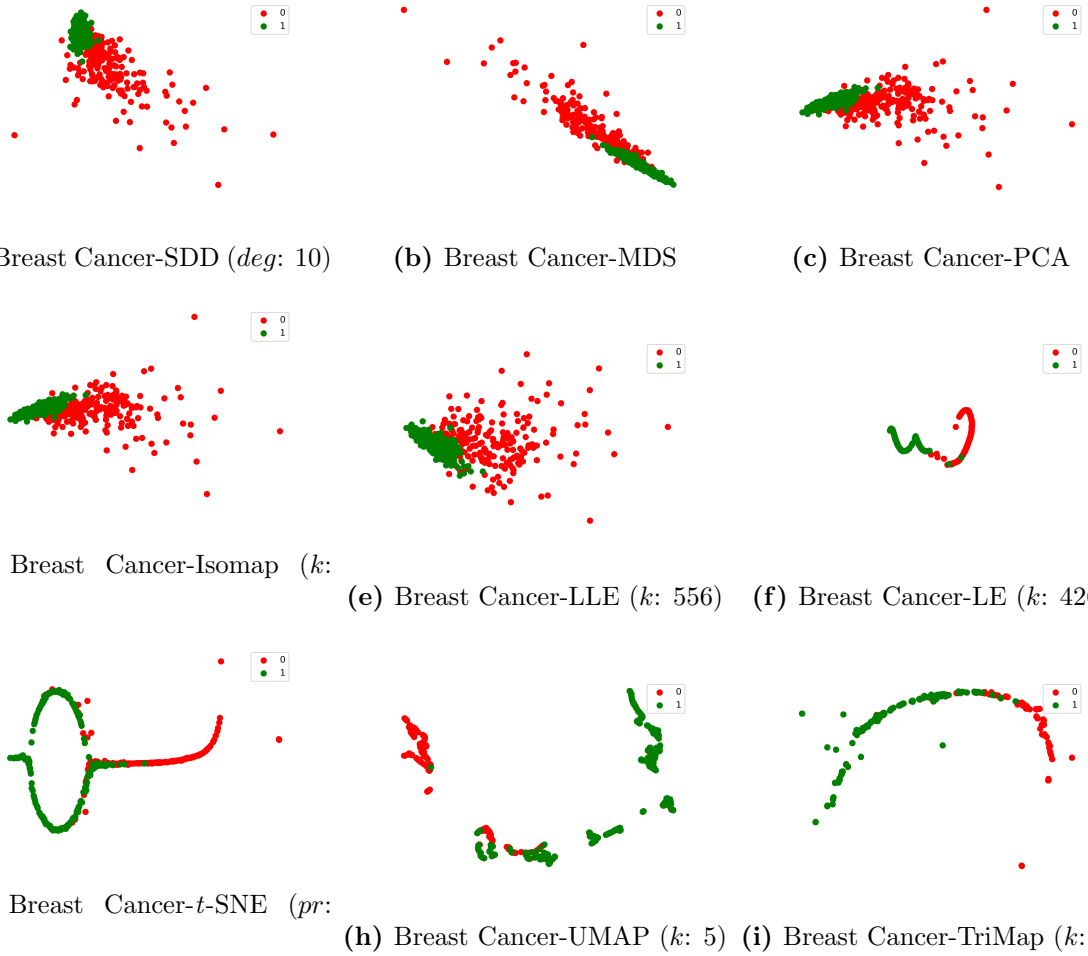


Figure 4.7: Kendall's Tau values based on the number of neighbours k (perplexity (pr) or degree of freedom (deg)).

Table 4.2: THE KENDALL'S TAU COEFFICIENTS FOR BREAST CANCER DATA

	Parameters	
Method (Parameter)	τ	Time ($Seconds$)
SDD ($deg : 10$)	0.9981	278.41
MDS	0.9970	87
PCA	0.9972	0.21
Isomap ($k : 515$)	0.9976	243
LLE ($k : 556$)	0.9728	1524
LE ($k : 426$)	0.7267	190
t-SNE ($pr : 501$)	0.8150	5952
Umap ($k : 5$)	0.709309	6105
Trimap ($k : 1$)	0.693643	3888

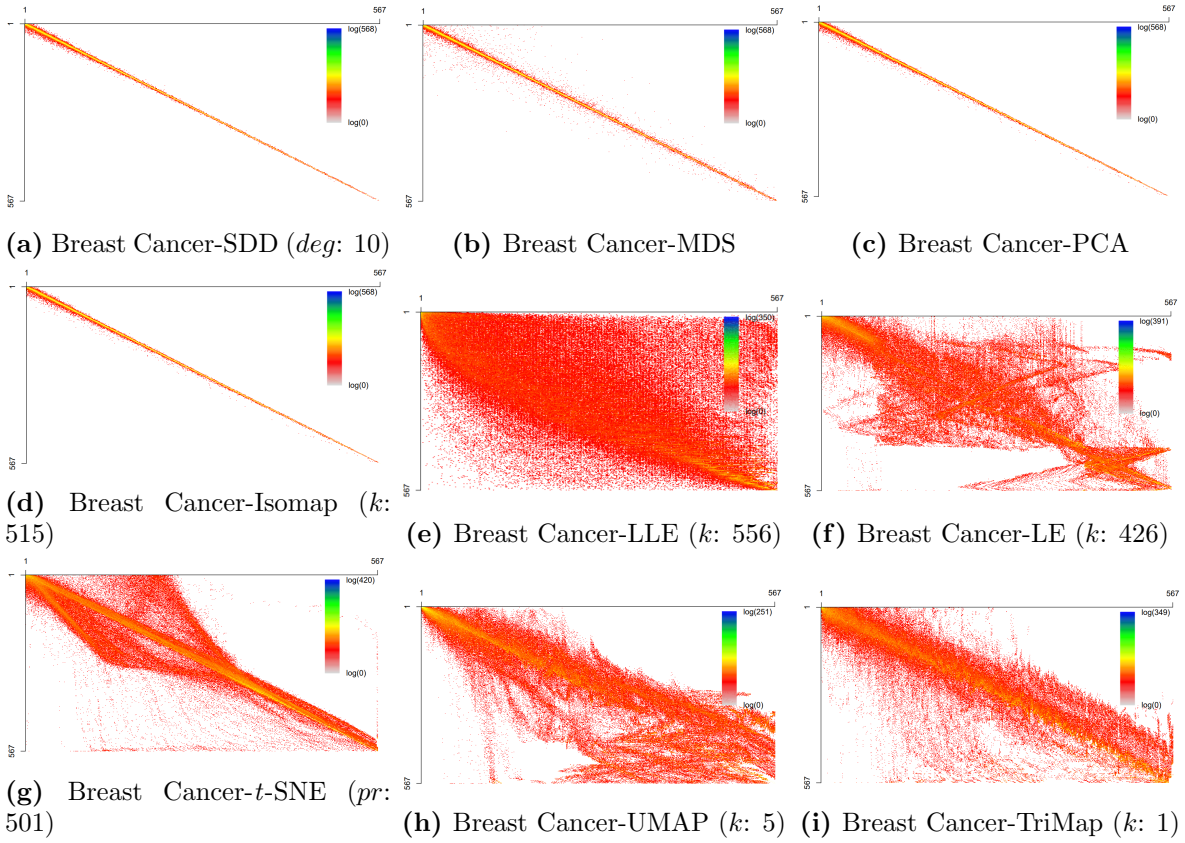


(a) Breast Cancer-SDD (*deg*: 10) **(b)** Breast Cancer-MDS **(c)** Breast Cancer-PCA

(d) Breast Cancer-Isomap (k: 515) **(e)** Breast Cancer-LLE (k: 556) **(f)** Breast Cancer-LE (k: 426)

(g) Breast Cancer-t-SNE (*pr*: 501) **(h)** Breast Cancer-UMAP (k: 5) **(i)** Breast Cancer-TriMap (k: 1)

Figure 4.8: The visualisation of two-dimensional representation of the Breast Cancer (30 attributes) generated by SDD, MDS, PCA, Isomap, LLE, LE, t-SNE, UMAP, and TriMap.

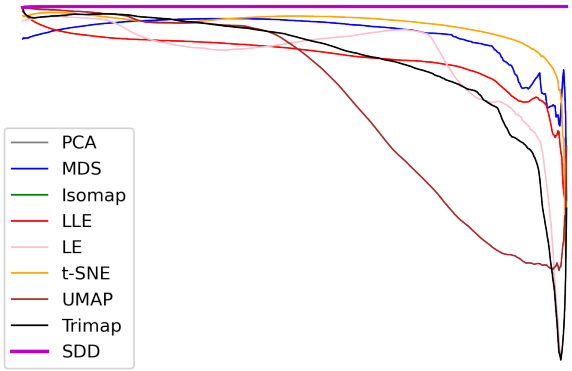


(a) Breast Cancer-SDD (*deg*: 10) **(b)** Breast Cancer-MDS **(c)** Breast Cancer-PCA

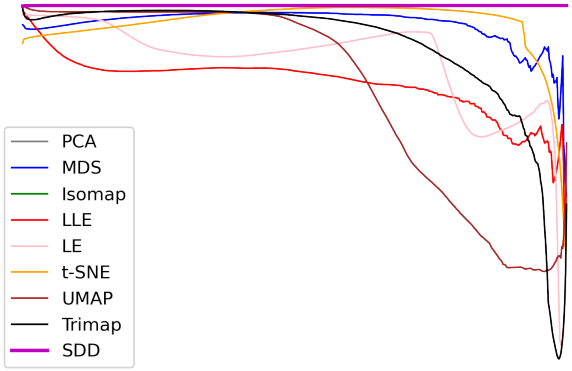
(d) Breast Cancer-Isomap (k: 515) **(e)** Breast Cancer-LLE (k: 556) **(f)** Breast Cancer-LE (k: 426)

(g) Breast Cancer-t-SNE (*pr*: 501) **(h)** Breast Cancer-UMAP (k: 5) **(i)** Breast Cancer-TriMap (k: 1)

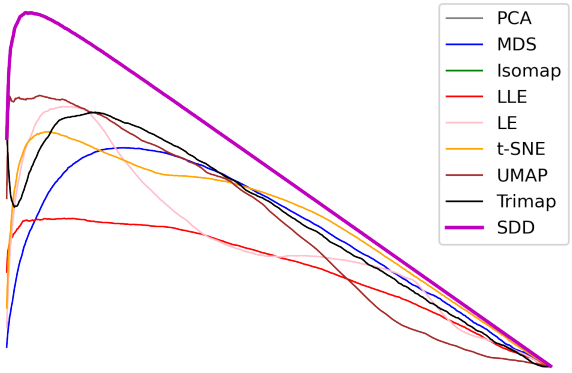
Figure 4.9: The Co-ranking matrixes of the Breast Cancer (30 attributes) generated by SDD, MDS, PCA, Isomap, LLE, LE, t-SNE, UMAP, and TriMap.



(a) Trustworthiness



(b) Continuity

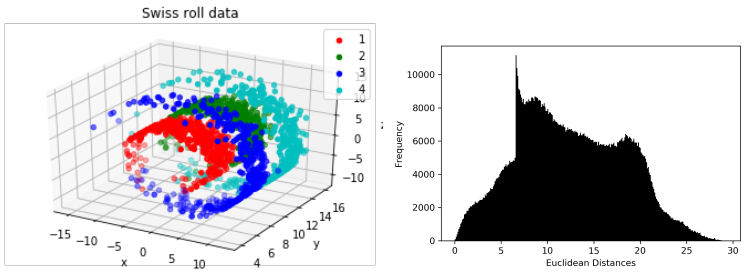


(c) LCMC

Figure 4.10: Trustworthiness (a), Continuilty (b), and LCMC (c) for Breast Cancer data.

4.1.3 Swiss Roll

Swiss Roll data with 1600 samples and three attributes is shown in Fig. 4.11(a), and its distance distribution is shown in Fig. 4.11(b), which is the third dataset considered.



(a) **(b)**

Figure 4.11: Swiss Roll data (a) and its Euclidean distance distribution (b).

SDD has achieved the highest Kendall's Tau in *deg*=1 as shown in Fig 4.12, whereas other methods such as Isomap, LLE, LE, *t*-SNE, UMAP and Trimap requires tuning up to the maximum number of data samples (1600). Also, the two-dimensional representations of SDD, PCA and Isomap (Fig 4.13) seems visually closer to the original Swiss Roll data shape than other methods. By examining the Co-ranking matrixes in Fig. 4.14, it can be seen that SDD (*deg*: 1), PCA and Isomap performed better than the other methods in preserving the data structure. SDD (*deg*: 1) produced the highest τ SDD (*deg*: 1) of 0.91461, followed by Isomap and PCA with τ of 0.9121 and 0.9115, respectively, as shown in Table 4.3. Although SDD was more expensive than two linear dimensionality reduction methods, PCA and MDS, it performed better than *t*-SNE, Isomap, LE, LLE, TriMap, and UMAP in structure maintaining and computational time, as shown in Table 4.3.

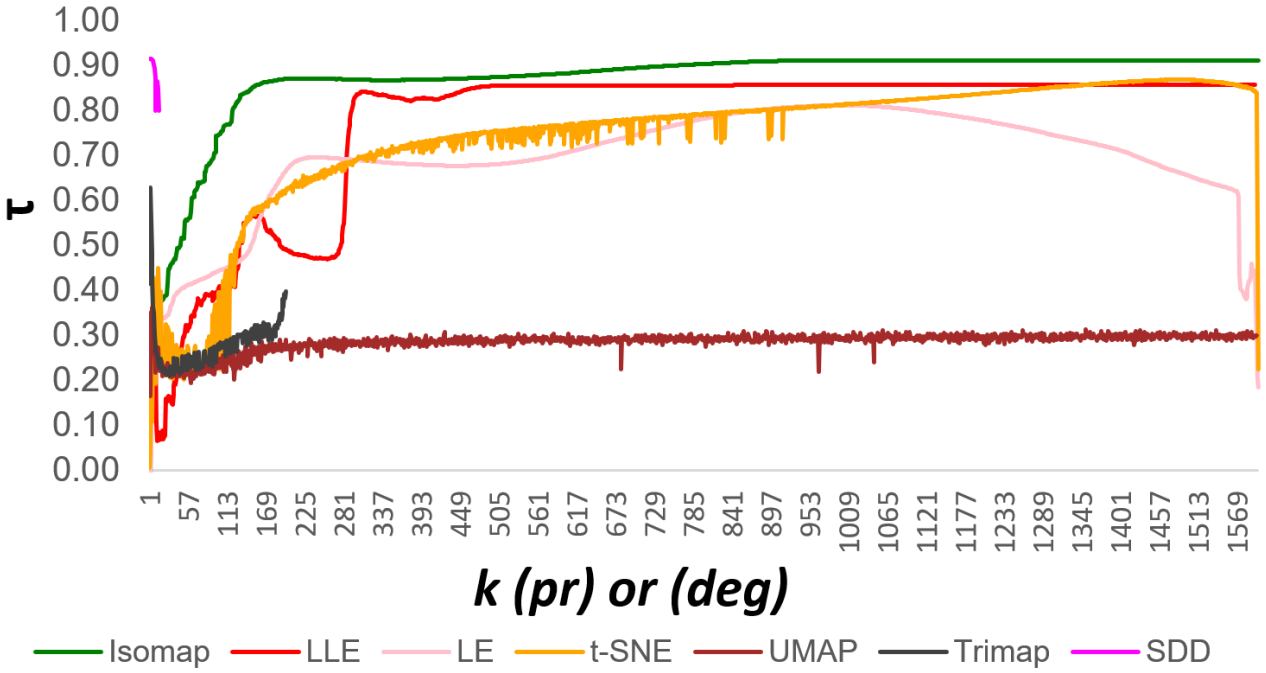
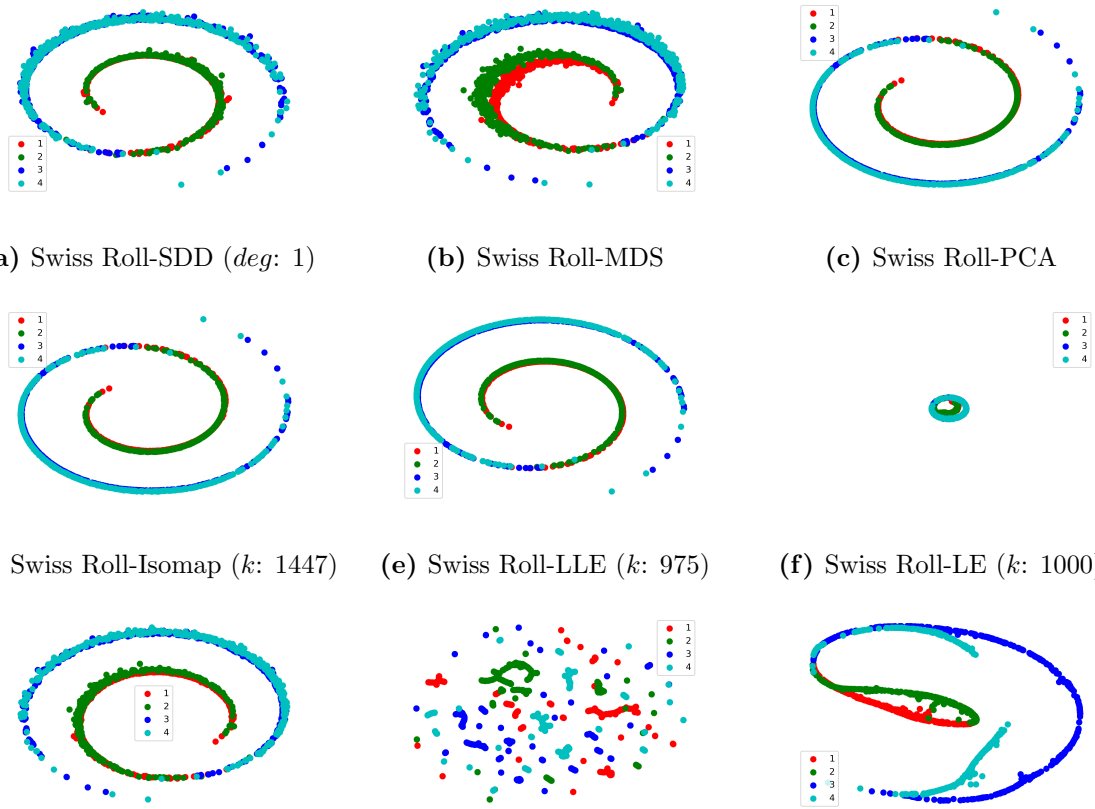


Figure 4.12: Kendall's Tau values based on the number of neighbours k (perplexity (pr) or degree of freedom (deg)).

Table 4.3: THE KENDALL'S TAU COEFFICIENTS FOR SWISS ROLL DATA

Parameters		
Method (Parameter)	τ	Time ($Seconds$)
SDD ($deg : 1$)	0.9146	1361.60
MDS	0.9041	194
PCA	0.9115	0.67
Isomap ($k : 1447$)	0.9121	15855
LLE ($k : 975$)	0.8571	118088
LE ($k : 1000$)	0.8122	4698
t-SNE ($pr : 1507$)	0.8683	75437
Umap ($k : 3$)	0.042234	58097
Trimap ($k : 12$)	0.6936	10385

SDD has shown an excellent performance of structure capturing evaluated by the other metrics such as Trustworthiness, Continuity, and LCMC, as in Fig. 4.15.

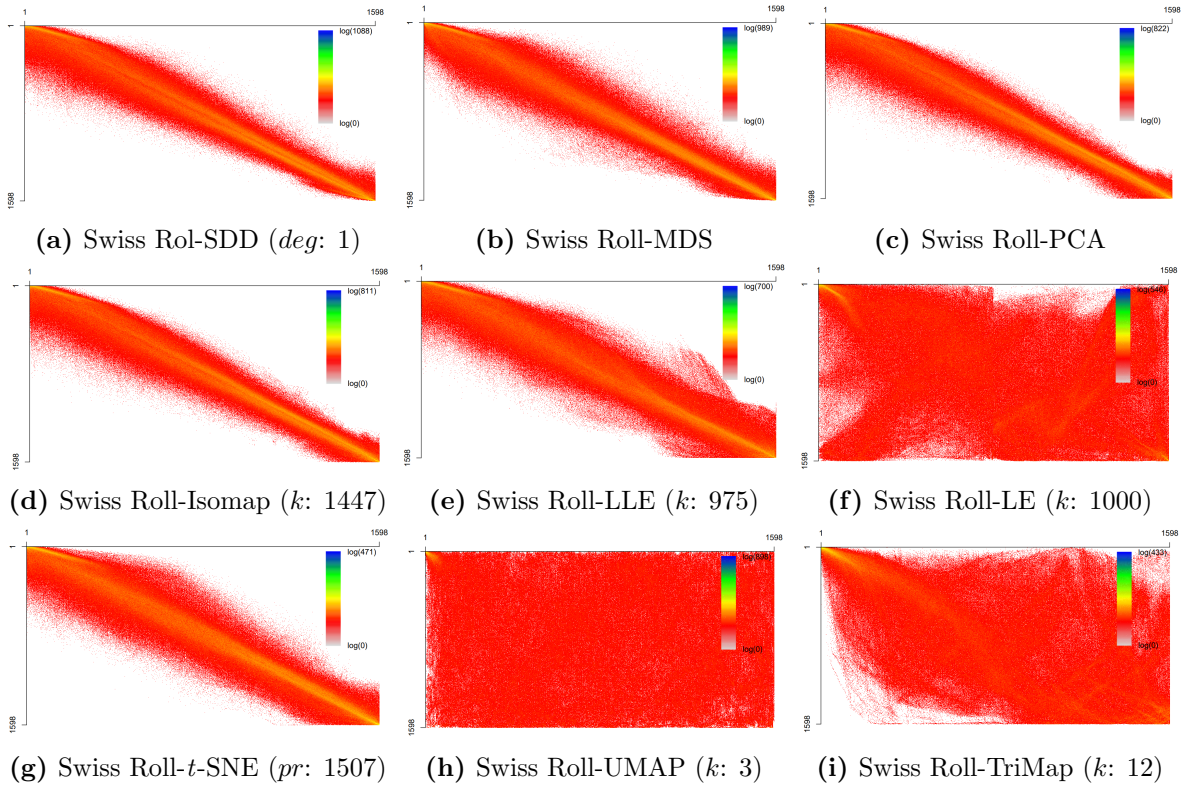


(a) Swiss Roll-SDD (*deg*: 1) **(b)** Swiss Roll-MDS **(c)** Swiss Roll-PCA

(d) Swiss Roll-Isomap (k: 1447) **(e)** Swiss Roll-LLE (k: 975) **(f)** Swiss Roll-LE (k: 1000)

(g) Swiss Roll-t-SNE (*pr*: 1507) **(h)** Swiss Roll-UMAP (k: 3) **(i)** Swiss Roll-TriMap (k: 12)

Figure 4.13: The visualisation of two-dimensional representation of the Swiss Roll (3 attributes) generated by SDD, MDS, PCA, Isomap, LLE, LE, t-SNE, UMAP, and TriMap.

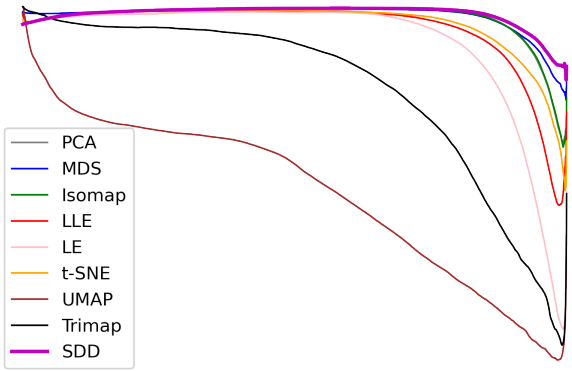


(a) Swiss Rol-SDD (*deg*: 1) **(b)** Swiss Roll-MDS **(c)** Swiss Roll-PCA

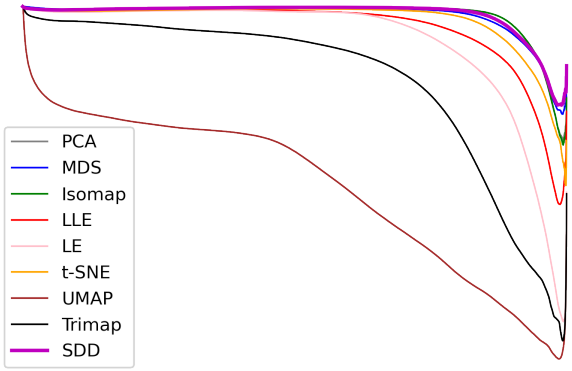
(d) Swiss Roll-Isomap (k: 1447) **(e)** Swiss Roll-LLE (k: 975) **(f)** Swiss Roll-LE (k: 1000)

(g) Swiss Roll-t-SNE (*pr*: 1507) **(h)** Swiss Roll-UMAP (k: 3) **(i)** Swiss Roll-TriMap (k: 12)

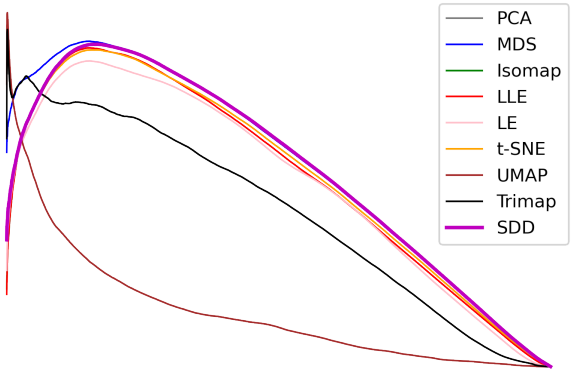
Figure 4.14: The Co-ranking matrixes of the Swiss Roll (3 attributes) generated by SDD, MDS, PCA, Isomap, LLE, LE, t-SNE, UMAP, and TriMap.



(a) Trustworthiness



(b) Continuity



(c) LCMC

Figure 4.15: Trustworthiness (a), Continuilty (b), and LCMC (c) for Swiss Roll data.

4.1.4 MNIST

MNIST with 2500 samples and 784 attributes is the fourth dataset considered, with distance distribution shown in Fig. 4.16, dominated by entries with medium, large distances. The best

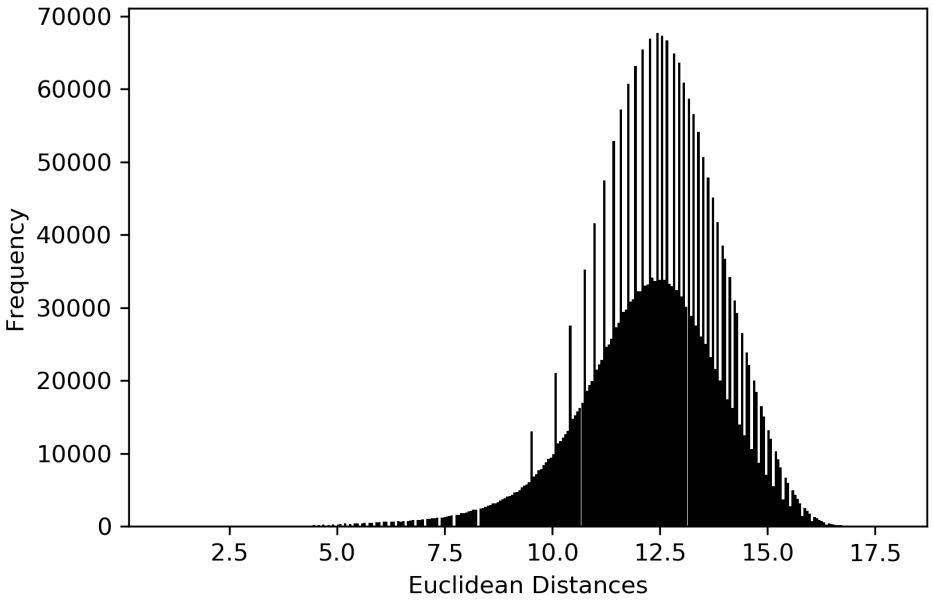


Figure 4.16: Euclidean distance distribution of MNIST data.

method in terms of structure maintenance is SDD (*deg*: 1) achieved the highest τ of 0.6065, followed by t-SNE (0.5495) and LE (0.5231), as shown in Table 4.4. As we can see, SDD hugely provided better structure preservation over the other considered methods using only $deg = 1$ as shown Fig 4.17. Furthermore, SDD was less expensive in computational time than Isomap, t-SNE, UMAP, Trimap LE, and LLE. Although PCA and MDS were faster than SDD, their performances in terms of τ were significantly low than the performance of SDD shown in Table 4.4. The two-dimensional visualisations shown in Fig 4.18 are quite similar and confirmed by the Co-ranking matrix in ig 4.19. To identify which of the methods has better captured local or global data structure are used other metrics such as Trustworthiness, Continuity, and LCMC are as in Fig. 4.20. The usage of SDD has been beneficial with the MNIST dataset in terms of both structure maintenance and computational time.

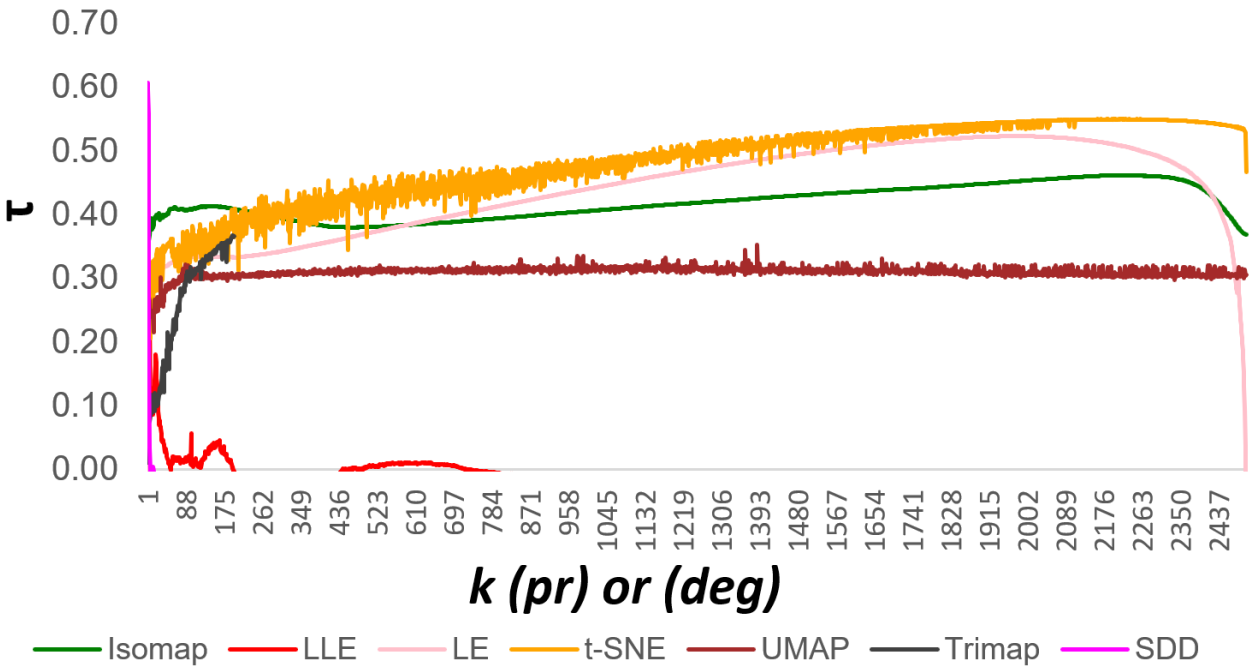
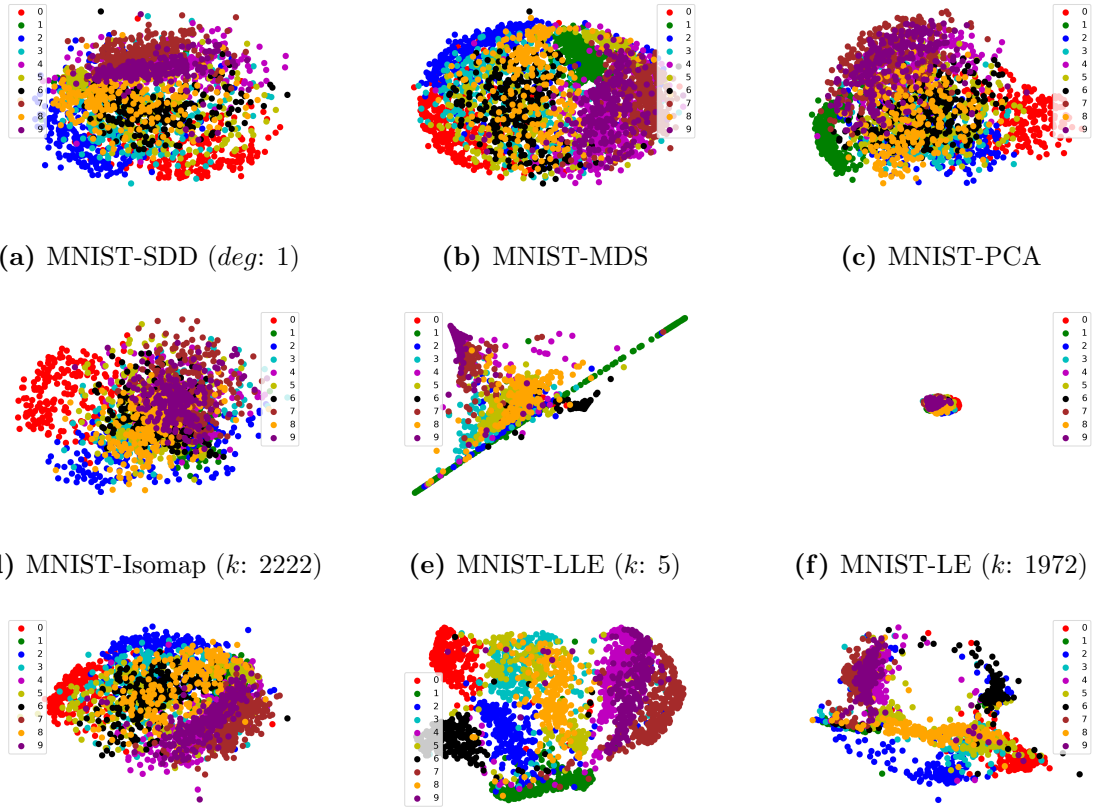


Figure 4.17: Kendall's Tau values based on the number of neighbours k (perplexity (pr) or degree of freedom (deg)).



(a) MNIST-SDD (deg: 1) **(b)** MNIST-MDS **(c)** MNIST-PCA

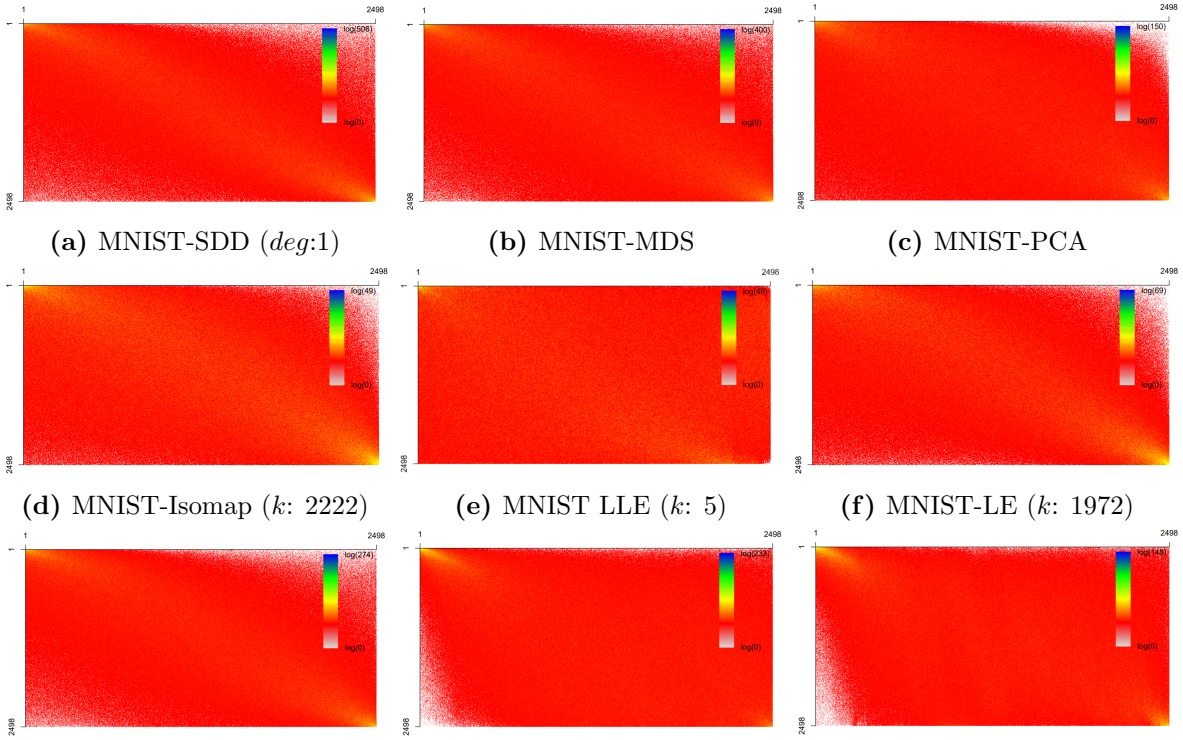
(d) MNIST-Isomap (k: 2222) **(e)** MNIST-LLE (k: 5) **(f)** MNIST-LE (k: 1972)

(g) MNIST-t-SNE (pr: 2216) **(h)** MNIST-UMAP (k: 1384) **(i)** MNIST-TriMap (k: 194)

Figure 4.18: The visualisation of two-dimensional representation of the MNIST (784 attributes) generated by SDD, MDS, PCA, Isomap, LLE, LE, t-SNE, UMAP, and TriMap.

Table 4.4: THE KENDALL'S TAU COEFFICIENTS FOR MNIST DATA

Parameters		
Method (Parameter)	τ	Time (*Seconds*)
SDD ($deg:1$)	0.6065	2454
MDS	0.5052	5387
PCA	0.3690	4
Isomap ($k:2222$)	0.4690	77957
LLE ($k:5$)	0.2433	114478
LE ($k:1972$)	0.5230	42696
t-SNE ($pr:2216$)	0.5495	295591
Umap ($k:1384$)	0.3070	236665
Trimap ($k:194$)	0.4643	17508

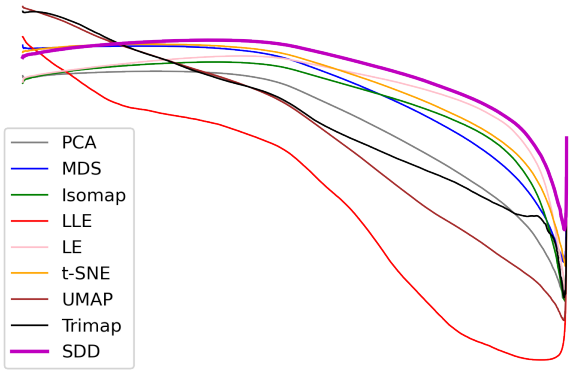


(a) MNIST-SDD (deg:1) **(b)** MNIST-MDS **(c)** MNIST-PCA

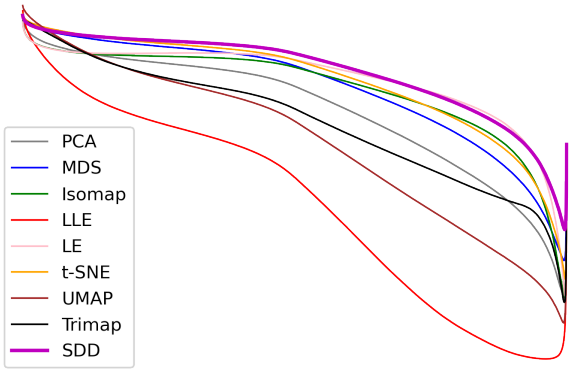
(d) MNIST-Isomap (k: 2222) **(e)** MNIST LLE (k: 5) **(f)** MNIST-LE (k: 1972)

(g) MNIST-t-SNE (pr: 2216) **(h)** MNIST-UMAP (k: 1384) **(i)** MNIST-TriMap (k: 194)

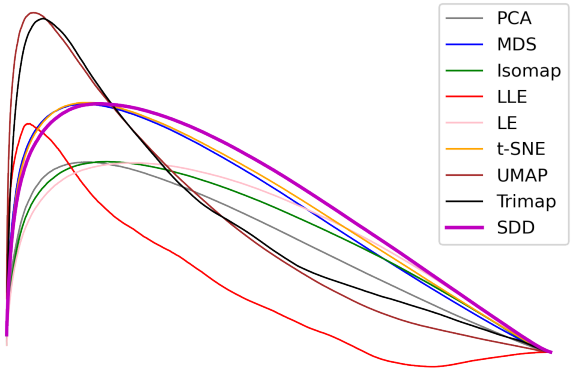
Figure 4.19: The Co-ranking matrixes of the MNIST (784 attributes) generated by SDD, MDS, PCA, Isomap, LLE, LE, t-SNE, UMAP, and TriMap.



(a) Trustworthiness



(b) Continuity



(c) LCMC

Figure 4.20: Trustworthiness (a), Continuilty (b), and LCMC (c) for MNIST data.

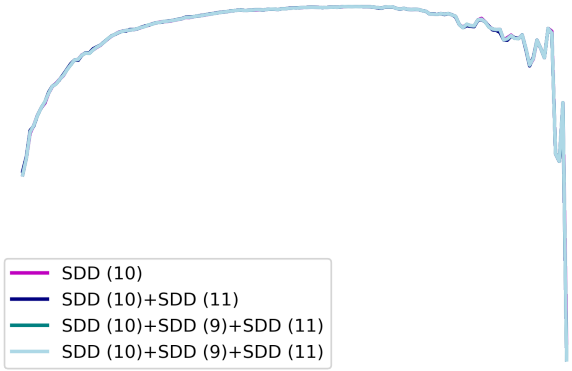
4.2 Experimental Results of MSDD

Multi SDD (MSDD) adds more distributions on top of the $best_{deg}$ that SDD employs. It is suggested that MSDD should employ degrees that are one more or less than $\{best_{deg}\}$ such as: $\{best_{deg}, best_{deg}+1\}$, $\{best_{deg}, best_{deg}-1\}$, and $\{best_{deg}, best_{deg}-1, best_{deg}+1\}$. MSDD will be implemented with the same datasets to SDD, to compare their performances in terms of structure capturing are evaluated by Kendall's Tau, Trustworthiness, Continuity and LCMC. Also, the computational time has been assessed as well.

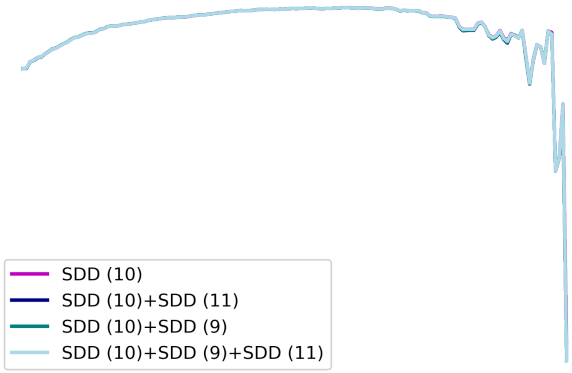
Based on Table 4.5, for Breast Cancer data, the best degree results to be degree=10, with Kendall's Tau ($\tau = 0.998121$), and adding degree-distribution with $deg = 9$, rises the maintained data structure to Kendall's Tau ($\tau = 0.998122$).

Table 4.5: THE PERFORMANCE OF METHODS (ROWS) IN DATASETS (COLUMNS) IN TERMS OF KENDALL'S TAU COEFFICIENT AND COMPUTATIONAL TIME

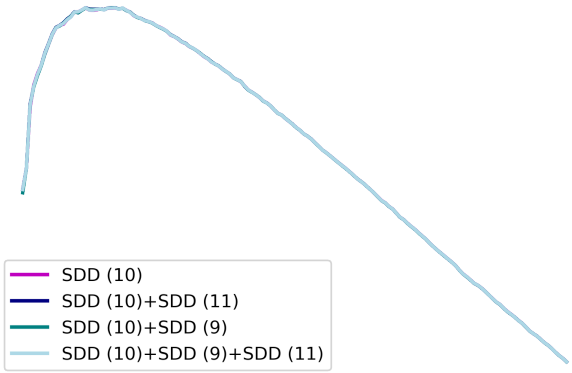
		Datasets (the number of attributes (dimensions) in the original space)			
		Iris (4)	**Breast Cancer** (30)	**Swiss Roll** (3)	**MNIST** (784)
	deg_{best}	8	10	1	1
	τ	0.9673284	0.998121	0.914619	0.606540
	t	9.49	278.41	562.57	8194.18
MSDD	deg	8 and 9	10 and 11	1 and 2	1 and 2
	τ	0.967309	0.998120	0.914707	
	t	4.06	34.22	118.93	4177.75
	deg	7 and 8	9 and 10	0 and 1	0 and 1
	τ	0.967316	0.998122	0.914707	0.600533
	t	3.04	25.82	90	3080.225
	deg	7,8 and 9	9,10 and 11	0, 1 and 2	0, 1 and 2
	τ	0.967334	0.998121	0.914709	0.598738
	t	7.61	44	90	8086



(a) Trustworthiness

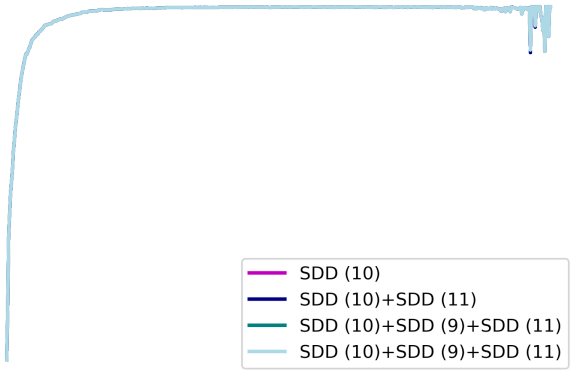


(b) Continuity

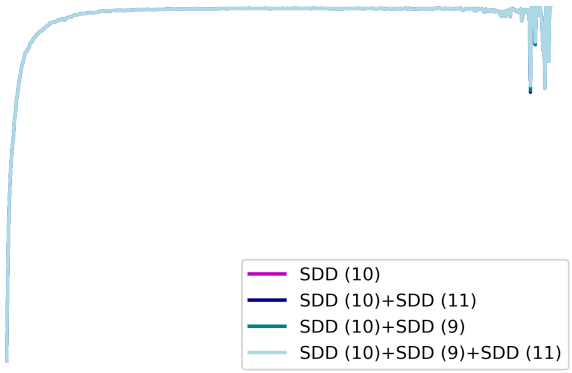


(c) LCMC

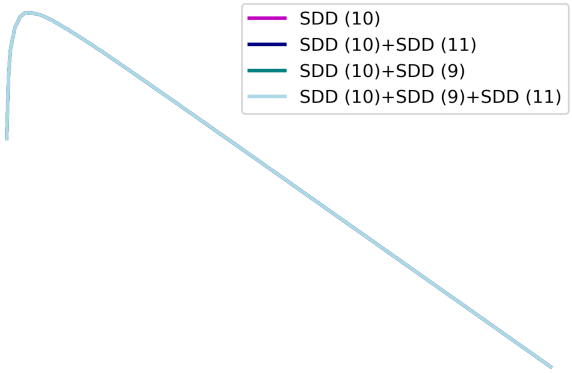
Figure 4.21: Trustworthiness (a), Continuilty (b), and LCMC (c) for Iris data.



(a) Trustworthiness

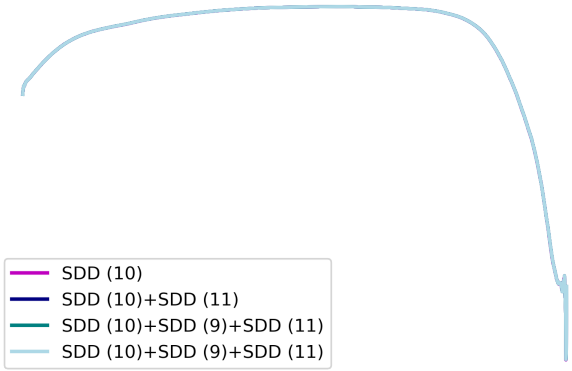


(b) Continuity

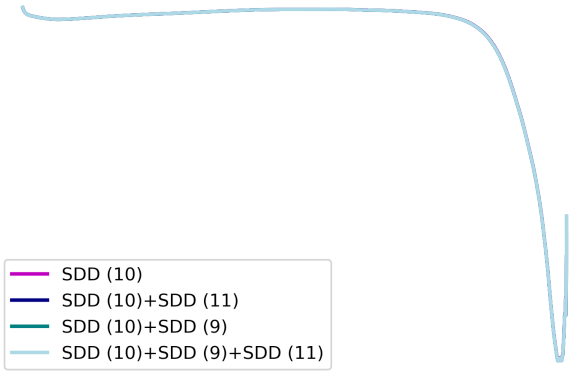


(c) LCMC

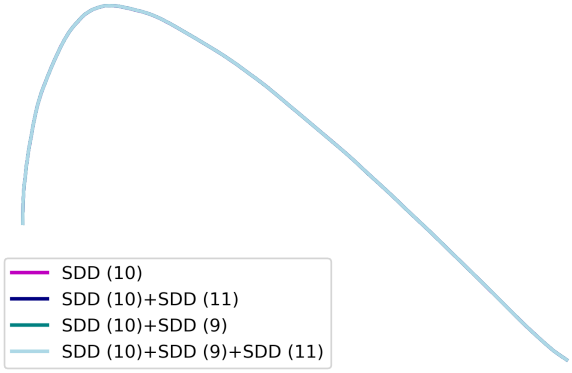
Figure 4.22: Trustworthiness (a), Continuilty (b), and LCMC (c) for Breast Cancer data.



(a) Trustworthiness

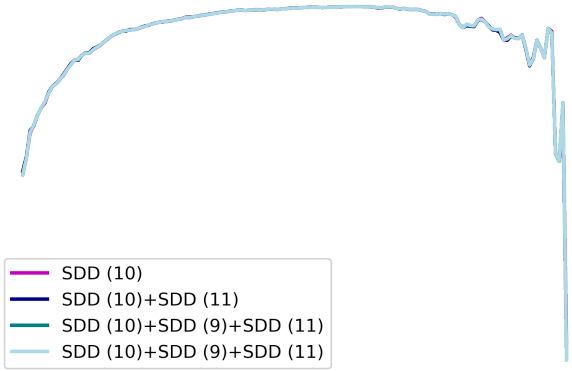


(b) Continuity

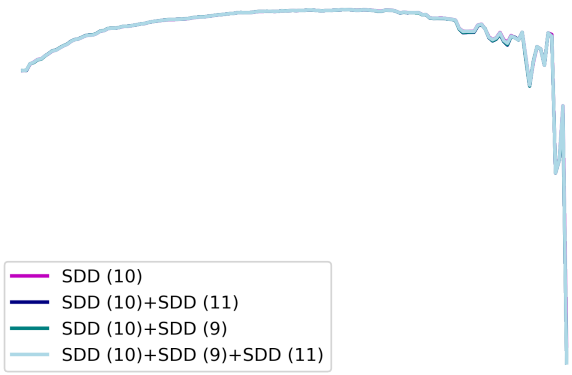


(c) LCMC

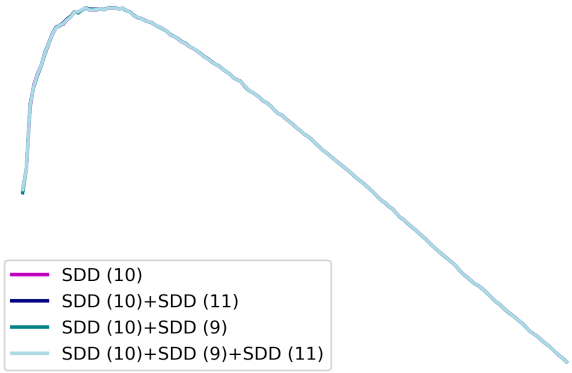
Figure 4.23: Trustworthiness (a), Continuilty (b), and LCMC (c) for Swiss Roll data.



(a) Trustworthiness



(b) Continuity



(c) LCMC

Figure 4.24: Trustworthiness (a), Continuilty (b), and LCMC (c) for MNIST data.

Adding more distribution on top of the best degree-distribution in tthe most of datasets does not improve the maintained data structure, as demonstrated from evaluated parameters such as Kendall's Tau in Table 4.5 and Trustworthiness, Continuity and LCMC, shown in Figs. 4.21, 4.22, 4.23, and 4.24.

In summary, adding more distributions destroys the maintained data structure, as shown in Table 4.5. In addition, employing more than one degree-distributions is more expensive than using one-degree distribution. As a result, MSDD is more costly than SDD, and in addition, the computational time increases if the number of degree-distributions employed increases. In conclusion, based on our experimental results, MSDD does not improve the structure-capturing in a significant amount; on the other hand, it needs more computational resources.

4.3 Experimental Results of Parameter-Free SDD

Parameter-free SDD is an innovative method that takes the highes performance of SDD but saves computational time significantly. As it is mentioned in Section 3.3, parameter-free SDD does not require tuning any parameter, and it uses only $deg = 1$, which results to be the $best_{deg}$ due to rescaling the pairwise distances in the range [0, 2].

The performance of parameter-free SDD has been evaluated using Kendall's Tau, Trustworthiness and Continuity and is compared with three different degrees of SDD such as 1, 15 and $best_{deg}$. The experiments are all done on Python with the same number of iterations and optimisation parameters. The dataset considered are Iris, Breast Cancer, Swiss and MNIST.

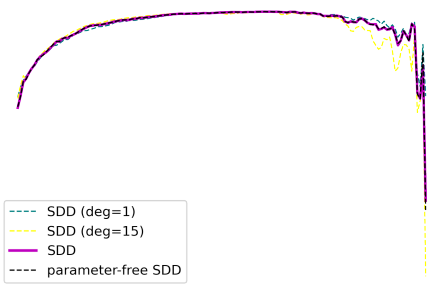
Based on experimental results, parametric-free SDD appears provenly appropriate to capture local and global data structures due to the high sensitivity of degree-distribution with $deg = 1$ has in the short and large distances into the intervals 0 and 2. As shown in Fig. 3.8, parameter-free SDD (SDD with $deg = 1$ in [0, 2]) captures slightly the same the local data structure (short distances) compared with SDD ($deg = best$) in [0, 1]. However, parameter-free SDD can capture global data structure better than SDD ($deg = best$), as demonstrated in Figs. 4.25 and 4.26.

In addition, the performance of parameter-free SDD has been evaluated using Kendall's Tau, which, as is demonstrated in Table 4.6, is very similar to SDD ($deg = best$). However, in terms of computational time, parameter-free SDD is significantly less expensive than SDD, as shown in Table 4.6. Parameter-free SDD takes 0.41, 7.79, 36.93, 183.94 seconds to generates low dimensional data of Iris, Breast Cancer, Swiss Rolls and MNIST data instead of 9.49, 111.33, 552.96 and 2452.12 seconds that SDD takes.

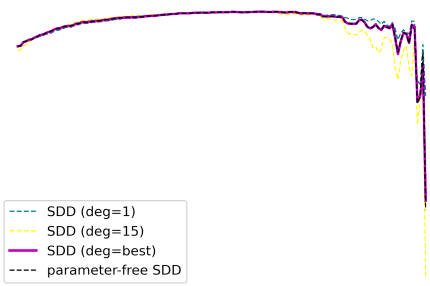
Table 4.6: THE PERFORMANCE OF SDD AND PARAMETER-FREE SDD IN TERMS OF KENDALL'S TAU COEFFICIENT AND COMPUTATIONAL TIME

		Datasets			
		Iris	**Breast Cancer**	**Swiss Rolls**	**MNIST**
SDD	deg	8	10	1	1
	τ	0.967328	0.998118	0.914711	0.606525
	$time(seconds)$	9.49	111.33	552.96	2454.12
Parameter-free SDD	τ	0.967339	0.998086	0.914578	0.607947
	$time(seconds)$	0.41	7.79	36.93	183.94

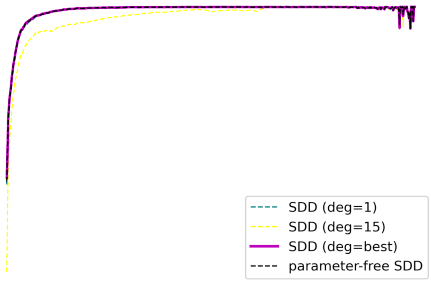
In summary, parameter-free SDD can capture the structure of the data very well due to 1) the long tail of Student-t distribution in capturing the global data structure and 2) to the advantages of rescaling the pairwise distances in the interval [0, 2], which improves capturing the local data structure. It might happen that because the sensitivity for large distances is small when distance is increased into [0, 2], they may be negligible from the cost function, making the global structure not as good as Student-t ($deg = 1$). However, the global structure captured by parameter-free SDD is better than the global data structure captured the best degree SDD in pairwise distance ranges in [0, 1].



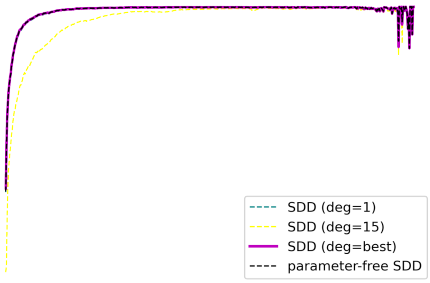
(a) Trustworthiness (Iris)



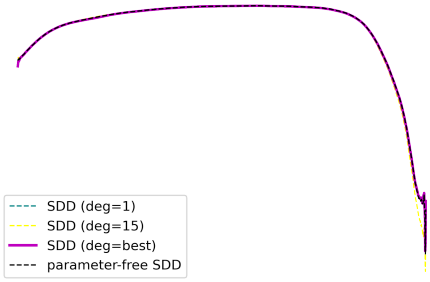
(b) Continuity (Iris)



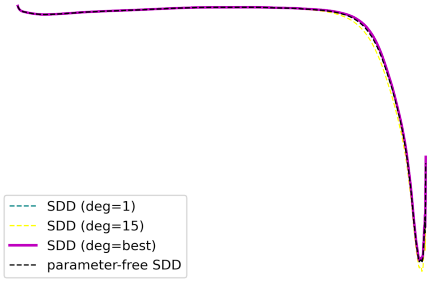
(c) Trustworthiness (Breast Cancer)



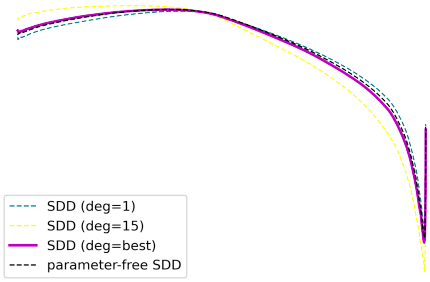
(d) Continuity (Breast Cancer)



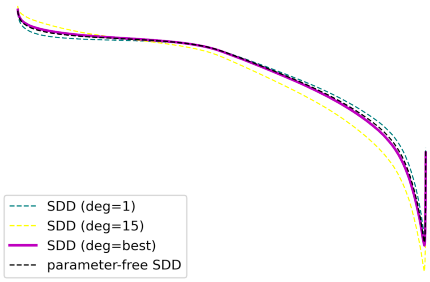
(e) Trustworthiness (Swiss Roll)



(f) Continuity (Swiss Roll)



(g) Trustworthiness (MNIST)

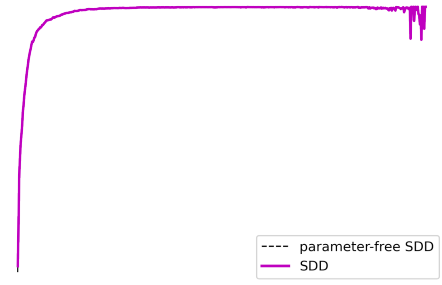


(h) Continuity (MNIST)

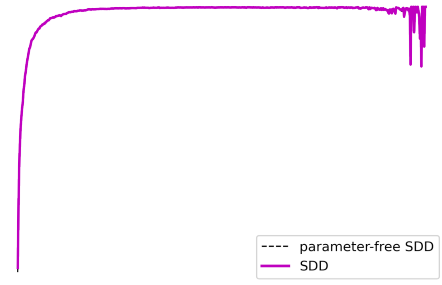
Figure 4.25: Trustworthiness and Continuity for SDD with degrees 1, degree (best) and 15 for rescaled distances in range [0, 1], and parameter-free SDD.

(a) Trustworthiness (Iris)

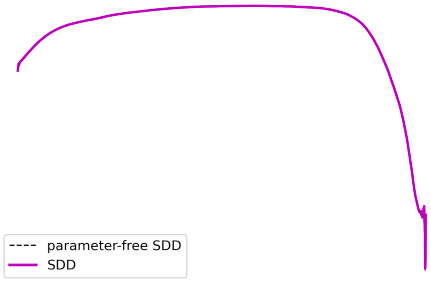
(b) Continuity (Iris)



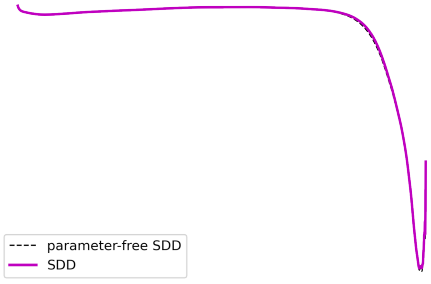
(c) Trustworthiness (Breast Cancer)



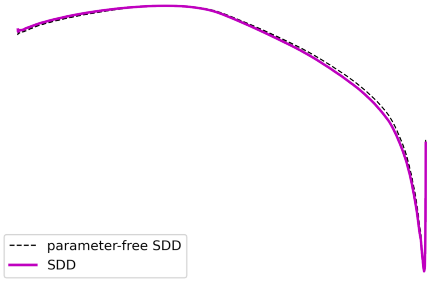
(d) Continuity (Breast Cancer)



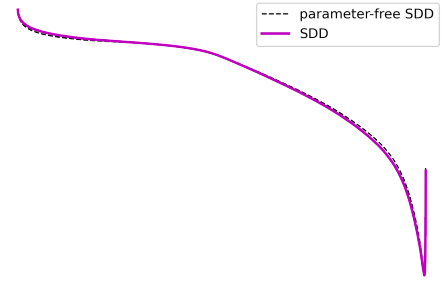
(e) Trustworthiness (Swiss Roll)



(f) Continuity (Swiss Roll)



(g) Trustworthiness (MNIST)



(h) Continuity (MNIST)

Figure 4.26: Trustworthiness and Continuity for SDD with best degrees, and parameter-free SDD.

4.4 Experimental Results of Parametric SDD

Despite the outstanding performance of parameter-free SDD in structure maintaining and computational time, it still remains an implicit method, which cannot reduce the data dimensionality of new considered data samples. In such as case, proposing an explicit SDD or (parametric SDD) is crucial in terms of saving computational time and resources. Also, to compare the performance of parametric SDD and other methods, are considered PCA, MDS, Isomap, LE, t-SNE, UMAP are implemented in Python with the same number of iterations. For all of the methods, the procedure is as follows:

1. Reduce data dimensionality of X_e,
2. Train ANN based on X_e and its reduced data from each method, and
3. Reduces the dimensionality of the test data based on each trained ANN.

The performance of Isomap, t-SNE and UMAP depend on some parameters that have been tunned to check their performance estimated by Kendall's Tau correlation coefficient (τ). Dataset considered in this section are non-temporal data such as Synthetic data (MNIST), Medical data (SEER Breast Cancer), Customer data (Churn data), Image processing data (AVletters (LIPS)). PCA is a parametric method, and there exists a parametric t-SNE; however, for comparison reasons, the same Neural Network architecture has been applied to all methods.

The *MNIST* data with 60,000 gray images from 0 to 9 with 28×28 pixels will be flattened into 784 dimensional record.[5]

The *SEER Breast Cancer* data contains a totally of 291,760 incidences registered in the US from 1974 to 2017. The original data sets need to be pre-processed and transformed to a target data set for analysis. The most crucial task in the data pre-processing process is identifying any data quality issues and adopting appropriate strategies to address them accordingly. The

[5]MNIST data from Keras.

data pre-processing process was very time-consuming, and it has eventually led to a resultant target data set with 260,000 incidences and 961 variables. The variable survival that represents if a patient survived has been considered the target variable since this analysis aims to identify crucial factors that potentially affect the survival of a breast cancer patient.

The *Churn* data contains 9786 customers that are described by 2 variables, a customer is churn or no.

The *AVletters* database (LIPS Reading data) consists of three repetitions by each of 10 talkers, five male (two with moustaches) and five female, of the isolated letters A-Z, a total of 780 utterances. Each talker was requested to begin and end each letter utterance with their mouth in the closed position. No head restraint was used, but talkers were provided with a close-up view of their mouths and asked not to move out of the frame. The full face images were further cropped to a region of 80×60 pixels after manually locating the centre of the mouth in the middle frame of each utterance.

Table 4.7: THE PERFORMANCE OF METHODS (ROWS) IN DATASETS (COLUMNS) IN TERMS OF KENDALL'S TAU COEFFICIENT

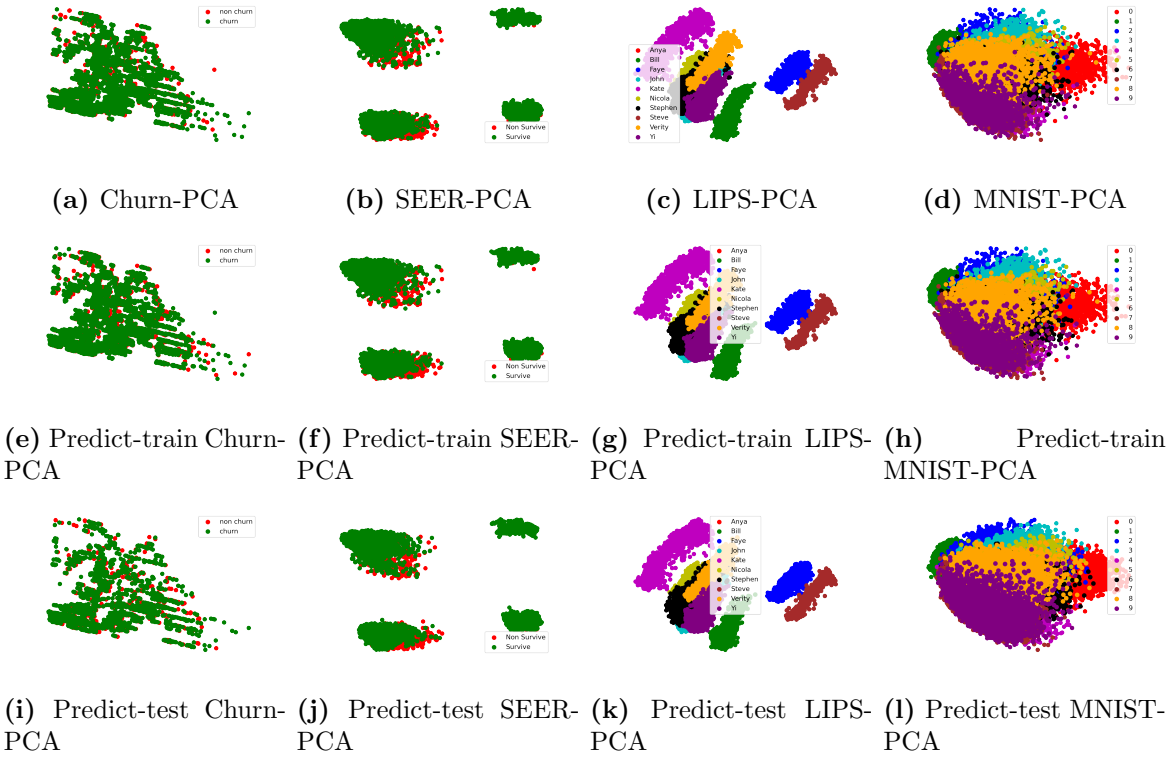
DR algorithm	**Churn**	**SEER**	**LIPS**	**MNIST**
PCA	0.9691	0.5154	0.7391	0.3533
PCA (predict train)	0.9688	0.5152	0.7401	0.3489
PCA (predict test)	0.9693	0.5194	0.7417	0.3401
t-SNE	0.7466	0.2729	0.3824	0.2395
t-SNE (predict train)	0.8509	0.2764	0.3825	0.2413
t-SNE (predict test)	0.8562	0.2796	0.3799	0.2432
Isomap	0.9043	0.4778	0.7399	0.4021
Isomap (predict train)	0.9062	0.4783	0.7399	0.4018
Isomap (predict test)	0.91081	0.4845	0.7416	0.3984
SDD	0.9723	0.7279	0.7948	0.6199
SDD (predict train)	0.9711	0.7258	0.7838	0.6130
SDD (predict test)	0.9715	0.7247	0.7849	0.6059

Note that the ratio of training/testing samples in % is different in different datasets as in Table 4. Training/testing samples (%) is 70%/30%, 50%/50%, 50%/50%, 25%/75% in Churn, SEER

Breast Cancer, LIPS and MNIST datasets, respectively.

Table 4.8: DATASETS (ROWS) AND TRAINING, TESTING SAMPLES AND DIMENSIONALITY

Datasets	Training Samples (%)	Testing Samples (%)	Dimensionality
Churn	6850 (70 %)	2936 (30%)	23
SEER Breast Cancer	15000 (50 %)	15000 (50 %)	960
LIPS	9280 (50 %)	9280 (50 %)	4800
MNIST	15000 (25 %)	45000 (75 %)	784

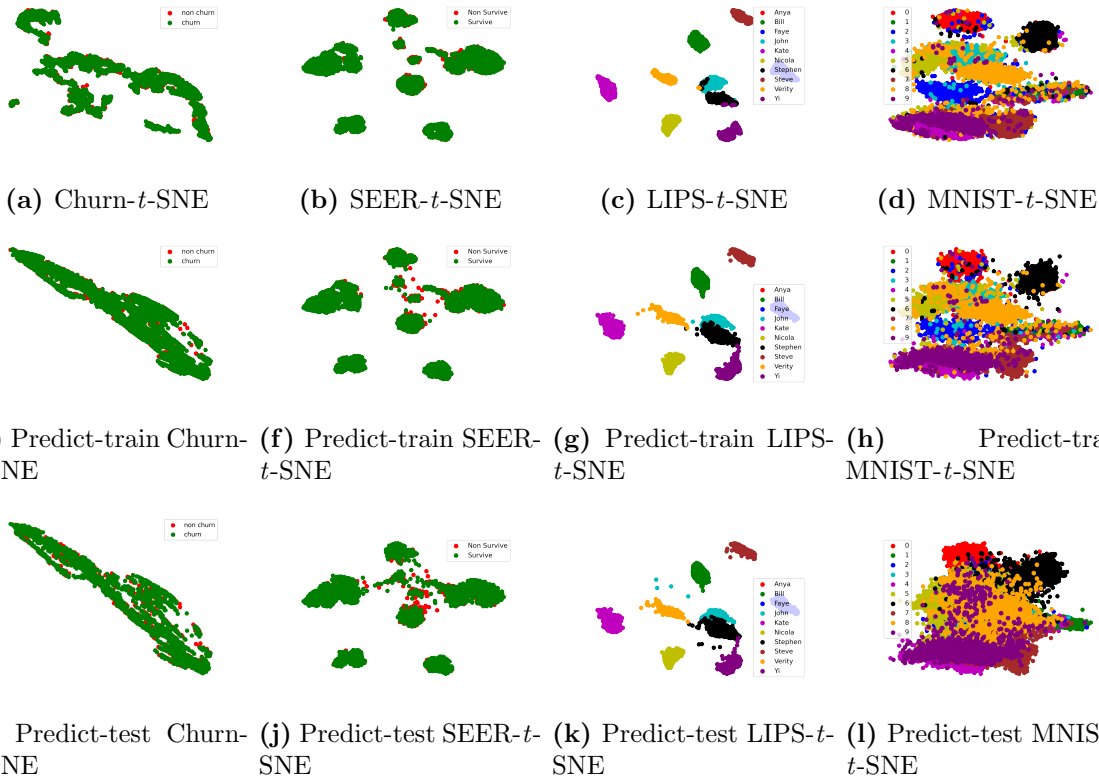


(a) Churn-PCA **(b)** SEER-PCA **(c)** LIPS-PCA **(d)** MNIST-PCA

(e) Predict-train Churn-PCA **(f)** Predict-train SEER-PCA **(g)** Predict-train LIPS-PCA **(h)** Predict-train MNIST-PCA

(i) Predict-test Churn-PCA **(j)** Predict-test SEER-PCA **(k)** Predict-test LIPS-PCA **(l)** Predict-test MNIST-PCA

Figure 4.27: The visualisation of the two-dimensional representation of the Churn (23 attributes), SEER Breast Cancer (960 attributes), Lips (4800 attributes), MNIST (784 attributes) generated by PCA.

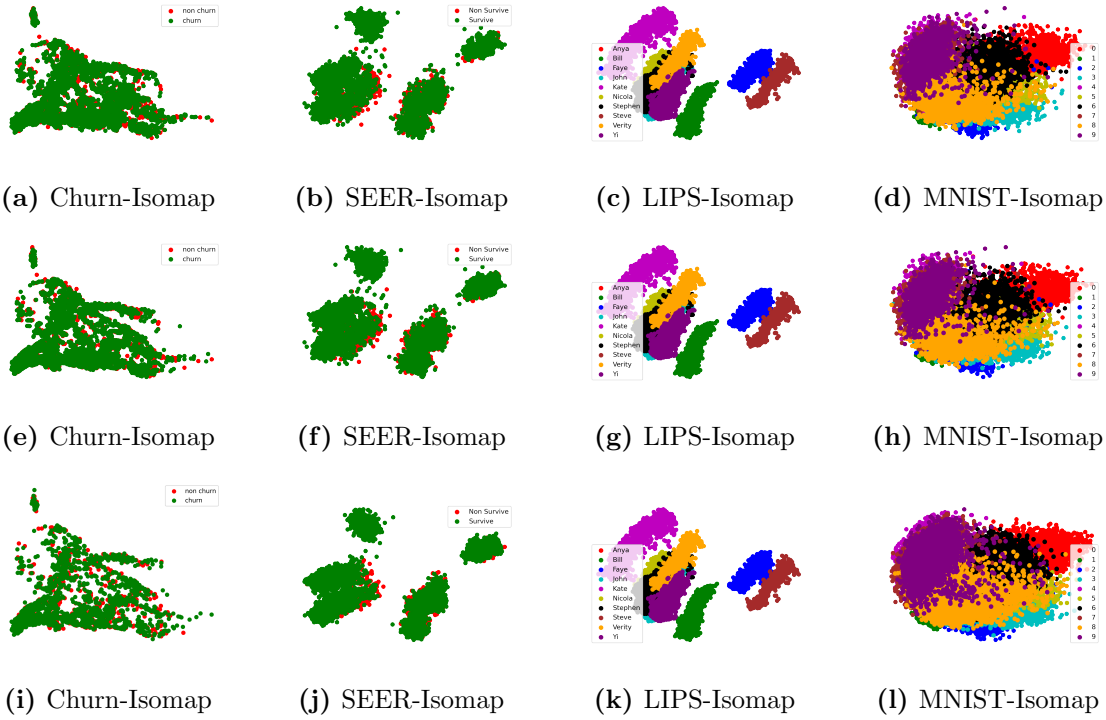


(a) Churn-t-SNE **(b)** SEER-t-SNE **(c)** LIPS-t-SNE **(d)** MNIST-t-SNE

(e) Predict-train Churn-t-SNE **(f)** Predict-train SEER-t-SNE **(g)** Predict-train LIPS-t-SNE **(h)** Predict-train MNIST-t-SNE

(i) Predict-test Churn-t-SNE **(j)** Predict-test SEER-t-SNE **(k)** Predict-test LIPS-t-SNE **(l)** Predict-test MNIST-t-SNE

Figure 4.28: The visualisation of the two-dimensional representation of the Churn (23 attributes), SEER Breast Cancer (960 attributes), Lips (4800 attributes), MNIST (784 attributes) generated by t-SNE.

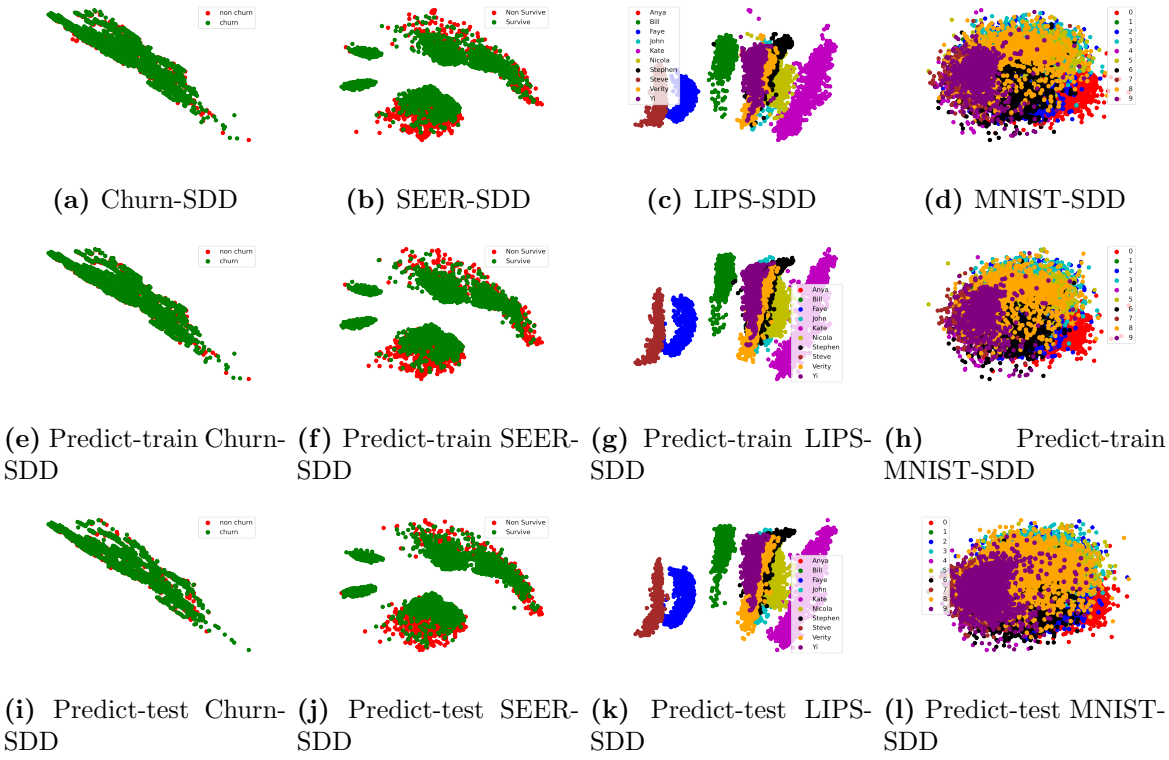


(a) Churn-Isomap **(b)** SEER-Isomap **(c)** LIPS-Isomap **(d)** MNIST-Isomap

(e) Churn-Isomap **(f)** SEER-Isomap **(g)** LIPS-Isomap **(h)** MNIST-Isomap

(i) Churn-Isomap **(j)** SEER-Isomap **(k)** LIPS-Isomap **(l)** MNIST-Isomap

Figure 4.29: The visualisation of the two-dimensional representation of the Churn (23 attributes), SEER Breast Cancer (960 attributes), Lips (4800 attributes), MNIST (784 attributes) generated by Isomap.



(a) Churn-SDD **(b)** SEER-SDD **(c)** LIPS-SDD **(d)** MNIST-SDD

(e) Predict-train Churn-SDD **(f)** Predict-train SEER-SDD **(g)** Predict-train LIPS-SDD **(h)** Predict-train MNIST-SDD

(i) Predict-test Churn-SDD **(j)** Predict-test SEER-SDD **(k)** Predict-test LIPS-SDD **(l)** Predict-test MNIST-SDD

Figure 4.30: The visualisation of the two-dimensional representation of the Churn (23 attributes), SEER Breast Cancer (960 attributes), Lips (4800 attributes), MNIST (784 attributes) generated by SDD.

Although the ratio of training/testing samples varies for different datasets, as shown in Table 4.8, it can be said that the employed ANN has been trained very good to embed the training samples and testing samples. Based on the experimental results shown in Table 4.7., it can be seen that the method that has been captured the best data structure is SDD to all datasets. As a result, the best structure of testing data has been captured by parametric SDD. Low dimensional visualisations generated by PCA, t-SNE, Isomap and SDD, and their prediction have been presented in Figs. 4.27, 4.28, 4.29, and 4.30, respectively.

In summary, parametric methods employ ANN to capture the same data structure as their corresponding versions. The better the training data structure has been captured, the better the data structure of testing data will be captured.

4.5 Chapter Summary

SDD demonstrated significantly higher performance than other dimensionality reduction methods (i.e., PCA, MDS, LLE, LE, Isomap, t-SNE, TriMap, and UMAP) from the conducted experiments in terms of the maintained data structure and computational time.

A summary of the simulations results of the three developed approaches, SDD, MSDD and parameter-free SDD, are presented in Table 4.9. From Table 4.9 can be seen that MSDD couldn't improve the scale of maintained data structure by SDD, and it significantly increased the computational time. As such, employing MSDD can be helpful in cases when the scale of maintained data structure is essential. On the other hand, parameter-free SDD has provided excellent performance in both the maintained data structure and computational time. Parameter-free SDD has demonstrated that it can capture a similar structure of data to SDD (which requires parameter-tunning) but in less computational time, shown in Table 4.9.

Overall, parameter-free SDD seems to be the best method between SDD and MSDD since it manages to capture the performance of SDD (considering that the improvement in performance that MSDD provides is negligible), and it is much faster than SDD and MSDD.

Also, the proposed parametric SDD performs similar to SDD. It mimics SDD using CNN in unseen data; even when the fraction of training samples is insufficient compared with the fraction of testing samples, parametric SDD captures the data structure very well. Parametric SDD compared to SDD, MSDD, and parameter-free SDD can be used in bigger datasets.

Table 4.9: THE PERFORMANCES OF SDD, MSDD AND PARAMETER-FREE SDD IN TERMS OF KENDALL'S TAU COEFFICIENT AND COMPUTATIONAL TIME

		Datasets			
		Iris	**Breast Cancer**	**Swiss Rolls**	**MNIST**
SDD	*deg*	8	10	1	1
	τ	0.967328	0.998118	0.914711	0.606525
	time(*seconds*)	9.49	111.33	552.96	2454.12
MSDD	*deg*	8 and 9	10 and 11	1 and 2	1 and 2
	τ	0.967309	0.998120	0.914707	0.598577
	time(*seconds*)	4.06	34.22	118.93	4177.75
MSDD	*deg*	7 and 8	9 and 10	0 and 1	0 and 1
	τ	0.967316	0.998122	0.914707	0.600533
	time(*seconds*)	3.04	25.82	90	3080.225
MSDD	*deg*	7, 8 and 9	9, 10 and 11	0, 1 and 2	0, 1 and 2
	τ	0.967334	0.998121	0.914709	0.598738
	time(*seconds*)	7.61	44	90	8086
Parameter-free SDD	τ	0.967339	0.998086	0.914578	0.607947
	time(*seconds*)	0.41	7.79	36.93	183.94

Chapter 5

The Impact of Dissimilarity Measures on Visualization and Classification Error

Supervised dimensionality reduction techniques are extensions of dimensionality reduction techniques, which employ dissimilarity measures instead of Euclidean distance to calculate the similarity between high dimensional space data samples. Although supervised dimensionality reduction techniques are not as widely used as their standard versions, they have been widely used to decrease the classification error and improve the maintained data structure. However, based on the literature gap, there are no theoretical analyses of the impact that dissimilarity measures on both classification error and maintained data structure. This Chapter is organised into three Sections: the impact of dissimilarity measure on structure maintaining: theoretical and practical analyses, the impact of dissimilarity measure is classification error: theoretical analyses, and Chapter summary.

5.1 The Impact of Dissimilarity Measures on Structure Maintaining

This Section investigates the impact that metrics used in a dimensionality reduction method have on high dimensional data structure maintenance. The dimensionality reduction methods considered are the manifold learning methods, which are the subcategory of dimensionality reduction methods but focus on learning the manifold that contains the low dimensional representation of the high dimensional data.

Standard manifold learning techniques use the Euclidean or Geodesic distance to calculate each data samples nearest neighbours in a manifold. On the other hand, supervised manifold learning techniques employ dissimilarity measures to calculate the nearest neighbours of each data sample. Dissimilarity measures dis_1 Eq. (2.107), dis_2 Eq. (2.108), and dis_3 Eq. (2.109) search the nearest neighbours by forcing the same class data samples to be close and/or forcing the different class data samples to be far away. As a consequence, for a given data sample, different neighbours set may be produced when using various measures such as Euclidean distances (dis), dis_1, dis_2, and dis_3. Each manifold learning technique seeks to keep the neighbourhood structure (neighbours set) defined in the high dimensional space data. Thus, four different low dimensional representations will be generated if four different neighbours sets have been defined in the high dimensional space data. However, the local neighbourhood structure of a manifold is determined using the Euclidean distance because a manifold is conceived to be a locally Euclidean space.

5.1.1 Theoretical Analysis

The theoretical analysis [97] of the impact of dissimilarity measures on structure maintenance is based on keeping the neighbourhood structure between two spaces. The neighbourhood structure has been considered as an order set. Maintaining the neighbourhood structure between high and low dimensional spaces is like having the same order set in high and low dimensional

spaces. To prove that two order sets are the same, then are used definitions about bijective functions, order preservation, and order isomorphism.

Let RO be the order set which contains the Euclidean distance of each data sample and its k-nearest neighbours of the high dimensional space data. Based on the manifold definition, Euclidean distance is the metric that calculates the local[1] neighbours for each data sample. Alternatively, it is defined with RO_1, RO_2, and RO_3 order sets that contain the distances of data samples and their k-nearest neighbours in the high dimensional using the dis_1, dis_2, and dis_3, respectively. It is also defined with ro_1, ro_2, and ro_3 order sets that contain distances of the low dimensional data samples and their k-nearest neighbours using the dis_1, dis_2, and dis_3, respectively. To simplify the analysis, it has been considered that every manifold learning approach has perfectly embedded data[2], and as a result, $ro = RO$, $ro_1 = RO_1$, $ro_2 = RO_2$, and $ro_3 = RO_3$.

A manifold learning technique maintains the manifold structure (the one that is locally Euclidean space) if the order set RO is the same with ro. To determine whether the neighbourhood structure has been captured, it must be proved whether $dis_0(a) = a$, dis_1, dis_2, and dis_3 are order isomorphism functions. In accordance with that, are used *Proposition 3*, *Definition 1*, and *Definition 2* that defines a function as order-isomorphism, bijective, and order-preservation, respectively.

Proposition 3 Let I and J be two order sets, then the function $f : I- > J$ is called an order-isomorphism function iff f is:

1. bijective, and
2. order-preservation (for all $a, b \in I$ so
   $a \leq b \Leftrightarrow f(a) \leq f(b)$).

Definition 1 A bijective function should be: 1) injective and 2) surjective. Let be I and J two sets, then the function $f : I- > J$ is injective if and only if whenever $f(a) = f(b)$ then $a = b$

[1]Define with local k-nearest neighbours.
[2]The manifold learning loss function has achieved its optimal value (zero).

for a, $b \in I$, and is surjective if and only if for every $d \in J$, there is at least one $c \in I$ such that $f(c) = d$.

Definition 2 Let I and J be two order sets, then the function $f : I- > J$ is called an order-preservation function iff for all elements $a, b \in I$, and $f(a), f(b) \in J$, $a \leq b \iff f(a) \leq f(b)$.

Consider $dis_0 : RO- > RO$, $dis_1 : RO- > RO_1$, $dis_2 : RO- > RO_2$, and $dis_3 : RO- > RO_3$. The functions dis_0, dis_1, dis_2, and dis_3 can be re-written as: $dis_0 : dis(x_i, x_j)- > dis(x_i, x_j)$,

$$dis_1 : dis(x_i, x_j)- > \begin{cases} \sqrt{1 - e^{\frac{-dis(x_i,x_j)^2)}{\beta}}} & l_i = l_j \\ \sqrt{e^{\frac{dis(x_i,x_j)^2)}{\beta}} - \alpha} & l_i \neq l_j \end{cases},$$

$$dis_2 : dis(x_i, x_j)- > \begin{cases} \frac{1}{\psi} dis(x_i, x_j) & l_i = l_j \\ dis(x_i, x_j) & l_i \neq l_j \end{cases},$$

$dis_3 :$

$$dis(x_i, x_j)- > \begin{cases} dis(x_i, x_j) & l_i = l_j \\ dis(x_i, x_j) + \max(dis(x_i, x_j))\mu & l_i \neq l_j \end{cases}.$$

Proposition 4 dis_0 is an order-isomorphism function whereas, dis_1, dis_2, and dis_3 are not order-isomorphism functions.

Proof. Based on Proposition 3, a function is order-isomorphism if it is: 1) bijective and 2) order-preservation. To check if dis_0, dis_1, dis_2, and dis_3 are order-isomorphism functions, firstly have to checked if they are bijective and order-preservation functions.

The first condition checks whether dis_0, dis_1, dis_2, and dis_3 are bijective functions.

1. dis_0 is a bijective, because it is injective and surjective. Suppose $a = dis(x_1, x_2), l(x_1) = l(x_2)$, $b = dis(x_1, x_3), l(x_1) \neq l(x_3)$, and $a = b = 2$. Since $dis_0 : dis(x_i, x_j)- > dis(x_i, x_j)$, then $dis_0(dis(x_1, x_2)) = dis(x_1, x_2) = 2$, and $dis_0(dis(x_1, x_3)) = dis(x_1, x_3) = 2 \Leftrightarrow dis_0(dis(x_1, x_2)) = dis_0(dis(x_1, x_3)) \Rightarrow dis_0$ is an injective function. dis_0 is also surjective, because $dis_0(dis(x_i, x_j)) = dis(x_i, x_j) \Leftrightarrow$ for every $dis(x_i, x_j)$, there exist at least

one $dis(x_i, x_j)$ that $dis_0(dis(x_i, x_j)) = dis(x_i, x_j)$.

2. The function $dis_0 : RO- > RO$ is an order-preservation function because the identity map is an order-preservation function.

In conclusion, dis_0 is a bijective and an order-preservation function; thus, it is an order-isomorphism function.

Let's check if dis_1 is bijective and order-preservation function.

1. Let $a = dis(x_1, x_2), l(x_1) = l(x_2)$, $b = dis(x_1, x_3), l(x_1) \neq l(x_3)$, where $a = b = 2$. We can prove that $dis_1(a) \neq dis_1(b)$. Let consider $\beta = 1$ and $\alpha = 0.5$, then $dis_1(dis(x_1, x_2)) = \sqrt{1 - e^{\frac{-2^2}{1}}} = 0.9908$ and $dis_1(dis(x_1, x_3)) = \sqrt{e^{\frac{2^2}{1}}} - 0.5 = 6.8890$. As a result $dis_1(dis(x_1, x_2)) \neq dis_1(dis(x_1, x_3))$.

2. To check if dis_1 is an order-preservation function, the order-preservation condition between each two order sets RO and RO_1 must be satisfied. Suppose $RO = \{dis(x_1, x_2), dis(x_1, x_3)\}$ and $RO_1 = \{dis_1(x_1, x_2), dis_1(x_1, x_3)\}$, where $dis(x_1, x_2) = 4$, and $dis(x_1, x_3) = 4.1$, thus, $RO = \{4, 4.1\}$. Conversely, x_1 and x_2 have different classes, and as a result, data samples x_1 and x_2 have been enforced to be far away with $dis_1(x_1, x_2) = 13.8919$. By contrast, data samples x_1 and x_3, which have the same class, have been enforced to be closer with $dis_1(x_1, x_3) = 0.9975$ for $\alpha = 0.5$, and as a conclusion, dis_1 is not an order-preservation function.

Since dis_1 is not injective function, it is not bijective function. Furthermore, dis_1 is not order-preservation function; as a conclusion it is not order-isomorphism function. Like dis_1, dis_2 and dis_3 are not bijective and order-preservation functions. Note that dis_2 favours the same class neighbours by decreasing their Euclidean distance with a positive value ψ. On the other hand, dis_3 favours the same class data samples by increasing the distance between data samples from different classes. As a result, the local manifold structures defined by dis_2 and dis_3 are not the same as the manifold structure defined by Euclidean distance, which is the distance that a manifold is assumed to use.

Overall, dis_0 is a bijective and an order-preservation function, such that it is an order-isomorphism function. On the other hand, dis_1, dis_2, and dis_3 are neither bijective functions or order-preservation functions, and subsequently they are not order-isomorphism functions. Thus, the low dimensional visualization produced by a manifold learning using dissimilarity measure is not the best representation of the high dimensional data structure. ■

To better understand the impact of dissimilarity measure on manifold learning techniques in terms of structure capturing, we apply Breast Cancer data in Isomap (uses Euclidean distance) and Supervised Isomap (uses dis_1), illustrated in the next subsection.

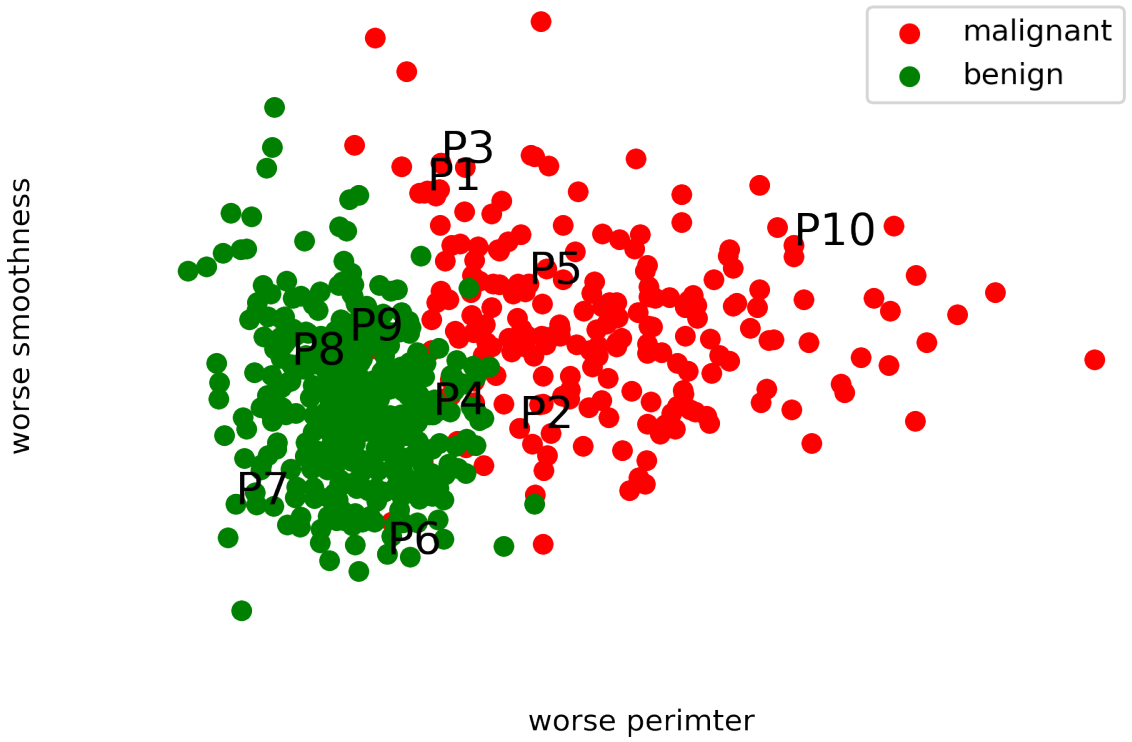
5.1.2 Practical Analysis

Visualisation helps explore and understand high dimensional space data. Providing a visualisation that does not maintain the neighbourhood structure of the high dimensional space data samples can lead to wrong decision-making. This Section provides a practical example of the impact of using a dissimilarity measure instead of Euclidean distance in dimensionality reduction methods on visualisation and decision-making in real-world Breast cancer data.

Breast Cancer data has been selected to demonstrate practically the impact of dissimilarity measure on structure capturing. To simplify the demonstration, have considered two variables *worse perimeter* and *worse smoothness* of the Breast Cancer data[3] and then have been considered ten randomly-selected data samples. Each data sample corresponds to a patient, and then the neighbours rank indexes[4] for each selected patient have been constructed, as shown in Fig. 5,1. Considering Patient 1; the nearest neighbour of Patient 1 is Patient 4 (rank 2), followed by Patient 3 (rank 3), Patient 6 (rank 4), Patient 9 (rank 5), Patient 2 (rank 6), Patient 5 (rank 7), Patient 8 (rank 8), Patient 7 (rank 9), and Patient 10 (rank 10).

To evaluate which of the methods has retained the data structure better, is constructed a difference matrix named *Retained-Structure* that contains the difference between the neighbourhood

[3]Breast Cancer with 569 samples and 30 variables from Sklearn, Python.
[4]The neighbourhood ranking index demonstrates the neighbourhood ranking index among patients.



(a)

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
P1 rank index	1	5	3	2	5	3	5	5	4	6
P2 rank index	6	1	5	6	2	7	8	8	8	3
P3 rank index	3	3	1	3	3	5	7	7	6	4
P4 rank index	2	4	2	1	4	4	6	6	5	5
P5 rank index	7	2	6	7	1	8	9	9	9	2
P6 rank index	4	6	4	4	6	1	4	4	2	7
P7 rank index	9	10	9	9	10	9	1	2	7	10
P8 rank index	8	8	8	8	8	6	2	1	3	9
P9 rank index	5	7	7	5	7	2	3	3	1	8
P10 rank index	10	9	10	10	9	10	10	10	10	1

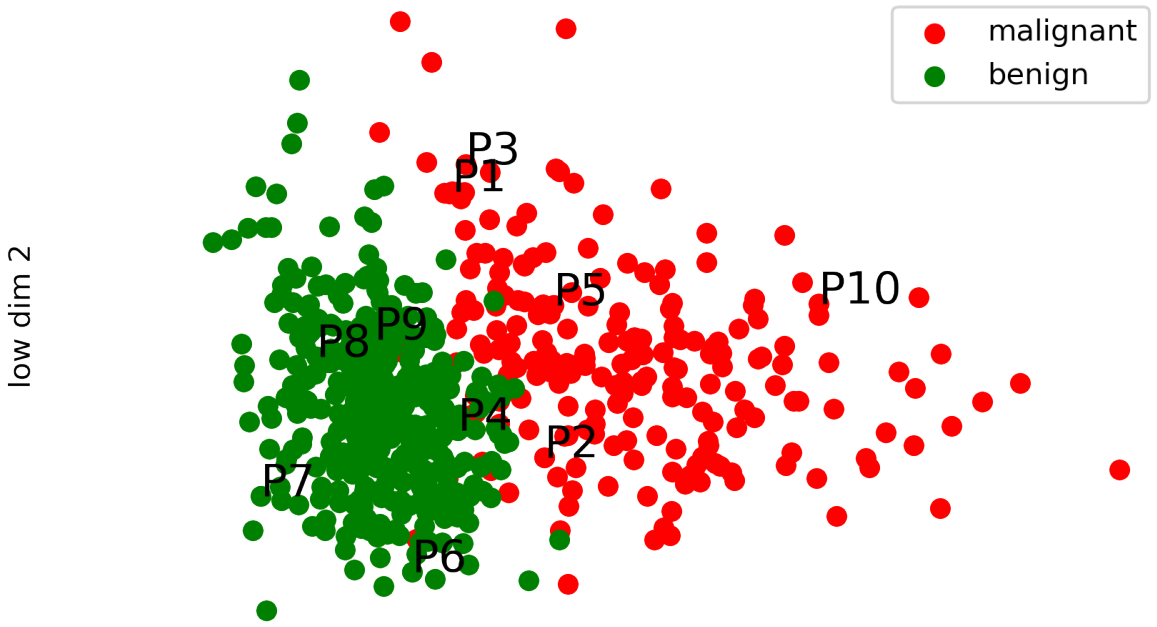
(b)

Figure 5.1: The visualisation of *worse perimeter* and *worse smoothness* variables from Breast Cancer dataset (a), and the neighbourhood rank indexes between ten randomly selected patients (b).

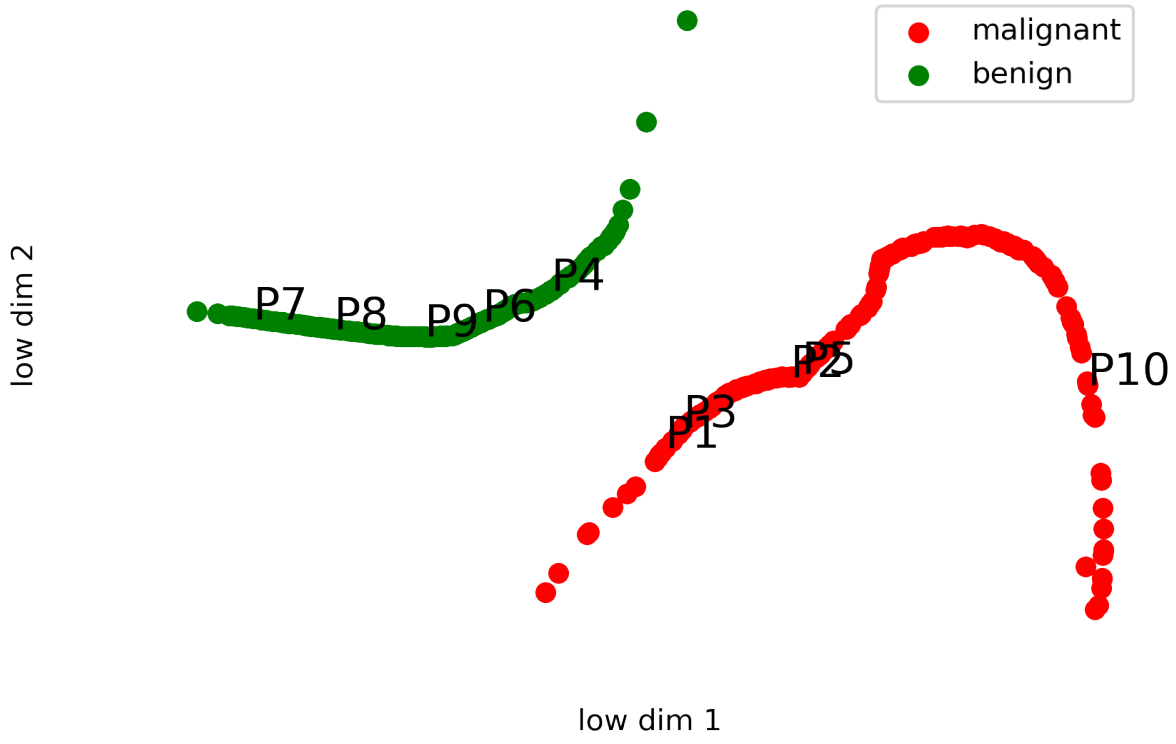
rank matrix of the high and the low dimensional space data. In an ideal case, the Retained-Structure matrix contains only element 0 (zero). Nonzero elements, which indicate a failure in retaining the neighbourhood structure, are positive or negative numbers. A positive number $P_{ij} = +in$ indicates that the method has jumped $+in$ positions closer the j^{th} data sample to the i^{th} data sample. By contrast, a negative number $P_{ij} = -in$ indicates that the method has been forced the i^{th} data sample to be in positions further away from the j^{th} data sample. In terms of medical interpretation, it can be said that *worse perimeter* and *worse smoothness* variables of Patient 1 are the most similar to Patient 4 and the least similar to Patient 10. Thus, if applying any manifold learning technique to the above-considered data, the best manifold learning (dimensionality reduction) method is the one that maintains the neighbourhood structure. In other words, Patient 1 should maintain the neighbours rank in the following order: Patient 4, Patient 3, Patient 6, Patient 9, Patient 2, Patient 5, Patient 8, Patient 7, and Patient 10 from the closest to the most distant patient.

To demonstrate the impact of a dissimilarity measure on structure capturing, it has been applied dis_1 to Isomap and have compared with the standard Isomap. Visually, supervised Isomap with dis_1 seems better, as samples of the same class are closer, and samples of different classes have become more separated. However, the visualization of standard Isomap seems more similar to the visualization of the original data, which is discussed below.

The visulizations of low dimensional represenations generated by Isomap and Supervised Isomap are showed in Fig. 452(a) and Fig. 5.2(b), respectively. Fig. 5.3 shows that the method that has captured the neighbourhood structure entirely is Isomap, as its Retained-Structure matrix Fig. 5.4(a) contains only elements 0. Contrastingly, the supervised Isomap has failed to maintain the neighbourhood structure, demonstrated by nonzero elements in the Retained-Structure matrix Fig. 5.4(b). Patients are organized into two classes where Patient 1, Patient 2, Patient 3, Patient 5, and Patient 10 are patients diagnosed with *malignant*, whereas Patient 4, Patient 6, Patient 7, Patient 8, and Patient 9 are patients diagnosed with *benign*. There can be spoted from the Retained-Structure matrix Fig. 5.4(b) that the same class samples have been forced to be closer, demonstrated by negative values in the Retained-Structure matrix, shown in Fig. 5.4(b). Different class patients have been forced to be further away, illustrated by positive values



(a)



(b)

Figure 5.2: The visualization of low dimensional representation of Isomap (a) and the visualization of low dimensional representation of Supervised Isomap.

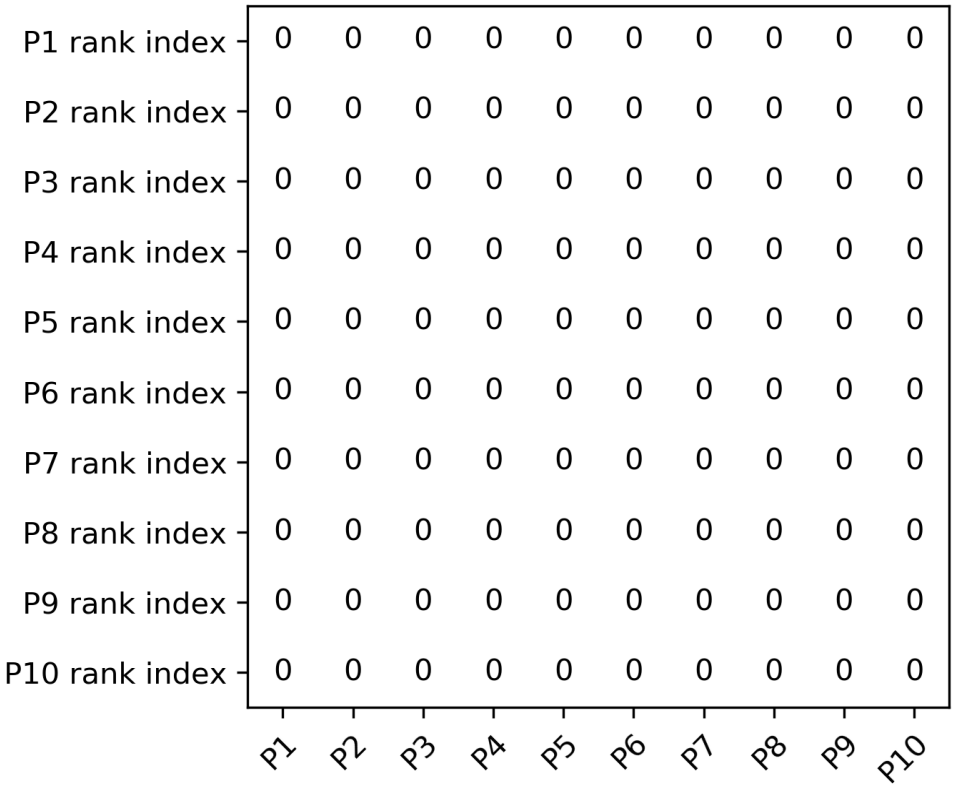
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
P1 rank index	1	5	3	2	5	3	5	5	4	6
P2 rank index	6	1	5	6	2	7	8	8	8	3
P3 rank index	3	3	1	3	3	5	7	7	6	4
P4 rank index	2	4	2	1	4	4	6	6	5	5
P5 rank index	7	2	6	7	1	8	9	9	9	2
P6 rank index	4	6	4	4	6	1	4	4	2	7
P7 rank index	9	10	9	9	10	9	1	2	7	10
P8 rank index	8	8	8	8	8	6	2	1	3	9
P9 rank index	5	7	7	5	7	2	3	3	1	8
P10 rank index	10	9	10	10	9	10	10	10	10	1

(a)

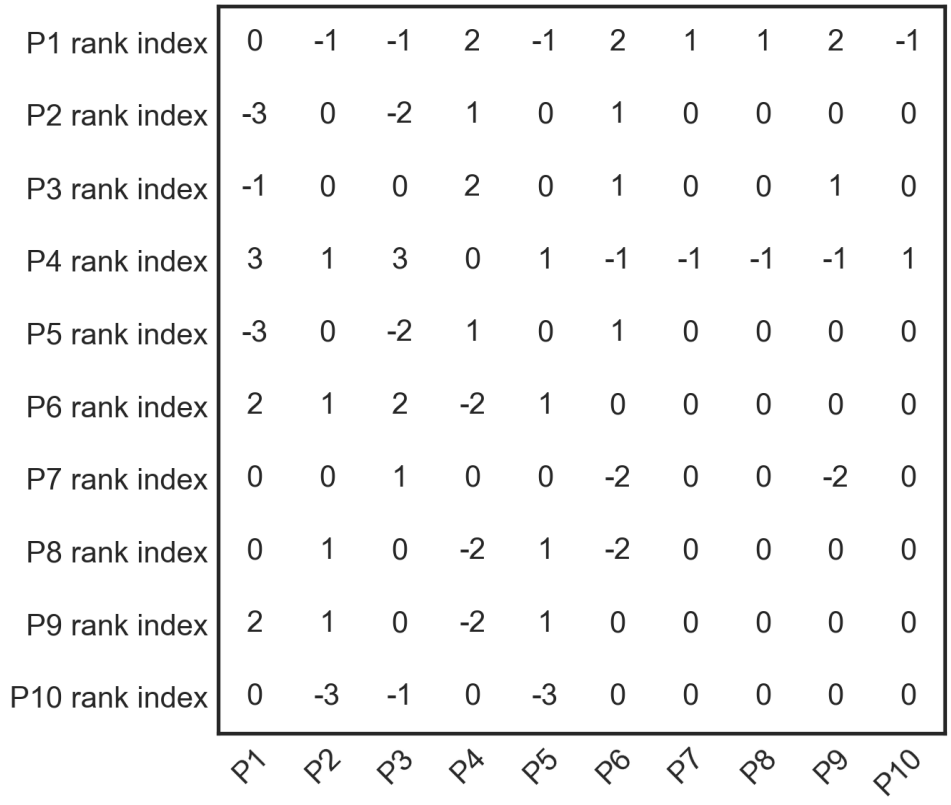
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
P1 rank index	1	4	2	4	4	5	6	6	6	5
P2 rank index	3	1	3	7	2	8	8	8	8	3
P3 rank index	2	3	1	5	3	6	7	7	7	4
P4 rank index	5	5	5	1	5	3	5	5	4	6
P5 rank index	4	2	4	8	1	9	9	9	9	2
P6 rank index	6	7	6	2	7	1	4	4	2	7
P7 rank index	9	10	10	9	10	7	1	2	5	10
P8 rank index	8	9	8	6	9	4	2	1	3	9
P9 rank index	7	8	7	3	8	2	3	3	1	8
P10 rank index	10	6	9	10	6	10	10	10	10	1

(b)

Figure 5.3: The neighbourhood rank indexes of the low dimensional space data generated by Isomap (a), and Supervised Isomap (b).



(a)



(b)

Figure 5.4: Retained-Structure matrix of Isomap (a), and Supervised Isomap (b).

in the Retained-Structure, matrix shown in Fig. 5.4(b). There can be concluded that forcing data samples to be closer or further away impacts the scale of maintaining the neighbourhood structure. As shown in Fig. 5.2(b), Patient 1 was more similar to Patient 4 in terms of *worse perimeter* and *worse smoothness* variables. However, using supervised Isomap, the nearest patient to patient 1 is Patient 3, shown in Fig. 5.3(b). Consequently, there can be assumed that Patient 1 and Patient 3, which are very close in the visualization of low dimensional representation, may need the same treatment. However, Patient 1 and Patient 3 have different corresponding values of *worse perimeter* and *worse smoothness* in the original data. As a result, the aforementioned decision for the same treatment may be wrong.

5.2 The Impact of Dissimilarity Measures on Clasiffication Error

A manifold learning technique can be employed as a pre-processing step for classification. However, the priority of a manifold learning technique is to capture data structure instead of separate data samples of different classes. Consequently, researchers have proposed class information in calculating the similarity between data samples (dissimilarity measures), i.e., dis_1, dis_2, and dis_3, in manifold learning to achieve a lower classification error. This section discusses how the dissimilarity measure affects a classification model to achieve a lower classification error. The theoretical analysis is based on the work of Balcan et al. [98] who proposed the (ϵ, γ) good similarity function based on intuitive and sufficient conditions that allow a similarity function to learn well, supported by *Definition 3*, *Definition 4*, *Theorem 1*, and *Theorem 2*.

Consider a manifold learning M that generates low dimensional data Y, Y_1, Y_2, and Y_3 using metrices dis (Euclidean distance), dis_1, dis_2, and dis_3, respectively. To simplify our analysis, we consider that the manifold learning method M has performed perfectly (the loss function employed in the manifold learning has reached i.e., its minimal value (zero)), such that the neighbourhood structures defined in the high dimensional space using Euclidean distance (dis), dis_1, dis_2, and dis_3 are preserved completely. Note that the neighbourhood structures defined

using dis, dis_1, dis_2, and dis_3 are the same with the neighbourhood structure defined using dis in the low dimensional data Y, Y_1, Y_2, and Y_3, respectively.

Definition 3 (Balcan et al. [98]) A similarity function over Y is any pairwise function $K : Y \times Y -> [-1, 1]$.

Definition 4 (Balcan et al. [98]) K is a strongly (ϵ, γ) good similarity function, if at least a- $(1-\epsilon)$ probability mass of examples y satisfy: $E_{y-Y}[dis(y, y')|l(y') \neq l(y)] > E_{y-Y}[dis(y, y')|l(y') = l(y)] + \gamma$.

Theorem 1 (Balcan et al. [98]) *If K is a valid kernel function, and is (ϵ, γ)-good similarity for some learning problem, then it is also (ϵ, γ)-kernel-good for the learning problem.*

Theorem 2 (Balcan et al. [98]) *If dis is a strongly (ϵ, γ) - good similarity function, then $\frac{4}{\gamma^2}\ln(\frac{2}{\delta})$ positive S^+ examples, and S^- negative examples are sufficient, so with probability $p \geq 1 - \delta$, the above algorithm produces a classifier with a maximum error of $\epsilon + \frac{\delta}{2}$.*

In the work of Balcan et al. [98], a learning problem was specified by a labelled example (x, y) drawn from a distribution of P over $X \times \{-1, 1\}$, where X is an abstract space. In this study, the learning problem is defined by providing the low dimensional space data (y, l), (y_1, l), (y_2, l), and (y_3, l) generated by a manifold learning method M over data $X \times \{-1, 1\}$ using the dis, dis_1, dis_2, and dis_3, respectively. The objective of a learning algorithm is to produce a classification function $g_i : Y_i \rightarrow \{-1,\ 1\}, i = 0 : 3$ to produce a low classification error.

The purpose of this analyse is to discover the goodness of a similarity function in a particular learning problem. In other words, using the same similarity function K, but in different data distribution (the low dimensional data Y, Y_1, Y_2, and Y_3 generated by the manifold learning M employing dis, dis_1, dis_2, and dis_3) having the same label l. Note that for a given i, $l(x_i) = l(y_i) = l(y_{1i}) = l(y_{2i}) = l(y_{3i})$. Consider that K is the radial basis function (RBF) kernel with formula, $K(x, x') = exp(-\frac{dis(x,x')^2}{2\sigma^2})$, *Theorem 1* states that a kernel function is a good similarity function; as such the theorems and definitions applied for similarity functions can also be applied for kernel functions. Standard algorithms such as Support Vector Machine (SVM) and Perceptron have used kernel functions to learn linear separations via computing dot

products on pairs of examples. The main idea of applying kernel function is to map nonlinear data in a very high dimensional space to find a hyperplane to separate data. This study employed the RBF kernel, which is usefully used in an SVM-based classifier.

The neighbourhood structure of Y, Y_1, Y_2, and Y_3 is the same as the neighbourhood structure of X using dis, dis_1, dis_2, and dis_3, since the manifold learning M has assumed to perfectly maintain the neighbourhood structure. Thus, the K function that is assumed to be applied to the low dimensional space Y, Y_1, Y_2, and Y_3 using the squared Euclidean distance dis, can be equally applied to the high dimensional data X, but using dis, dis_1, dis_2, and dis_3, respectively. As a result, four RBF kernel functions $K(y, y')$, $K(y_1, y'_1)$, $K(y_2, y'_2)$, and $K(y_3, y'_3)$ can be reformulated as follows:

1. $K(y, y') = exp(-\frac{dis(x,x')^2}{2\sigma^2})$
2. $K(y_1, y'_1) = exp(-\frac{dis_1(x,x')^2}{2\sigma^2})$
3. $K(y_2, y'_2) = exp(-\frac{dis_2(x,x')^2}{2\sigma^2})$
4. $K(y_3, y'_3) = exp(-\frac{dis_3(x,x')^2}{2\sigma^2})$

The aim is to prove that the RBF kernel can produce a lower classification error using low dimensional data Y_1, Y_2, and Y_3 than using low dimensional data Y. Let U represents the set of y that satisfy $E_{y'\sim Y}[K(y, y')|l(y) = l(y')] \geq E_{y'\sim Y}[K(y, y')|l(y) \neq l(y')] + \gamma$, and $P(U) = 1 - \epsilon$.

Proposition 5 RBF kernel K achieves a lower classification error using the low dimensional data Y_1 than using the low dimensional data Y.

Proof.

Let U_1 denotes the set of y_1 that satisfy:
$E_{y'_1\sim Y_1}[K(y_1, y'_1)|l(y_1) = l(y'_1)] \geq E_{y'_1\sim Y_1}[K(y_1, y'_1)|l(y_1) \neq l(y'_1)] + \gamma$.
Since $K(y_1, y'_1) = exp(-\frac{dis_1(x,x')^2}{2\sigma^2})$, then
$E_{x'\sim X} e^{-\frac{1-e^{\frac{-dis(x,x')}{\beta}}}{2\sigma^2}} |l(y) = l(y') \geq E_{x'\sim X} e^{-(\frac{e^{\frac{dis(x,x')^2}{\beta}}}{2\sigma^2} - \alpha)} |l(y) \neq l(y') + \gamma$.

For $\alpha \geq 0.5$, $e^{\frac{dis(x,x')^2}{\beta}} - \alpha \geq 1$, and $1 - e^{-\frac{dis(x,x')^2}{\beta}} \in [0, 1[$, as such
$E_{x' \sim X}[e^{-\frac{1-e^{\frac{-dis(x,x')}{\beta}}}{2\sigma^2}} | l(y) = l(y')] \geq E_{x' \sim X}[e^{-\frac{dis(x,x')^2}{2\sigma^2}} | l(y) = l(y')] \geq E_{x' \sim X}[e^{-\frac{dis(x,x')^2}{2\sigma^2}} | l(y)] \neq l(y') \geq E_{x' \sim X}[e^{-\frac{e^{\frac{-dis(x,x')^2}{\beta}}}{2\sigma^2}} - \alpha$
$|l(y) \neq l(y')] + \gamma$.

Finally, $U_1 = U \cup R_1$, where R_1 contains data samples x that satisfy:
$E_{x' \sim X} e^{-\frac{1-e^{\frac{-dis(x,x')}{\beta}}}{2\sigma^2}} | l(y) = l(y') \geq E_{x' \sim X} e^{-\frac{dis(x,x')^2}{2\sigma^2}} | l(y) = l(y')$
and data samples x that satisfy:
$E_{x' \sim X} e^{-\frac{dis(x,x')^2}{2\sigma^2}} | l(y) \neq l(y') \geq E_{x' \sim X} e^{-\frac{e^{\frac{-dis(x,x')^2}{\beta}} - \alpha}{2\sigma^2}} | l(y) \neq l(y') + \gamma$.
Therefore, $P(U_1) = P(U \cup R_1) = P(U) + P(R_1)$ as $U \cap R_1 = \varnothing$.

Let's define $P(R_1) = \rho_1$, as such $P(U_1) = 1 - \epsilon + \rho_1$.

Based on *Definition 4*, RBF kernel in the low dimensional data Y_1 is a strongly $(\epsilon - \rho_1, \gamma)$-good similarity function, whereas RBF kernel in the low dimensional data Y is strongly (ϵ, γ)-good similarity function. Under the conditions of *Theorem 2*, the classification error of RBF kernel using the low dimensional data Y_1 is $\epsilon - \rho_1 + \frac{\delta}{2}$ which is lower than $\epsilon + \frac{\delta}{2}$ produced by RBF kernel low dimensional data Y.■

Proposition 6 RBF kernel K achieves a lower classification error using the low dimensional data Y_2 than using the low dimensional data Y.

Proof.

Let's define U_2 as the set of y_2 that satisfy:
$E_{y_2' \sim Y_2}[K(y_2, y_2') | l(y_2) = l(y_2')] \geq E_{y_2' \sim Y_2}[K(y_2, y_2') | l(y_2) \neq l(y_2')] + \gamma$.
Since $K(y_2, y_2') = exp(-\frac{dis_2(x,x')^2}{2\sigma^2})$, then
$E_{x' \sim X} e^{-\frac{dis(x,x')^2}{\psi^2 2\sigma^2}} | l(y) = l(y') \geq E_{x' \sim X} e^{-\frac{dis(x,x')^2}{2\sigma^2}} | l(y) \neq l(y') + \gamma$.
$e^{-\frac{(dis(x,x'))^2}{2\psi^2\sigma^2}} = (e^{-\frac{dis(x,x')^2}{2\sigma^2}})^{1/\psi^2}, (e^{-\frac{dis(x,x')^2}{2\sigma^2}})^{1/\psi^2} \geq e^{-\frac{dis(x,x')^2}{2\sigma^2}}, \psi \geq 1$.

As a result,
$E_{x' \sim X}[(e^{-\frac{dis(x,x')^2}{2\sigma^2}})^{1/\psi^2} | l(y) = l(y')] \geq E_{x' \sim X}[e^{-\frac{dis(x,x')^2}{2\sigma^2}} | l(y) = l(y')] \geq E_{x' \sim X}[\frac{e^{-\frac{dis(x,x')^2}{2\sigma^2}}}{c} | l(y) \neq l(y')] + \gamma$.

On the other hand, $U_2 = U \cup R_2$, where R_2 contains data samples x that satisy:
$E_{x`\sim X}[(e^{-\frac{dis(x,x')^2}{2\sigma^2}})^{1/\psi^2}|l(y) = l(y')] \geq E_{x`\sim X}[e^{-\frac{dis(x,x')^2}{2\sigma^2}}|l(y) = l(y')]$.
Thus, $P(U_2) = P(U \cup R_2) = P(U) + P(R_2))$ as $U \cap R_2 = \varnothing$. Let's define $P(R_2) = \rho_2$, such that $P(U_2) = 1 - \epsilon + \rho_2$.

Based on *Definition 4*, it can be said that RBF kernel in the low dimensional data Y_2 is a strongly $(\epsilon - \rho_2, \gamma)$-good similarity function, and RBF kernel in the low dimensional data Y is a strongly (ϵ, γ)-good similarity function. Under the conditions of *Theorem 2*, the classification error of RBF kernel using the low dimensional data Y_2 is $\epsilon - \rho_2 + \frac{\delta}{2}$, which is lower than $\epsilon + \frac{\delta}{2}$ produced by RBF kernel using Y. ■

Proposition 7 RBF kernel K achieves a lower classification error using the low dimensional data Y_3 than using the low dimensional data Y.

Proof.

Suppose that U_3 is the set of all y_3 that satisfy:
$E_{y_3'\sim Y_3}[K(y_3, y_3')|l(y_3) = l(y_3')] \geq E_{y_3'\sim Y_3}[K(y_3, y_3')|l(y_3) \neq l(y_3')] + \gamma$.
Because $K(y_3, y_3') = exp(-\frac{dis_3(x,x')^2}{2\sigma^2})$, then
$E_{x'\sim X}e^{-\frac{dis(x,x')^2}{2\sigma^2}}|l(y) = l(y') \geq E_{x'\sim X}e^{-\frac{(dis(x,x')+dis(x,x')\mu)^2}{2\sigma^2}}|l(y) \neq l(y') + \gamma$.
On the other hand, $e^{-\frac{(dis(x,x')+maxdis(x,x')\mu)^2}{2\sigma^2}} = e^{-\frac{dis(x,x')^2}{2\sigma^2}}e^{-\frac{(maxdis(x,x')\mu)^2}{2\sigma^2}}$.
Let be $c = e^{\frac{(maxdis(x,x')\mu)^2}{2\sigma^2}}$, and $c \geq 1$, then
$E_{x`\sim X}[e^{-\frac{dis(x,x')^2}{2\sigma^2}}|l(y) = l(y')] \geq E_{x`\sim X}[\frac{e^{-\frac{dis(x,x')^2}{2\sigma^2}}}{c}|l(y) \neq l(y')] + \gamma$.
On the other hand, $U_3 = U \cup R_3$, where R_3 contains the x data samples that satisy:
$E_{x'\sim X}[\frac{e^{-dis(x,x')^2}}{c}|l(y) \neq l(y')] + \gamma \leq E_{x'\sim X}[e^{-\frac{dis(x,x')^2}{2\sigma^2}}|l(y) = l(y')] \leq E_{x'\sim X}[e^{-\frac{dis(x,x')^2}{2\sigma^2}}|l(y) \neq l(y')] + \gamma$.
As a result, $P(U_3) = P(U \cup R_3) = P(U) + P(R_3))$ as $U \cap R_3 = \varnothing$. Let's define $P(R_3) = \rho_3$, such that $P(U_3) = 1 - \epsilon + \rho_3$.

Based on *Definition 4*, it has been proved that RBF kernel in the low dimensional data Y_3 is strongly $(\epsilon - \rho_3, \gamma)$-good similarity function, whereas RBF kernel applied in the low dimensional data Y is strongly (ϵ, γ)-good similarity function. Under the conditions of *Theorem 2*, the

classification error of RBF kernel using the low dimensional data Y_3 is $\epsilon - \rho_1 + \frac{\delta}{2}$, which is lower than $\epsilon + \frac{\delta}{2}$ produced by RBF kernel using the low dimensional data Y.■

Overall, RBF kernel applied in the low dimensional data generated by a manifold learning M[5] using dissimilarity measures dis_1, dis_2, and dis_3 can help a learning problem to achieving lower classification errors than RBF kernel applied in the low dimensional data generated by the manifold learning M using the Euclidean distance dis.

5.3 Chapter Summary

This Chapter has provided theoretical and practical analysis of the impact of dissimilarity measure is structure maintaining, followed by a theoretical analysis on the impact of dissimilarity measure in the classification accuracy.

It has been demonstrated that dissimilarity measures theoretically does not improve the structure maintained but instead destroy it. Also, it has been practically demonstrated that employing dissimilarity measures in a dimensionality reduction method (Isomap in that case) does not generate the real structure of the high dimensional data. Generating low dimensional data that does not represent the real structure of high dimensional data leads end-users to make the wrong decisions. On the other hand, there is theoretically proven that dissimilarity measures can improve classification accuracy.

[5]Note that manifold learning M perfectly preserves the neighborhood structure using dis, dis_1, dis_2, and dis_3.

Chapter 6

Experiments of the Impact of Dissimilarity measures on Structure Capturing

This Chapter is organised into two main Sections, experimental results of supervised methods and Chapter Summary. Three different dissimilarity measures and Euclidean distance have been used to calculate the similarity between high dimensional data samples in three dimensionality reduction methods Isomap, t-SNE and LE. Their performance has been evaluated using Kendall's Tau and Co-ranging matrix in two datasets, Breast cancer and Swiss roll data. The Chapter Summary summarises the main findings of experimental results.

6.1 Experiments and Discussions

Isomap, t-SNE, and LE are three manifold learning (dimensionality reduction) techniques considered to demonstrate the impact of dissimilarity measures in maintaining the data structure in the dimensionality reduction process. Three considered methods have been tested with two datasets, Breast Cancer and Swiss Roll, using Euclidean distance and three dissimilarity measures dis_1 Eq. (2.107), dis_2 Eq. (2.108), and dis_3 Eq. (2.109). This study will discard

considering dis_4 Eq. (2.110) as it includes temporal information, and the datasets considered in this study do not include any temporal information. They were implemented in Python using the corresponding Sklearn versions and the same number of iterations (2000). Supervised manifold learning methods were also implemented using their Sklearn versions, but by selecting the *pre-computed* metric, we pre-computed the dissimilarity measures separately. Their performance in maintaining the neighbourhood structure of data in a manifold has been evaluated by Kendall's Tau coefficients [41] and Co-ranking matrices [46]. Furthermore, the number of neighbours for each method has been tuned from 1 to $N-1$ because it substantially impacts the scale of preserving the neighbourhood structure of a manifold.

6.1.1 Breast Cancer

Breast Cancer data with 569 data samples (patients), thirty variables and two classes is the first dataset considered. The thirty-dimensional data will be transformed into two-dimensional space data (visualization in Fig. 6.1) by employing four different metrics: Euclidean distance, dis_1, dis_2, and dis_3 to Isomap, t-SNE, and LE. Their performances have been evaluated by Kendall's Tau coefficients presented in Table 6.1, and Co-ranking matrices demonstrated in Fig. 6.2.

Table 6.1: KENDALL'S TAU FOR METHODS (COLUMNS) USING METRICS (ROWS) IN BREAST CANCER DATA

	Methods		
Metric	Isomap	t-SNE	LE
Euclidean	0.9977	0.8150	0.7267
dis_1	0.8288	0.7291	0.3878
dis_2	0.8528	0.7025	0.0941
dis_3	0.3192	0.8137	0.0988

The experiments conducted on Breast Cancer data show that Euclidean distance helps Isomap (k: 515) to capture the best data structure as demonstrated by a nearly diagonal Co-ranking matrix demonstrated in Fig.6.2(a), and Kendall's Tau coefficient with 0.9977, as shown in Table 4.9. The dis_1, dis_2, and dis_3 used in Isomap are less useful in capturing the neighbourhood structure, estimated by Kendall's Tau coefficients (Table 6.1), and the Co-ranking matrices

(Fig. 6.1). The Euclidean distance has resulted in the best metric for t-SNE, regarding the maintenance of the data structure, with a Kendall's Tau coefficient of 0.8150. However, dis_3 demonstrated excellent performance by competing with Euclidean distance for t-SNE. Note that the Gaussian distribution becomes broader because if σ increases and the broader the Gaussian distribution is, the more sensitive it becomes to more distant neighbours. This conclusion is supported by the result of the Co-ranking matrix of t-SNE using dis_3, which has fewer off-diagonal entries. Contrastingly, dissimilarity measure dis_2 enforces the data samples of the same class to have a smaller distance; and as such, the number of data samples with small distances becomes higher. As a result, the Gaussian distribution(s), which relates to the density of data σ, becomes sharp when the density is small.

LE using the Euclidean distance preserves the data structure better than using other metrics, supported by their Co-ranking matrices and Kendall's Tau coefficients. The dissimilarity measure dis_1 employed to LE reduces Kendall's Tau coefficient by 0.3878, as demonstrated in Table 6.1. The deterioration of the structure preservation can be seen in the respective Co-ranking matrices, as shown in Figs. 6.2 (j), in which the supervised LE has more off-diagonal entries.

6.1.2 Swiss Roll

The second dataset considered is the three-dimensional Swiss Roll data with 1600 data samples, which will be transformed into two-dimensional space data (shown in Fig. 6.3), by using four different metrics including dis, dis_1, dis_2, and dis_3 in Isomap, t-SNE, and LE.

Performances of Isomap, t-SNE, and LE using dis (Euclidean distance), dis_1, dis_2, and dis_3 with Swiss Roll data, were estimated using Kendall's Tau coefficients as shown in Table 6.2, and Co-ranking matrices illustrated in Fig. 6.4. The two-dimensional data visualizations are demonstrated in Fig. 6.3. Based on Kendall's Tau coefficient values and Co-ranking matrices, manifold learning techniques that employ Euclidean distance, have preserved better Swiss Roll data than three other metrics (dis_1, dis_2, and dis_3). Among unsupervised manifold learning methods, Isomap (Euclidean distance) captures the best Swiss Roll data structure, with

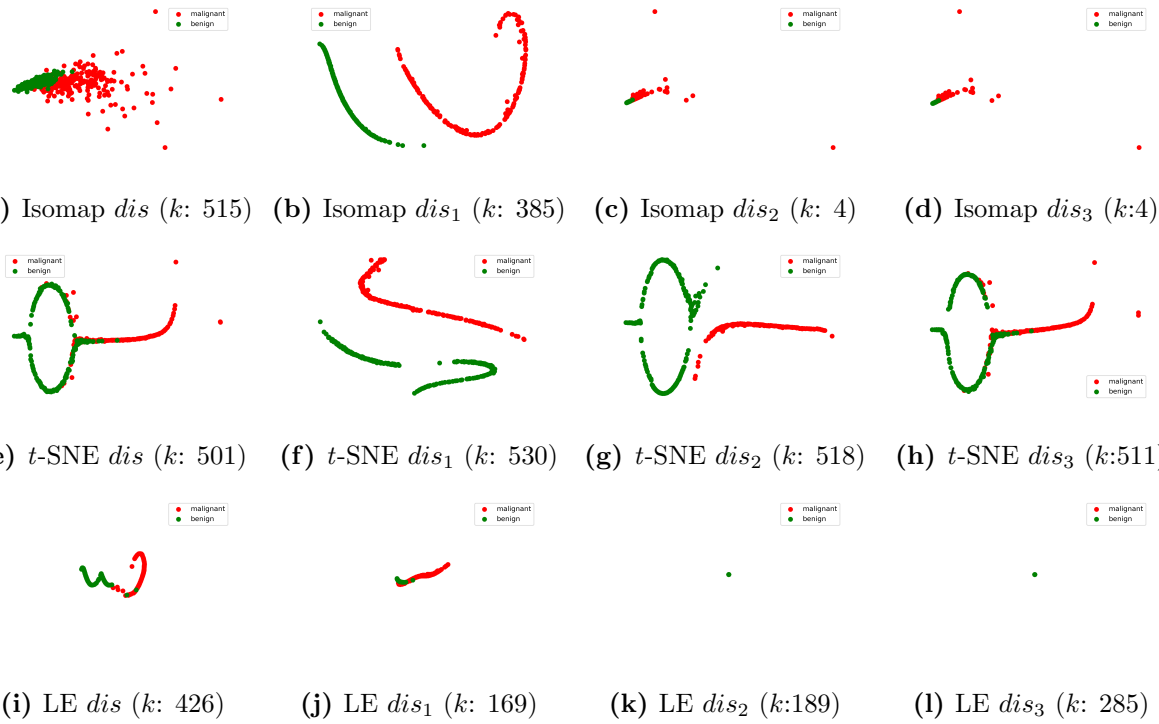
Kendall's tau 0.9121. The LE with dis_1 captures the best data structure across supervised methods, with Kendall's tau 0.8508.

Unlike with Breast Cancer data, in Swiss Roll data t-SNE managed to capture the highest data structure by using Euclidean distance and not a dissimilarity measure. However, among dissimilarity measures, dis_3 resulted in capturing global data structure the best (t-SNE shown in Fig. 6.3(h)). As previously noted, the broader the distance range of data, the broader the Gaussian distribution and the more sensitive to large distances it is, the more it improves the data structure capturing.

Table 6.2: KENDALL'S TAU FOR METHODS (COLUMNS) USING METRICS (ROWS) IN SWISS ROLL DATA

	Methods			
Metric		Isomap	t-SNE	LE
	Euclidean	0.9121	0.8700	0.9043
	dis_1	0.8269	0.8268	0.8508
	dis_2	0.2473	0.7686	0.7120
	dis_3	0.3192	0.8460	0.8515

Overall, employing a dissimilarity measure in a manifold learning technique does not improve data structure preservation. However, in some scenarios, dis_3 helps t-SNE to capture a more global data structure, but it may lose some local information.

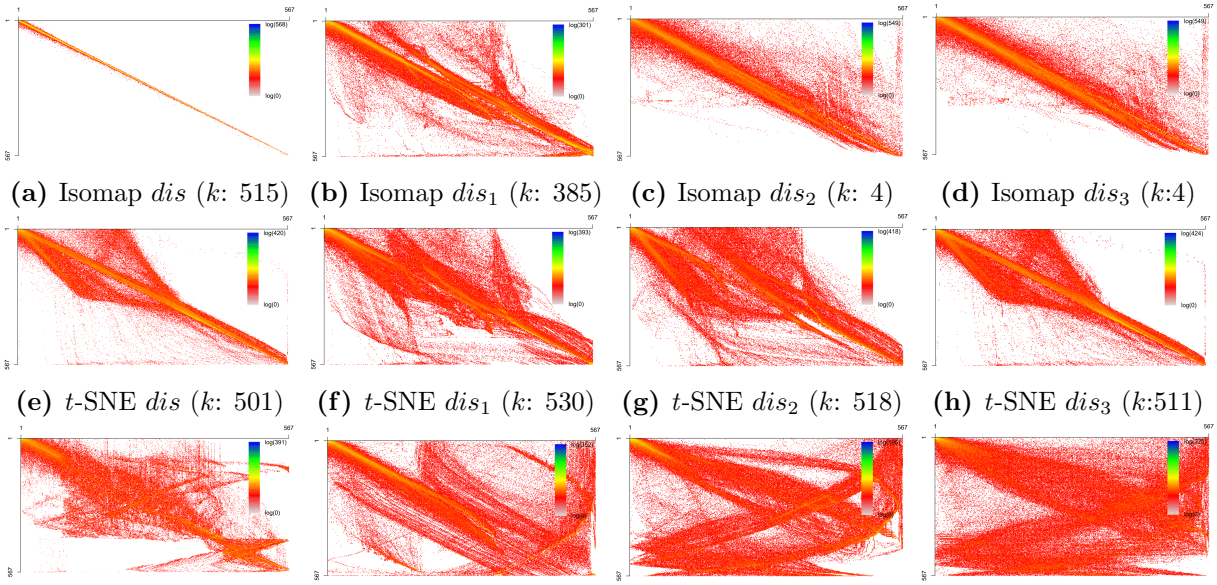


(a) Isomap dis (k: 515) **(b)** Isomap dis_1 (k: 385) **(c)** Isomap dis_2 (k: 4) **(d)** Isomap dis_3 (k:4)

(e) t-SNE dis (k: 501) **(f)** t-SNE dis_1 (k: 530) **(g)** t-SNE dis_2 (k: 518) **(h)** t-SNE dis_3 (k:511)

(i) LE dis (k: 426) **(j)** LE dis_1 (k: 169) **(k)** LE dis_2 (k:189) **(l)** LE dis_3 (k: 285)

Figure 6.1: Visualization of two-dimensional Breast Cancer data generated by Isomap, t-SNE and LE using Euclidean Distance, dis_1, dis_2, and dis_3.

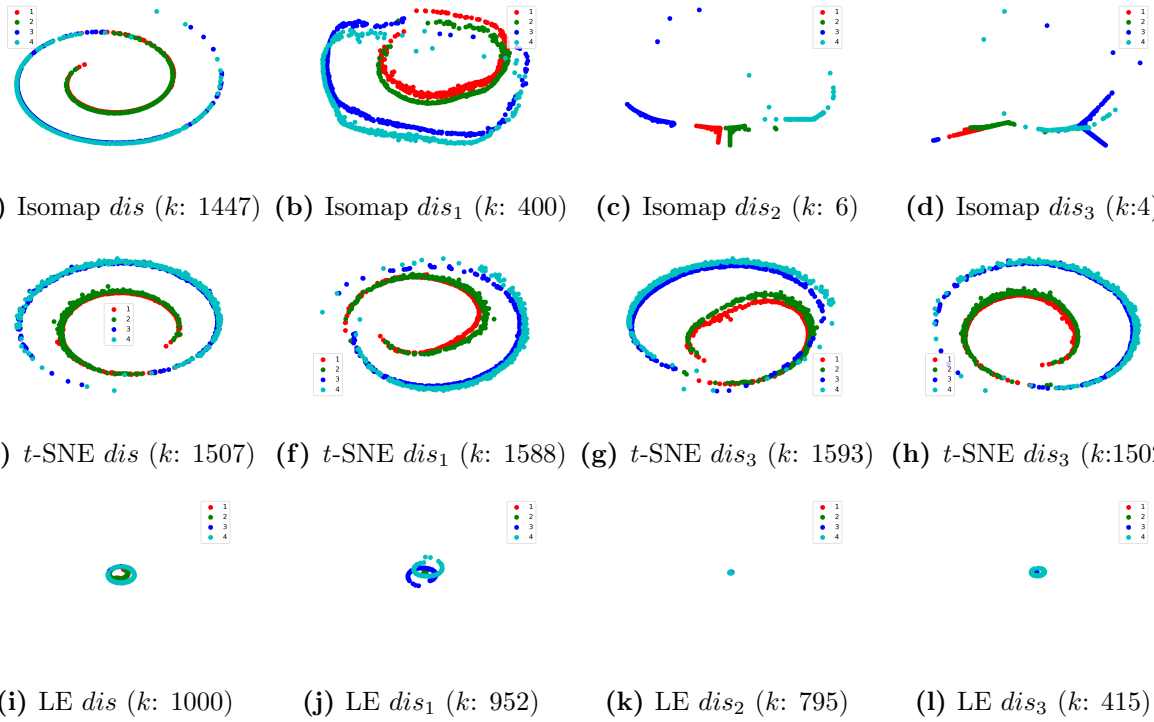


(a) Isomap dis (k: 515) **(b)** Isomap dis_1 (k: 385) **(c)** Isomap dis_2 (k: 4) **(d)** Isomap dis_3 (k:4)

(e) t-SNE dis (k: 501) **(f)** t-SNE dis_1 (k: 530) **(g)** t-SNE dis_2 (k: 518) **(h)** t-SNE dis_3 (k:511)

(i) LE dis (k: 426) **(j)** LE dis_1 (k: 169) **(k)** LE dis_2 (k:189) **(l)** LE dis_3 (k: 285)

Figure 6.2: Co-ranking matrixes of two-dimensional Breast Cancer data generated by Isomap, t-SNE and LE using Euclidean Distance, dis_1, dis_2, and dis_3.

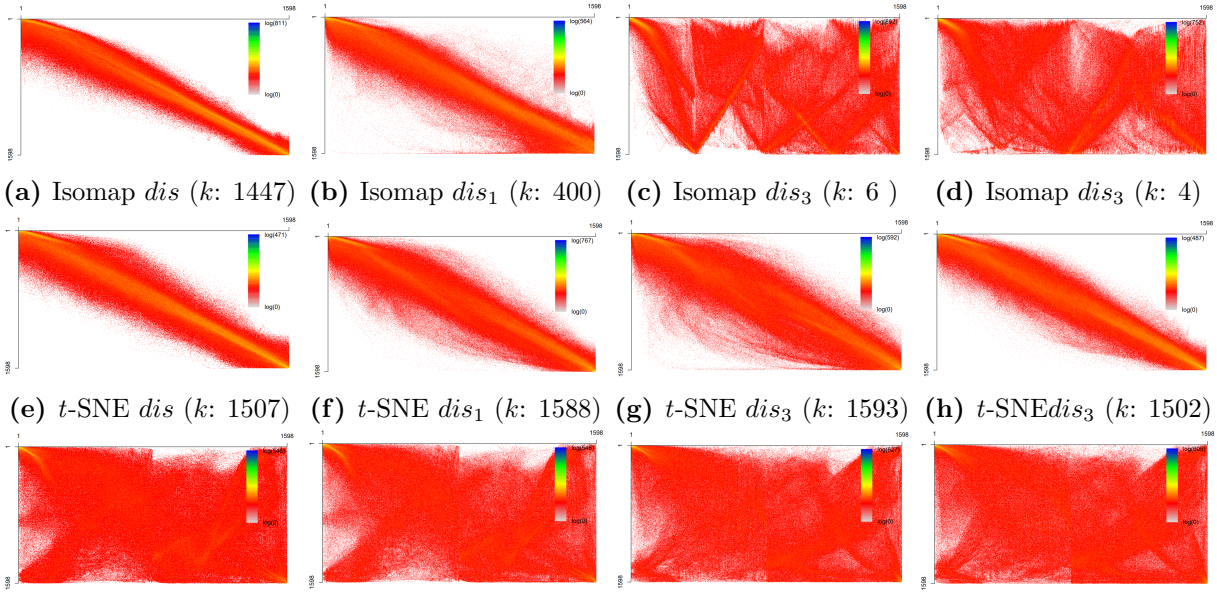


(a) Isomap dis (k: 1447) **(b)** Isomap dis_1 (k: 400) **(c)** Isomap dis_2 (k: 6) **(d)** Isomap dis_3 (k:4)

(e) t-SNE dis (k: 1507) **(f)** t-SNE dis_1 (k: 1588) **(g)** t-SNE dis_3 (k: 1593) **(h)** t-SNE dis_3 (k:1502)

(i) LE dis (k: 1000) **(j)** LE dis_1 (k: 952) **(k)** LE dis_2 (k: 795) **(l)** LE dis_3 (k: 415)

Figure 6.3: Visualization of two-dimensional Swiss Roll data generated by Isomap, t-SNE and LE using Euclidean Distance, dis_1, dis_2, and dis_3.



(a) Isomap dis (k: 1447) **(b)** Isomap dis_1 (k: 400) **(c)** Isomap dis_3 (k: 6) **(d)** Isomap dis_3 (k: 4)

(e) t-SNE dis (k: 1507) **(f)** t-SNE dis_1 (k: 1588) **(g)** t-SNE dis_3 (k: 1593) **(h)** t-SNEdis_3 (k: 1502)

(i) LE dis (k: 1000) **(j)** LE dis_1 (k: 952) **(k)** LE dis_2 (k: 795) **(l)** LE dis_3 (k: 415)

Figure 6.4: Co-ranking matrixes of two-dimensional Swiss Roll data generated by Isomap, t-SNE, and LE using Euclidean Distance, dis_1, dis_2, and dis_3.

6.2 Chapter Summary

The simulation results have confirmed the theoretical analyses: the dissimilarity measure does not improve the maintaining structure but destroys it. Both quality measures, Kendall's Tau and Co-ranking matrix, demonstrated that the three dimensionality reduction techniques, Isomap, t-SNE and LE that use Euclidean distances instead of dissimilarity measures produced more trustworthy visualisations, which is far from what many researchers have claimed in their study.

Chapter 7

Conclusion

7.1 Summary of Thesis Achievements

Several achievements towards trustworthiness and computational time of visualising high dimensional data have been made in this study. The main contributions of this research are demonstrated in the following.

An in-depth literature review has been made from where two research questions get answers as follows:

1. The similarity measure employed in a dimensionality reduction technique has a massive impact on the scale of maintained data structure.
2. Employing different similarity measures in high and low dimensional spaces data to produce tear and false neighbours problems.

Also, this research has made significant contributions and has filled the knowledge gaps by developing four dimensionality reduction techniques intending to improve the scale of the maintained data structure and reduce computational time, as mentioned below:

1. Developed a nonlinear dimensionality reduction technique named Same Degree Distribution (SDD) that captures the data structures better than other current dimensionality

reduction methods in less computational time. SDD employs degree-distribution, which is the same as Student-t with degree 1, and for higher degrees, it has longer tails than Student-t. By using the degree-distribution, which has longer tails than the Gaussian distribution, SDD ensures that it captures the global data structure better than other Gaussian-based methods. Also, SDD ensures that tears and false neighbours problems are prevented by using the same degree-distribution in high low dimensional spaces. Also, degree-distribution does not require tuning the number of neighbours, perplexity or other expensive parameters but instead, it requires tuning the degree, which usually ranges from 1 to 15.

SDD outperformed benchmarks methods such as t-SNE, UMAP, Isomap, PCA, MDS, Trimap, LLE, and LE in structure capturing in data that is dominated by small and medium distances.

2. Developed an extension of SDD named Multi Same Degree Distribution (MSDD) method to capture better the high dimensional space data structure than SDD. MSDD ensures that both the local and global structure of the data has been preserved by employing more than one degree-distribution and correspondingly more than one objective function that is optimised by a multi-objective optimisation method. However, from simulation results conducted, MSDD didn't show significant improvements in structure maintenance; on the other hand, the time complexity increases significantly.

3. Developed a parameter-free same degree-distribution (parameter-free SDD) dimensionality reduction method that captures the same scale of data structure with SDD but does not require tuning the degree of distribution or any other parameter that makes parameter-free SDD a significantly low costly method.

   The benefits of parameter-free SDD are by using degree-distribution ($deg = 1$) in high and low dimensional space but re-scaling the pairwise distances of original data in the interval [0, 2] instead of [0, 1]. The performance of parameter-free SDD has been demonstrated that it achieves the same performance in terms of structure maintenance but in significantly less time than SDD. In terms of structure maintenance, parameter-free SDD

outperforms all considered methods. And in terms of computational time, it outperforms t-SNE, UMAP, Trimap, Isomap, LE, LLE, and MDS. A theoretical proof also supports the excellent performance of parameter-free SDD.

4. Developed parametric SDD, which also addresses the problem of out-of-sample data samples. Parametric SDD proposes using Neural Networks to mimic the two-dimensional data produced by SDD. It has been demonstrated experimentally that parametric SDD maintains the training data structure (where the networks have been trained) and testing data (for unseen data for network).

Theoretical and practical analyses have been made to demonstrate the impact of dissimilarity measures on structure capturing and classification accuracy, which then generates two useful findings:

1. Supervised manifold learning can be used for classification purposes with the advantage of classification error reduction.

2. Supervised manifold learning/dimensionality reduction techniques can not be used for visualising high dimensional space data, as the class information involved in dissimilarity measures can destroy data structure capturing.

Dissimilarity measure forces relocating data samples using class information, but it does not improve data structure capturing. Following the theoretical analysis and supported by experimental results, we can conclude that the dissimilarity measure in Isomap, t-SNE, and LE worsens data structure capturing. Therefore, using Euclidean distance would be more beneficial than dissimilarity measures. However, dissimilarity measure dis_3 has a positive impact on t-SNE, which can help preserve global data information better. In addition, a dissimilarity measure can be usefully incorporated in manifold learning techniques to achieve a better RBF-based classifier, and the class-separation achieved by supervised dimensionality reduction methods can reduce the classification error.

7.2 Future Work

Further work can be done to improve the performances of SDD, MSDD, and parameter-free SDD in both capturing a better data structure and spending lower computational time and sources. SDD, MSDD, and parameter-free SDD have been demonstrated to outperform current dimensionality reduction methods in structure maintenance and computational time with data having a large fraction of short distances. However, in cases where the fraction of large distances are significantly higher than the fraction of short distances, the developed methods can not outperform other global methods like Isomap. The problems mentioned above of SDD, MSDD, and parameter-free SDD can be further addressed as follows:

1. Weighting differently the cost functions in the multi-objective optimisation technique, where the cost functions with huge sensitivity in large distances have a large influence than the cost functions with huge sensitivity in short distances, or
2. Employing another distribution on top of degree-distribution(s) that has a high sensitivity to large distances.

In addition, like t-SNE, in a very big dataset (with a large number of data samples), the three developed approaches, SDD, MSDD, and parameter-free SDD, consume high computational sources and time to calculate the pairwise distance matrix. Implementing these methods in a big dataset (with millions of data samples) faces memory issues for the capacity that a computer has, and as such, it requires additional memory sources. How to address this issue remains a further research question.

Bibliography

[1] J.R. Munkres. Topology: A First Course. Prentice-Hall, Englewood Cliffs, NJ, 1975.

[2] Izenman, A.J., 2008. Modern multivariate statistical techniques. Regression, classification and manifold learning, 10, pp.978-0.

[3] Lee, J.A. and Verleysen, M., 2007. Nonlinear dimensionality reduction. Springer Science Business Media.

[4] Li, Z., Xu, W., Huang, A. and Sarrafzadeh, M., 2012, May. Dimensionality reduction for anomaly detection in electrocardiography: A manifold approach. In 2012 Ninth International Conference on Wearable and Implantable Body Sensor Networks (pp. 161-165). IEEE.

[5] Blázquez-García, A., Conde, A., Mori, U. and Lozano, J.A., 2020. A review on outlier/anomaly detection in time series data. arXiv preprint arXiv:2002.04236.

[6] Fujiwara, T., Sakamoto, N., Nonaka, J., Yamamoto, K. and Ma, K.L., 2020. A visual analytics framework for reviewing multivariate time-series data with dimensionality reduction. IEEE transactions on visualization and computer graphics, 27(2), pp.1601-1611.

[7] Lin, Y., Zhu, X., Zheng, Z., Dou, Z. and Zhou, R., 2019. The individual identification method of wireless device based on dimensionality reduction and machine learning. The Journal of Supercomputing, 75(6), pp.3010-3027.

[8] Yuan, C., Sun, X. and Lv, R., 2016. Fingerprint liveness detection based on multi-scale LPQ and PCA. China Communications, 13(7), pp.60-65.

[9] Zhang, L., Yang, M., Feng, Z. and Zhang, D., 2010, August. On the dimensionality reduction for sparse representation based face recognition. In 2010 20th International Conference on Pattern Recognition (pp. 1237-1240). IEEE.

[10] Huang, W. and Yin, H., 2012. On nonlinear dimensionality reduction for face recognition. Image and Vision Computing, 30(4-5), pp.355-366.

[11] Li, W., Prasad, S., Fowler, J.E. and Bruce, L.M., 2011. Locality-preserving dimensionality reduction and classification for hyperspectral image analysis. IEEE Transactions on Geoscience and Remote Sensing, 50(4), pp.1185-1198.

[12] Alhayani, B. and Ilhan, H., 2017. Hyper spectral image classification using dimensionality reduction techniques. International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, 5(4), pp.71-74.

[13] Luo, F., Zhang, L., Du, B. and Zhang, L., 2020. Dimensionality reduction with enhanced hybrid-graph discriminant learning for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 58(8), pp.5336-5353.

[14] Kadhim, A.I., Cheah, Y.N. and Ahamed, N.H., 2014, December. Text document preprocessing and dimension reduction techniques for text document clustering. In 2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology (pp. 69-73). IEEE.

[15] Lacoste-Julien, S., Sha, F. and Jordan, M.I., 2008. DiscLDA: Discriminative learning for dimensionality reduction and classification. In Advances in neural information processing systems (pp. 897-904).

[16] Zahorian, S. and Hu, H., 2011. Nonlinear dimensionality reduction methods for use with automatic speech recognition. Speech Technologies, pp.55-78.

[17] Sharma, S., Kumar, M. and Das, P.K., 2015, May. A technique for dimension reduction of MFCC spectral features for speech recognition. In 2015 International Conference on Industrial Instrumentation and Control (ICIC) (pp. 99-104). IEEE.

[18] Belarbi, M.A., Mahmoudi, S. and Belalem, G., 2017. PCA as dimensionality reduction for large-scale image retrieval systems. International Journal of Ambient Computing and Intelligence (IJACI), 8(4), pp.45-58.

[19] Chamberland, M., Raven, E.P., Genc, S., Duffy, K., Descoteaux, M., Parker, G.D., Tax, C.M. and Jones, D.K., 2019. Dimensionality reduction of diffusion MRI measures for improved tractometry of the human brain. NeuroImage, 200, pp.89-100.

[20] Beyeler, M., Rounds, E.L., Carlson, K.D., Dutt, N. and Krichmar, J.L., 2019. Neural correlates of sparse coding and dimensionality reduction. PLoS computational biology, 15(6), p.e1006908.

[21] Ji, S., 2013. Computational genetic neuroanatomy of the developing mouse brain: dimensionality reduction, visualization, and clustering. BMC bioinformatics, 14(1), pp.1-14.

[22] Reddy, G.T., Reddy, M.P.K., Lakshmanna, K., Kaluri, R., Rajput, D.S., Srivastava, G. and Baker, T., 2020. Analysis of dimensionality reduction techniques on big data. IEEE Access, 8, pp.54776-54788.

[23] Tsai, F.S., 2011. Dimensionality reduction techniques for blog visualization. Expert Systems with Applications, 38(3), pp.2766-2773.

[24] Gutierrez, C.E., Alsharif, M.R., Cuiwei, H., Khosravy, M., Villa, R., Yamashita, K. and Miyagi, H., 2013. Uncover news dynamic by principal component analysis. ICIC Express Lett, 7(4), pp.1245-1250.

[25] Alkhayrat, M., Aljnidi, M. and Aljoumaa, K., 2020. A comparative dimensionality reduction study in telecom customer segmentation using deep learning and PCA. Journal of Big Data, 7(1), pp.1-23.

[26] Scott, D.W. and Thompson, J.R., 1983, March. Probability density estimation in higher dimensions. In Computer Science and Statistics: Proceedings of the fifteenth symposium on the interface (Vol. 528, pp. 173-179). North-Holland, Amsterdam.

[27] Francois, D. High-dimensional data analysis: optimal metrics and feature selection. PhD thesis, Universite catholique de Louvain, Departement d'Ing´enierie Math´ematique, Louvain-la-Neuve, Belgium, September 2006.

[28] Van der Maaten, L. and Hinton, G., 2008. Visualizing data using t-SNE. Journal of machine learning research, 9(11).

[29] Amid, E. and Warmuth, M.K., 2018. A more globally accurate dimensionality reduction method using triplets. arXiv preprint arXiv:1803.00854.

[30] McInnes, L., Healy, J. and Melville, J., 2018. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426.

[31] Lespinats, S., Verleysen, M., Giron, A. and Fertil, B., 2007. DD-HDS: A method for visualization and exploration of high-dimensional data. IEEE transactions on Neural Networks, 18(5), pp.1265-1279.

[32] Lespinats, S. and Aupetit, M., 2009. False neighbourhoods and tears are the main mapping defaults. How to avoid it? How to exhibit remaining ones?. Proceeding of Quality Issues, Measures of Interestingness and Evaluation of data mining models (QIMIE'09), p.55.

[33] Zhang, D., Zhao, Y. and Du, M., 2016. A novel supervised feature extraction algorithm: enhanced within-class linear discriminant analysis. International Journal of Computational Science and Engineering, 13(1), pp.13-23.

[34] Hotelling, H., 1933. Analysis of a complex of statistical variables into principal components. Journal of educational psychology, 24(6), p.417.

[35] Kruskal, J.B., 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika, 29(1), pp.1-27.

[36] Sammon, J.W., 1969. A nonlinear mapping for data structure analysis. IEEE Transactions on computers, 100(5), pp.401-409.

[37] Demartines, P. and Hérault, J., 1997. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. IEEE Transactions on neural networks, 8(1), pp.148-154.

[38] Bishop, C., Svensén, M. and Williams, C., 1996. GTM: A principled alternative to the self-organizing map. Advances in neural information processing systems, 9.

[39] Schölkopf, B., Smola, A. and Müller, K.R., 1998. Nonlinear component analysis as a kernel eigenvalue problem. Neural computation, 10(5), pp.1299-1319.

[40] Werbos, P., 1974. Beyond regression:" new tools for prediction and analysis in the behavioral sciences. Ph. D. dissertation, Harvard University.

[41] Tenenbaum, J.B., Silva, V.D. and Langford, J.C., 2000. A global geometric framework for nonlinear dimensionality reduction. science, 290(5500), pp.2319-2323.

[42] Roweis, S.T. and Saul, L.K., 2000. Nonlinear dimensionality reduction by locally linear embedding. science, 290(5500), pp.2323-2326.

[43] Belkin, M. and Niyogi, P., 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. Advances in neural information processing systems, 14.

[44] Donoho, D., 2003. Hessian eigenmaps: new tools for nonlinear dimensionality reduction. Proc. National Academy of Science, 100, pp.5591-5596.

[45] Scholz, M., Kaplan, F., Guy, C.L., Kopka, J. and Selbig, J., 2005. Non-linear PCA: a missing data approach. Bioinformatics, 21(20), pp.3887-3895.

[46] Zhang, Z. and Zha, H., 2004. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM journal on scientific computing, 26(1), pp.313-338.

[47] Weinberger, K.Q. and Saul, L.K., 2006. Unsupervised learning of image manifolds by semidefinite programming. International journal of computer vision, 70(1), pp.77-90.

[48] Coifman, R.R. and Lafon, S., 2006. Diffusion maps. Applied and computational harmonic analysis, 21(1), pp.5-30.

[49] Hinton, G.E. and Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. science, 313(5786), pp.504-507.

[50] Zhang, Z. and Wang, J., 2006. MLLE: Modified locally linear embedding using multiple weights. Advances in neural information processing systems, 19.

[51] Gashler, M., Ventura, D. and Martinez, T., 2007. Iterative non-linear dimensionality reduction with manifold sculpting. Advances in Neural Information Processing Systems, 20.

[52] Lespinats, S., Fertil, B., Villemain, P. and Hérault, J., 2009. RankVisu: Mapping from the neighborhood network. Neurocomputing, 72(13-15), pp.2964-2978.

[53] Rosman, G., Bronstein, M.M., Bronstein, A.M. and Kimmel, R., 2010. Nonlinear dimensionality reduction by topologically constrained isometric embedding. International Journal of Computer Vision, 89(1), pp.56-68.

[54] Pearson, K., 1901. LIII. On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin philosophical magazine and journal of science, 2(11), pp.559-572.

[55] Saul, L.K. and Roweis, S.T., 2003. Think globally, fit locally: unsupervised learning of low dimensional manifolds. Journal of machine learning research, 4(Jun), pp.119-155.

[56] Hinton, G.E. and Roweis, S., 2002. Stochastic neighbor embedding. Advances in neural information processing systems, 15.

[57] Zhou, Y. and Sharpee, T.O., 2018. Using global t-SNE to preserve inter-cluster data structure. bioRxiv, p.331611.

[58] Lee, J.A., Peluffo-Ordóñez, D.H. and Verleysen, M., 2015. Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. Neurocomputing, 169, pp.246-261.

[59] Crecchi, F., de Bodt, C., Verleysen, M., Lee, J.A. and Bacciu, D., 2020. Perplexity-free Parametric t-SNE. arXiv preprint arXiv:2010.01359.

[60] De Bodt, C., Mulders, D., Verleysen, M. and Lee, J.A., 2018. Perplexity-free t-SNE and twice Student tt-SNE. In ESANN.

[61] Hinton, G.E., Osindero, S. and Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. Neural computation, 18(7), pp.1527-1554.

[62] Bengio, Y., Lamblin, P., Popovici, D. and Larochelle, H., 2006. Greedy layer-wise training of deep networks. Advances in neural information processing systems, 19.

[63] R. Salakhutdinov and G. E. Hinton, Deep Boltzmann machines, 2009. in Proc. Int. Conf. Ar-tif. Intell. Statist, pp. 448–455.

[64] Salakhutdinov, R. and Larochelle, H., 2010, March. Efficient learning of deep Boltzmann machines. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 693-700). JMLR Workshop and Conference Proceedings.

[65] Vrábel, J., Pořízka, P. and Kaiser, J., 2020. Restricted Boltzmann Machine method for dimensionality reduction of large spectroscopic data. Spectrochimica Acta Part B: Atomic Spectroscopy, 167, p.105849.

[66] Chen, D., Lv, J. and Yi, Z., 2017. Graph regularized restricted Boltzmann machine. IEEE transactions on neural networks and learning systems, 29(6), pp.2651-2659.

[67] Kadoury, S., 2018. Manifold learning in medical imaging. In Manifolds II-Theory and Applications. IntechOpen.

[68] Seo, K., Pan, R., Lee, D., Thiyyagura, P., Chen, K. and Alzheimer's Disease Neuroimaging Initiative, 2019. Visualizing Alzheimer's disease progression in low dimensional manifolds. Heliyon, 5(8), p.e02216.

[69] Huang, Y., Kou, G. and Peng, Y., 2017. Nonlinear manifold learning for early warnings in financial markets. European Journal of Operational Research, 258(2), pp.692-702.

[70] Hajderanj, L., Weheliye, I. and Chen, D., 2019, April. A new supervised t-SNE with dissimilarity measure for effective data visualization and classification. In Proceedings of

the 2019 8th International Conference on Software and Information Engineering (pp. 232-236).

[71] Zhang, S.Q., 2009. Enhanced supervised locally linear embedding.Pattern Recognition Letters, 30(13), pp.1208-1218.

[72] Vlachos, M., Domeniconi, C., Gunopulos, D., Kollios, G. and Koudas, N., 2002, July. Non-linear dimensionality reduction techniques for classification and visualization. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 645-651).

[73] Ribeiro, B., Vieira, A. and das Neves, J. C., 2008. Supervised Isomap with dissimilarity measures in embedding learning, in: Iberoamerican Congress on Pattern Recognition, Springer, pp. 389-396.

[74] Yu, M., Zhang, S., Zhao, L. and Kuang, G., 2017, April. Deep supervised t-SNE for SAR target recognition. In Frontiers of Sensors Technologies (ICFST), 2017 2nd International Conference on (pp. 265-269). IEEE.

[75] Cheng, J., Liu, H., Wang, F., Li, H. and Zhu, C., 2015. Silhouette analysis for human action recognition based on supervised temporal t-SNE and incremental learning. Ieee transactions on image processing, 24(10), pp.3203-3217.

[76] Geng, X., Zhan, D.C. and Zhou, Z.H., 2005. Supervised nonlinear dimensionality reduction for visualization and classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 35(6), pp.1098-1107.

[77] Wei, C., Chen, J. and Song, Z., 2016. Developments of two supervised maximum variance unfolding algorithms for process classification. Chemometrics and Intelligent Laboratory Systems, 159, pp.31-44.

[78] De Ridder, D. and Duin, R.P., 2002. Locally linear embedding for classification. Pattern Recognition Group, Dept. of Imaging Science and Technology, Delft University of Technology, Delft, The Netherlands, Tech. Rep. PH-2002-01, pp.1-12.

[79] Sen, P.K., 1968. Estimates of the regression coefficient based on Kendall's tau. Journal of the American statistical association, 63(324), pp.1379-1389.

[80] Sammon, J.W., 1969. A nonlinear mapping for data structure analysis. IEEE Transactions on computers, 100(5), pp.401-409.

[81] Sidney, S., 1957. Nonparametric statistics for the behavioral sciences. The Journal of Nervous and Mental Disease, 125(3), p.497.

[82] Bauer, H.U. and Pawelzik, K.R., 1992. Quantifying the neighborhood preservation of self-organizing feature maps. IEEE Transactions on neural networks, 3(4), pp.570-579.

[83] Konig, A., 2000. Interactive visualization and analysis of hierarchical neural projections for data mining. IEEE transactions on neural networks, 11(3), pp.615-624.

[84] Venna, J. and Kaski, S., 2006. Local multidimensional scaling. Neural Networks, 19(6-7), pp.889-899.

[85] Chen, L. and Buja, A., 2009. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. Journal of the American Statistical Association, 104(485), pp.209-219.

[86] Lee, J.A. and Verleysen, M., 2008, April. Rank-based quality assessment of nonlinear dimensionality reduction. In ESANN (pp. 49-54).

[87] Lee, J.A. and Verleysen, M., 2009. Quality assessment of dimensionality reduction: Rank-based criteria. Neurocomputing, 72(7-9), pp.1431-1443.

[88] Banda, N. and Engelbrecht, A., 2017, November. Quality assessment of large scale dimensionality reduction methods. In 2017 IEEE 4th International Conference on Soft Computing & Machine Intelligence (ISCMI) (pp. 6-10). IEEE

[89] Colange, B., Vuillon, L., Lespinats, S. and Dutykh, D., 2019, October. Interpreting distortions in dimensionality reduction by superimposing neighbourhood graphs. In 2019 IEEE Visualization Conference (VIS) (pp. 211-215). IEEE.

[90] Chatzimparmpas, A., Martins, R.M. and Kerren, A., 2020. t-viSNE: Interactive Assessment and Interpretation of t-SNE Projections. IEEE transactions on visualization and computer graphics, 26(8), pp.2696-2714.

[91] Zhao, D., Wang, J., Lin, H., Chu, Y., Wang, Y., Zhang, Y. and Yang, Z., 2021. Sentence representation with manifold learning for biomedical texts. Knowledge-Based Systems, 218, p.106869.

[92] Hajderanj, L., Chen, D., Grisan, E. and Dudley, S., 2020. Single-and multi-distribution dimensionality reduction approaches for a better data structure capturing. IEEE Access, 8, pp.207141-207155.

[93] Miettinen, K., 2012. Nonlinear multiobjective optimization (Vol. 12). Springer Science & Business Media.

[94] Emmerich, M. and Deutz, A.H., 2018. A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Natural computing, 17(3), pp.585-609.

[95] Van Der Maaten, L., 2009, April. Learning a parametric embedding by preserving local structure. In Artificial Intelligence and Statistics (pp. 384-391). PMLR.

[96] Espadoto, M., Hirata, N.S.T. and Telea, A.C., 2020. Deep learning multidimensional projections. Information Visualization, 19(3), pp.247-269.

[97] Hajderanj, L., Chen, D. and Weheliye, I., 2021. The Impact of Supervised Manifold Learning on Structure Preserving and Classification Error: A Theoretical Study. IEEE Access, 9, pp.43909-43922.

[98] Balcan, M.F., Blum, A. and Srebro, N., 2008. A theory of learning with similarity functions. Machine Learning, 72(1), pp.89-112.