Non-Euclidean Dissimilarity Data in Pattern Recognition

Weiping Xu

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
February 2013

Department of Computer Science
University of York

Abstract

This thesis addresses problems in dissimilarity (proximity) learning, particularly focusing on identifying the sources and rectifying the non-Euclidean dissimilarity in pattern recognition. We aim to develop a framework for analyzing the non-Euclidean dissimilarity by combining the methods from differential geometry and manifold learning theory. The algorithms are applied to objects represented by the dissimilarity measures.

In Chapter 3 we describe how to reveal the origins of the non-Euclidean behaviors of the dissimilarity matrix for the purpose of rectifying the dissimilarities. We commence by developing a new measure which gauges the extent to which individual data give rises to departures from metricity in a set of dissimilarity data. This allows us to assess whether the non-Euclidean artifacts in a dataset can be attributed to individual objects or are distributed uniformly. The second novel contribution of Chapter 3 is to provide simple empirical tests that can be used to determine the sources of the negative dissimilarity eigenvalues. We consider three sources of the negative dissimilarity eigenvalues, namely a) that the data resides on a manifold, b) that the objects may be extended and c) that there is Gaussian noise. We experiment with the algorithms on a set of public dissimilarities used in various applications available from the EU SIMBAD project.

In Chapter 4, we propose a framework for rectifying the dissimilarities using Ricci flow on the manifolds so that the non-Euclidean artifacts are eliminated, as the second main contribution of this thesis. We consider the objects of interest to be represented by points on a manifold consisting of local patches with constant curvatures, and the given dissimilarities to be the geodesic distances on the manifold between these points. In differential geometry, Ricci flow changes the metric of a Riemannian manifold according to the curvature of the manifold. We seek to flatten the curved manifold so that a corrected set of Euclidean distances are obtained. We achieve this by deforming the manifold using

Ricci flow. In the first technique, we consider each edge as a local patch and apply Ricci flow independently to flatten each patch. In this way, the local structure of the manifold is ignored, as Ricci flow is applied independently on each edge. To overcome this problem, we propose a second technique, where add a curvature regularization process before evolving the manifold. Specifically we use the heat kernel to smooth out the curvatures on the edges. The results show both improved numerical stability and lower classification error in the embedded space.

To reduce the reliance on the piecewise embedding and its effects on individual edges, we extend the previous two techniques and develop a third means of correcting non-Euclidean dissimilarity data as the first contribution of Chapter 5. This is done by using a tangent space reprojection to inflate the local hyperspherical patches and align the local patches with the shortest edge-connected path. These three Ricci-flow-based techniques proposed through this thesis are investigated as a means of correcting the dissimilarities so that the the non-Euclidean artefacts are eliminated. We experiment on two datasets represented by dissimilarities, namely the CoilYork and the Chickenpieces datasets.

In the framework for correcting the non-Euclidean dissimilarities using the Ricci flow process, estimating the curvatures of the embedded manifold is an important component prior deforming the manifold. The second contribution of Chapter 5 is the investigation of the effects of the piecewise embedding methods (the kernel embedding and the Isomap embedding) on the curvatures computation and the introduction of a new way of computing the curvatures from a set of dissimilarities. We consider each local patch on a hypersphere, and deduce the enclosed volume of the points in terms of the curvature. We estimate the curvature by fitting the volume. We illustrate the utility of this method for estimating curvatures on the artificial dataset (2-sphere dataset).

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thanks my supervisors Prof. Edwin Hancock and Prof. Richard Wilson for their support and advice throughout my research. I would like to express my sincere appreciation for their valuable suggestions and time given to me for widening my view of my research. I also thank my assessor, Dr. Adrian Bors for his valuable suggestion and comments on my work. Many sincere thanks also go to the friends I made in York for their encouragement and interesting discussions. Thanks also to the SIMBAD for financial support under a PhD studentship. Finally, I would like to thank my family for their continuous support over the years.

Declaration

I declare that all the work in this thesis is solely my own, except where attributed and cited to other authors. Most of the material in this thesis has been previously published by the author. For a complete list of publications, please refer to the next page.

List of Publications

- Weiping Xu, Richard C. Wilson, Edwin R. Hancock: Determining the Cause of Negative Dissimilarity Eigenvalues, CAIP, 2011
- Weiping Xu, Edwin R. Hancock, Richard C. Wilson: Rectifying Non-euclidean Similarity Data through Tangent Space Reprojection, IbPRIA, 2011
- Weiping Xu, Edwin R. Hancock, Richard C. Wilson: Regularising the Ricci Flow Embedding, SSPR/SPR, 2010
- Weiping Xu, Edwin R. Hancock, Richard C. Wilson: Rectifying Non-Euclidean Similarity Data Using Ricci Flow Embedding, ICPR, 2010

Chapter 1

Introduction

1.1 The Problems

Every day we performs many patten recognition tasks, like recognising a face, identifying a friend's voice and reading characters on a newspaper. Pattern recognition is used to put raw objects to categories by looking for similar (or dissimilar) characteristics (patterns) [26, 5]. The two main data types of representation used in pattern recognition are vectorial data and pairwise dissimilarity (similarity) data. Vectorial data is feature based and has geometric meaning in which an object is viewed as a point in a Euclidean space [79, 26, 5]. Traditional machine learning methods are feature-based. Within the last fifty years, many powerful traditional pattern recognition techniques have been developed such as neural networks and support network machines. However, the vectorial representation limits the applicability of these feature-based methods. In many pattern recognition applications, it is difficult or impossible to extract efficient features [4], when objects have high dimensionality or structural properties or categorical features. Shapes, graphs, bags of words and survey results from psychology judgement are examples of such data. In these cases, it is often possible to define a dissimilarity measure (proximity) of objects for classification, in the same way that people are often able to tell the difference for two objects without being able to specify meaningful features. Moreover, these relational data are usually abstracted by graphs because graphs provide a powerful and natural way for capturing the relationship between objects that are not characterised by ordinal measurements or feature vectors [96, 102]. This reduces the notion of the dissimilarity of objects to the distances of graphs. This leads to the shift from feature-based classification to the dissimilarity (similarity)-based classification for solving pattern recognition problems.

The dissimilarity data describes the properties of objects in terms of their differences. It is relation based and thus more general than vectorial representation. However unlike the vectorial data, it is less developed, and limited numbers of practical classifiers have been developed for the dissimilarity data like the nearest neighbor classifier and indefinite kernel methods [47, 87]. From the dissimilarity data, we only know the distance measures of certain properties, and from these distance measures we are able to to build an isometric embedding of vectors so that the distance between vectors is the same as the given pairwise dissimilarity [43, 83]. However in many applications, the distance measures violate the restrictive conditions Euclidean space, and these distance measures are commonly used in computer vision and pattern recognition for their good performance which are derived from robust matching [8, 25, 93, 84]. Thus the dissimilarity data can not be transformed into a vector space by embedding without distortion. Such distances are non-Euclidean. The missing geometric meaning hinders the use of numerous powerful machine learning techniques such as Support Vector Machines and Neural Networks.

Although many researchers are aware of the importance of the dissimilarity-based approaches [114, 44, 60, 81, 40, 27, 31, 9, 80, 108, 41, 72, 83, 12], there is no systematic study of purely dissimilarity-based pattern analysis and recognition methods. The SIMBAD project named ”Beyond Features: Similarity-Based Pattern Analysis and Recognition” aims to analyse new concepts and propose approaches to deal with the challenging problems specific to dissimilarity-based representation, and present a comprehensive study of dissimilarity-based approaches for non-metric dissimilarities from the theoretical and applicative perspective. Specifically, the SIMBAD project focuses on deriving suitable dissimilarities from non-vectorial data and learning and classification for non-metric dissimilarities. Considering the learning and classification for non-metric dissimilarities, there are two different approaches: a)The first approach aims to find a vectorial representation from the dissimilarity data by performing some rectifications on the data . b) The second approach aims to develop new algorithms tailored for dissimilarity data without direct corrections. As part of the SIMBAD project, this thesis undertakes the research of dissimilarity-based recognition under the first approach and the goal in the technical annex is defined as:

"The basic assumption underlying the work within this workpackage is that similarity data is given, possibly in the form of a weighted graph, and we aim at developing algorithms for transforming them into instance-specific vectorial representations (embedding) that are suitable for traditional geometric learning algorithms." [82]

To learn the non-Euclidean dissimilarity data, the distance is usually transformed into a vectorial form by embedding, before the statistical learning techniques are applied on the data represented as feature vectors. The common embedding techniques for non-Euclidean pairwise data introduce various assumptions about the information and representation. One main assumption is that the Euclidean violations are an artifact of noise and do not carry valuable information about the dataset. Several authors have shown that the Euclidean violations can be beneficial to the classification results [85, 86, 72, 71]. Thus it is advisable to analyse the causes of the non-Euclidean violations before using embedding techniques to rectify the pairwise data. We will also explore how to perform suitable Euclidean corrections of non-Euclidean data so that the new Euclidean distances contains useful information hidden in the initial non-Euclidean violations. It is strongly linked to the differential geometry and manifold learning. This is achieved by considering non-Euclidean data to be represented as points on a curved manifold, modeling a manifold consisting of a set of local patches with individual constant sectional curvatures and evolving the local patches using Ricci flow to remove the curvature from the manifold.

1.2 Our Goals

The ultimate goal of this thesis is to use negative spectrum of the associated Gram matrix from a dissimilarity data for detecting possible sources of the non-Euclidean dissimilarity data in pattern recognition, and to use ideas from differential geometry and manifold learning to develop new embedding methods that can be used to analyse object and learn object class categories. Moreover, we explore different routes of embedding to preserve useful information for classification results which are usually ignored by the current embedding methods in the literature. To this end, we focus on:

- Identification of causes for non-Euclidean data: We find three measures to identify the possible causes of non-Euclidean violations. This is achieved by modeling the non-Euclidean pairwise data from the three cases: a simple manifold, extended objects and Gaussian noise. Then three measures are derived from observing the non-metricity and the shape of the spectrum of the negative eigenvalues of the Gram matrix.

- Piecewise Ricci flow embedding and regularised Ricci flow embedding: Here we consider each pair of points as an individual local patch with constant curvature, and the corresponding pairwise distance as the geodesic distances of the two points. We use two distance-preserving Euclidean embeddings of points (Isomap and the kernel embedding) and use the relationship between the geodesic distance on the manifold and the Euclidean distance in the embedded space to compute the curvature. We gradually move the curvatures of the local patches towards zero by performing the Ricci flow process. Finally we obtain a more Euclidean dissimilarity. In addition to the above method, we also explore the heat kernel diffusion to regularize the curvature and and obtain corrected data which preserve the local structure and give better classification performance.

- Tangent space reprojection: We explore whether tangent space representation and manifold embedding (the spherical embedding) can be combined with the Ricci flow to remove the dependence on the piecewise Euclidean embedding methods and their effects on individual edges. Here we consider a second order of k-NN graph as a local patch, consisting of the first and second neighbours of the reference object. The idea is to estimate the curvature by using the spherical embedding and gradually flatten the local patches with Ricci flow.

1.3 Thesis Layout

Having described the overall goals of the thesis, we proceed to give a brief introduction to the structure of the thesis. Chapter 2 reviews the literature of dissimilarity learning, looking particularly at the informativeness of non-Euclidean dissimilarities and the approaches to rectify non-Euclidean violations.

In Chapter 3, we present how three measures from the non-metricity and the negative eigenvalue spectrum can be used for characterising sources of non-Euclidean dissimilarity and to perform source detection to find suitable embedding methods for correcting the non-Euclidean violations. We will introduce a new measure to gauge the contribution of an individual data point to non-Euclidean violations. In addition to this, we will explore three measures extracted from the modeled sources and apply them on real world dissimilarity databases to investigate the sources of non-Euclidean artefacts.

In Chapter 4, piecewise Ricci flow embedding and regularised piecewise Ricci flow embedding are explored and applied on two public dissimilarity datasets. We use the embedding methods to map the non-Euclidean dissimilarity to Euclidean space. Once embedded in the Eucldiean space, we explore whether our embedded methods preserve the useful information contained in the original non-Euclidean violations by comparing our classification results with those obtained using original non-Euclidean dissimilarities and some alternative non-Euclidean distance rectification methods.

In Chapter 5, to further extend the two techniques in the previous Chapter, we combine the tangent space representation, the Ricci flow and the spherical embedding to develop a third method for rectifying non-Euclidean dissimilarity. To complement the framework of Ricci flow, we also explore the curvature of the embedded manifold of dissimilarities, an intrinsic geometric property of the embedded manifold and also an important component in our Ricci flow embedding framework. We commence this by applying the curvature estimation used in the piecewise Ricci flow on a 2D sphere dataset. It is observed that the Euclidean distances produced from the kernel embedding and Isomap embedding are not exactly the same as the initial configuration of Euclidean distance. This leads to distorted group structure of data from incorrect curvature estimation. Hence, we explore a method for computing curvature of the embedded manifold from dissimilarities, to remove the dependence on embedding techniques for the curvature estimation. We achieve this by curve fitting the volume of the manifold and the curvature of the manifold with the least-square method

Finally chapter 6 summarises the contribution of the thesis and identifies the limitations of the methods and areas for further work. The main findings of this thesis are that:

- The three measures for analysing and identifying the sources of negative dissimilarity eigenvalues are useful features for artificial dissimilarity data. They suggest possible sources for real dissimilarity data but not give complete conclusion in real dissimilarity data without ground truth.

- Our second contribution in this thesis is the introduction of a general framework for the computation of Ricci flow that can find Riemannian metrics (geodesic distances) by target curvatures. The aim of correcting non-Euclidean dissimilarities is to find Euclidean metrics from non-Euclidean dissimilarities. We introduce three techniques under this framework.

- Piecewise Ricci flow embedding can transform the non-Euclidean dissimilarity into Euclidean space but with some loss of discriminating power.

- In addition to eliminating the non-Euclidean artefacts of dissimilarity data, the regularised piecewise manifold embedding improves the classification performance by preserving the local structure in the data and the nearest neighbour relations remain constant or are improved.

- The tangent space reprojection can rectify the non-Euclidean dissimilarities to some extent because the edge-based approximation is not accurate for the distances between points on different patches and leads to more curved manifold after some iterations.

- The Euclidean embedding methods, the Isomap and kernel embedding, do not preserve exactly the local non-Euclidean distances. Thus the curvature estimation based on those is not accurate. We explore how to estimate the curvature from the volume but this method works well only on data with uniform distribution and high density. Given only a set of dissimilarity data, curvature estimation is very difficult.

Chapter 2

Literature Review

2.1 Introduction

A typical strategy for classifying dissimilarity-based data is to consider the given similarities as inner products in a Hilbert space or consider dissimilarities as distances in a Euclidean space [12]. The former leads to the use of kernel methods by altering the similarities and the latter leads to the use of embedding techniques for embedding dissimilarity data into vector space. We give a brief review of kernel methods, embedding procedures and pairwise clustering, which are the three major research activities involved in dissimilarity-based recognition.

Since our aim in this thesis is to develop embedding methods for correcting non-Euclidean artefacts by combining manifold learning and differential geometry, we focus on embedding methods. In this chapter, we start with the properties of dissimilarities, the distance measures and their applications in computer vision and pattern recognition in Section 2.3. We review the kernel methods in Section 2.4. This is followed in Section 2.5 by a survey of the state-of-art in embedding methods for two different objectives: a) The first category of embedding methods aims to find a low dimensionality for visualization; b) The second category aims to correct the non-Euclidean effects of the dissimilarity data.

Finally we review the pairwise clustering in Section 2.6.

Note: Many dissimilarity and similarity functions can be easily converted and substituted, hence we use the two terms interchangeably in this thesis.

2.2 Dissimilarity Representation

For dis(-similarity) data, an object is defined by its relationship to all other objects. The relationship is represented as the dissimilarity or similarity between each pair of objects. It can be viewed as a function defined on pair of objects. Each object is described by a vector of some continuous non-negative symmetric measure between the object and the rest.

$$d_{ij} \in R^+$$

for all $i = 1, 2, \cdots, j = 1, 2, \cdots$

$$s_{ij} \in R^+$$

for all $i = 1, 2, \cdots, j = 1, 2, \cdots$ where d_{ij} is a dissimilarity function, s_{ij} is a similarity function. Generally there is no requirements for a dissimilarity or a similarity. Considering the semantics of distances, a dissimilarity is usually assumed to be positive $d_{ij} \geq 0$ and $d_{ii} = 0$. In the context of embedding which produces a representation in terms of distances, a similarity matrix can be interpreted as inner product matric (covariance matrix) in vector space, which is required to be positive positive semi-definite, otherwise a pseudo-covariance matrix.

A dissimilarity (distance) is **metric** [90] when it satisfy following four properties:

1. non-negative, i.e. $d_{ij} \geq 0$ if object i is different from object j.

2. identity and uniqueness, i.e. $d_{ij} = 0$ if and only if the object i and j are identical.

3. symmetry, i.e. $d_{ij} = d_{ji}$, the distance from i to j is the same as the distance from j

to i.

4. triangle inequality, i.e. $d_{ij} \leq d_{ik} + d_{jk}$ for every k, the distance from i to j is always less to the sum of the distance from i to k and the distance from j to k.

As an inner product in vector space, a similarity must satisfy the following properties [90]:

1. non-negative, i.e. $s_{ij} \geq 0$ if object i is different from object j and $s_{ii} = 0$ only if object i is the origin.

2. symmetry, i.e. $s_{ij} = s_{ji}$, the similarity between i and j is the same as the similarity from j to i.

The thesis focus on the dissimilarity or similarity matrices. Similarity and dissimilarity can be transformed into each other via some function such as $D = 1 - S, d_{ij} = -log(s_{ij}), d_{ij} = \frac{1}{s_{ij}}$. We assume our pairwise data is given as dissimilarities.

Most dissimilarities satisfy the first two properties. The last two constraints are often violated. Take social science as an example, we can see the fact that person A likes person B does not mean person B likes person A, which is obvious not symmetric. Another fact is that A is B's friend, B is C's friend does not mean A is C's friend. Given a dissimilarity measure describing the friendship, we assume the lower the dissimilarity, the closer the friendship, hence dissimilarity measure describing the above friendship does not meet the triangle inequality.

Dissimilarity is obtained from direct comparison between different objects. There are many ways to compare two objects, hence there are many different distance measures. We introduce two standard dissimilarity data representations: matrix and graph.

$$\begin{bmatrix} 0 & 1.5 & 3.0 & 1.0 \\ 1.5 & 0 & 3.0 & 2.0 \\ 3.0 & 3.0 & 0 & 3.0 \\ 1.0 & 2.0 & 3.0 & 0 \end{bmatrix}$$

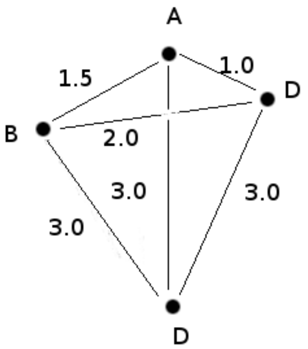


FIGURE 2.1: The matrix representation of pairwise dissimilarities on the left side and the corresponding graph representation on the right side.

In order to illustrate the two representation of pairwise dissimilarities, we consider a toy dataset given by 4×4 dissimilarities between four objects A through D in Figure 2.1. The left column gives a matrix representation of the pairwise data. The matrix structure stores a set of dissimilarities that are available for all pair of objects. It is square, and maybe symmetric. Each element $d(i, j)$ is the measured difference between object i and j, i.e. a distance measure. In general, the distance value is non-negative number. This value is zero in the case of the dissimilarity between an object and itself, hence the diagonal elements are usually zero. This value is small or close to zero when the pairwise objects are highly similar or close to each other and increase as the pairwise objects differ. This representation is natural and straightforward as the matrix reveals all the pairwise relations.

The right column of Figure 2.1 gives a weighted graph representation of the pairwise data. A weighted graph contains a set of vertices and a set of weighted edges between them. In a weighted graph, each edge has an associated numerical value, called edge weight. Usually, edge weights are non-negative. For pairwise data, the data objects are given by vertices and their corresponding dissimilarities are given by the weights of edges. In the case of symmetric distances, the graph is undirected. If all pairwise dissimilarities are known, the graph is fully connected. In our example, the vertices represent the objects A to D, the weighted edges represent the dissimilarities. Please note that the vertices in our example can be embedded in three-dimensional Euclidean space but not two-dimensional space without loss. Generally the vertices in graph representation do not correspond to points embedded in a two dimensional space.

Given the matrix representation of a pairwise dataset, its related graph structure can be obtained and vice versa.

In this thesis we discuss dissimilarity data given as a $n \times n$ dissimilarity matrix $D(R, R)$, where R is the set of n objects. Each element is a dissimilarity, measuring the difference between two objects. Usually dissimilarity is a function of observed objects based on some measurements. The dissimilarity measure is small for similar objects. Since many traditional geometric learning techniques work in metric or Euclidean space, it is necessary to review the definition of metric dissimilarity matrix and Euclidean dissimilarity matrix.

A distance matrix is said to be metric if its element d_{ij} satisfies the metric properties. When a dissimilarity matrix is obtained by transform from a similarity matrix via some decreasing function such as $D = 1 - S$, Being positive can be violated. Similar improper transformation can lead to the violation of identity and uniqueness so that there exist zero elements on the off-diagonal of the dissimilarity matrix or non-zero elements on the diagonal of the dissimilarity matrix. The violations can also be intrinsic from psychology. Symmetry can be violated via some asymmetic distance function and an asymmetric dissimilarity matrix are usually made symmetric by averaging. Triangle inequality is often violated.

In addition to the metric properties of a dissimilarity, being Euclidean is another important properties in embedding methods and machine learning. A dissimilarity matrix is said to be **Euclidean** if there exists vectors in n-dimensional Euclidean space $\mathbf{x}_i, \mathbf{x}_j \in R^n$ such that distance is faithfully preserved. That is, the pairwise input distance is equivalent to the Euclidean norm of pairwise vectors $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$, where $\| \;\|_2$ denotes the

Euclidean norm [71, 42].

A Euclidean space provides a connection between the traditional inner product and Euclidean distances, as it is both metric and inner product space. Hence it is possible to embed isometrically any Euclidean distance matrix into a finite Euclidean space. Most of the traditional machine learning techniques assumes objects exist in a Euclidean space. Each object is a vector of coordinates relative to the features. This is vectorial data, which is represented by a matrix $X \in R^{p \times N}$ where p is the dimension of features,N is the number of sample in the dataset the i^{th} column represents object $\mathbf{x}_i$. The problem is how to transform the general pairwise data to vectorial data so that more data analysis tools can apply on the given data. We start with exploring the relation between vectors in Euclidean space and Euclidean dissimilarity matrix. Based on the definitions of the Euclidean distance and the inner product, given n vectors $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$ in Euclidean space,

$$\begin{aligned} d^2(\mathbf{x_i}, \mathbf{x_j}) &= \langle \mathbf{x}_i - \mathbf{x}_j, \mathbf{x}_i - \mathbf{x}_j \rangle \\ &= \langle \mathbf{x}_i, \mathbf{x}_i \rangle + \langle \mathbf{x}_j, \mathbf{x}_j \rangle - 2\langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ &= d^2(\mathbf{x}_i, \mathbf{0}) + d^2(\mathbf{x}_j, \mathbf{0}) - 2\langle \mathbf{x}_i, \mathbf{x}_j \rangle \qquad (2.1) \end{aligned}$$

hence the inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ is

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle = -\frac{1}{2}[d^2(\mathbf{x}_i, \mathbf{x}_j) - d^2(\mathbf{x}_i, \mathbf{0}) - d^2(\mathbf{x}_j, \mathbf{0})] \qquad (2.2)$$

Let us move the origin $\mathbf{0}$ to the mean vector, the new vector $\mathbf{y}_i = \mathbf{x}_i - \bar{\mathbf{x}}, \bar{\mathbf{y}} = \mathbf{0}$, $d^2(\mathbf{y}_i, \mathbf{0}) = \mathbf{d^2(y_i - \bar{y})}$, where $\bar{\mathbf{x}}$ the mean of the configuration, the inner product in Equation 2.2 is:

$$\langle \mathbf{y}_i, \mathbf{y}_j \rangle = -\frac{1}{2}[d^2(\mathbf{y}_i, \mathbf{y}_j) - \frac{1}{n}\sum_{k=1}^{n} d^2(\mathbf{y}_i - \mathbf{y}_k) - \frac{1}{n}\sum_{k=1}^{n} d^2(\mathbf{y}_j - \mathbf{y}_k) + \frac{1}{n}\sum_{k=1}^{n}\sum_{p=1} d^2(\mathbf{y}_k, \mathbf{y}_p)]\,(2.3)$$

The matrix whose elements are the above inner products are called centralized Gram matrix $g_{ij} = \langle \mathbf{y}_i, \mathbf{y}_j \rangle$, $G = YY^T$. Let D^2 be the element-wise squared Euclidean distances, the Gram matrix can also be expressed as $G = -\frac{1}{2} J D^2 J$ where $J = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$. Thus given Euclidean dissimilarity matrix D, the vectorial representation Y can be determined by the eigendecomposition of the Gram matrix:

$$Y = \Lambda^{1/2}\Phi$$

where $G = \Phi\Lambda\Phi^T = YY^T, \Lambda$ is a diagonal matrix whose diagonal consists of nonnegative eigenvalues and ϕ is an orthogonal matrix of the corresponding eigenvectors. The Gram matrix is very important in this factorization. It is because the Gram matrix of a Euclidean distance matrix is positive semidefinite (that is, its factorized eigenvalues are non-negative), the vector configuration Y can be obtained such that the distances between vectors are exactly equal to the original distances in the distance matrix $d(\mathbf{y}_i, \mathbf{y}_j) = d_{ij}$ [42].

Please note any distance can be interpreted as a dissimilarity. A dissimilarity matrix is metric if it satisfied certain conditions described earlier. A dissimilarity matrix is Euclidean if it can be embedded into a Euclidean space without distance distortion. A non-Euclidean dissimilarity is not necessarily non-metric. A metric dissimilarity is not necessarily non-Euclidean. A non-Euclidean dissimilarity is non-metric.

2.3 Distance Measures

Dissimilarity plays an important role in pattern recognition, due to its ability to capture the relational and structural information in objects. Research in data in terms of dissimilarity in pattern recognition is critical for two reasons. Firstly, it can be used as a complement to the vectorial feature-based recognition. Secondly, it can be a natural bridge between structural and statistical pattern recognition [87]. More and more dissimilarity data are generated from various applications, in the meantime distance measures from structured data like graph and tree are very popular in pattern recognition. Euclidean distance is the most commonly used distance due to is simplicity, but Jacobs et al [60, 25] show dissimilarity measure is often non-metric when derived from graph matching for shapes and image objects.

One of the earliest graph distances is edit distance introduced by Sanfeliu and Fu [96]. The edit distance is defined as the minimum over all the costs of sequences of edit operations that will make the two graphs isomorphic with each other, given a set of basic edit operations on nodes and edges of a structure and the costs of corresponding edit operations. It provides a very effective way of measuring the similarity of relational structures. However, computing edit distance is NP-hard [113]. Hence various methods are proposed to approximate the edit distance and most of the approximated distance measure are non-metric. The other important approach on distance measure is based on the maximum common subgraph introduced by Bunke and Shearer and the resulting distance is non-metric[7]. It shows that the graph edit distance computation is equivalent to the maximum common subgraph problem. Many distance measures are its variants. Wallis [119] defines a distance measure based on the size of the minimum common supergraph. Hidovic and Pelillo [53] develops Bunke and Shear's distance measure on attributed graphs from the original undistributed graphs. Torsello et al [113] finds four metric but non-Euclidean tree distances based on the maximum common subtree.

Another important class of distance measures is distance between point sets. The well-known example is the general Hausdorff distance (HD) introduced by Huttenlocher, Klanderman and Rucklidge [68] to deal with outliers and occlusions [68]. It measures the degree of mismatch between two sets. As a result, it allows a portion of one image to be compared with another and saves the computing-costly process of finding correspondence between objects and is consequently, robust to noise. It has become a standard measure for comparing shapes and images [57, 106, 95, 62], although it is metric but non-Euclidean.

Dubuisson and Jain propose a modified Hausdorff distance (MHD) and show that the modified Hausdorff distance has more discriminatory power than the general Hausdorff, although the modified Hausdorff is non-metric [25].

2.4 Kernel Methods

Kernel methods, such as Gaussian processes [77], support vector machines (SVM) [10] and kernel PCA [78] have attracted much interest due to their computational efficiency and generalization performance for various application problems. These methods implicitly map data to a usually much higher dimensional, feature space where the inherent structure of the data is better captured. The common idea is to interpret the similarities between data points as the inner product in a Hilbert space (or some feature space) and apply any classification algorithm that only depends on inner-products.

Standard kernel methods treat the data points in a Euclidean space and choose proper kernel functions which are positive semidefinite (psd) [100]. In contrast, another class of kernel methods fit a probability distribution to data points after predicting the likelihood under different models. These kernel methods are called generative kernels, because these methods integrate the generative models into kernels. Existing generative kernels include the Fisher kernel [59], the Jensen-Shannor divergence kernel [20] and probability product kernels [61].

The need to handle non-psd (indefinite) similarities led to the development of indefinite kernels. One approach is to modify the original algorithm but still use non-psd similarities as kernels. Lin and Lin [74] alter the sequential minimal optimization (SMO) [89] and show that the solution converges to the original algorithm, although the solution is a stationary point instead of a global minimum. Haasdonk extends the traditional SVM to handle indefinite kernels and shows that his solution is equivalent to minimising the distance between the reduced convex hull in a pseudo-Euclidean space [46, 48]. Pekaska et al. extend a generalized nearest mean classifier, Fisher linear discriminant classifier, and SVM in both a pesudo-Euclidan space and a similarity space [84, 87]. Another approach to deal with indefinite kernels is to regularize the kernels to be psd with either spectrum clip, spectrum flip , or spectrum shift [125, 12, 93, 72, 134].

2.5 Embedding Methods

2.5.1 Multidimensional Scaling Methods

Any method that produces a vectorial representation of the data which is represented as a set of dissimilarities is called embedding. The embedding procedures aim to preserve the dissimilarity either locally or globally. Rather than a single procedure, Multidimensional scaling (MDS) [6] is a family of most popular embedding methods which construct a configuration of points in a metric space from information about interpoint distances. In general, there are two types of MDS: metric and non-metric. Metric MDS makes the assumption that the input data is numeric data, while the non-metric MDS [1] requires simply that the data be in the form of ranks. Faloutsos and Lin [36] introduce the fast MDS using a heuristic technique by generating a new dimension recursively at each step. See [111, 112, 19, 6, 35, 107] for different kinds of MDS.

As all nonlinear distance-preserving embedding methods can be considered as the extension from the classical MDS, the classical MDS is of particular importance. The classical MDS finds the low-dimensional representation by minimizing a cost function called stress which has a form:

$$\text{Stress}_{\text{MDS}} = \sum_{i,j=1}^{N} (s_{ij} - \langle \mathbf{y_i}, \mathbf{y_j} \rangle)^{\mathbf{2}} \tag{2.4}$$

where s_{ij} is the similarity (inner product) between ith and jth objects, $\mathbf{y}_i$ is the vector for ith object, $\mathbf{y}_j$ is the vector for jth object determined by MDS. It aims to preserve the pairwise inner product by minimizing the differences of inner product from the input data and the vectorial data. Since the Euclidean distance can be defined by the pairwise inner product (Gram matrix) from basic vector computation, the classical MDS preserves the interpoint distance if the input dissimilarity data is Euclidean.

Another embedding method which is closely related to MDS is Principal Component Analysis (PCA) [55, 66], which takes the coordinates as input data and produces vectorial representation in low dimensional space that preserves the variance of data points as measured in the high dimensional space. Moreover, PCA is equivalent to MDS when the input dissimilarity in MDS is Euclidean. Both MDS and PCA are spectral methods, since the core operation of these methods is based on eigenvalue decomposition of either the covariance matrix (PCA) or the Gram matrix (MDS). MDS and PCA are efficient techniques that can detect the linear subspace of the input data. However, these methods fail to detect the nonlinear low-dimensional subspace such as the curved manifold as they attempt to preserve global structure [110, 11].

2.5.2 Manifold Embedding

More recent approaches try to reduce dimensionality that minimizes the distortion by inferring a low dimensional manifold where data resides. MDS and PCA are suitable when the low-dimensional manifold is embedded linearly in the ambient space. Recently, a number of spectral embedding methods motivated by graph theory have been developed to deal with general non-linear manifolds. They usually approximate the geodesic distance on the manifold by using graph edge distance. A well-known embedding method of this type is Isomap [110] which is closely related to the classical MDS. The only difference between these two methods is the input distances. Isomap uses graph distances instead of input-space distances as in classical MDS. It uses a neighborhood ball to build a graph where the graph distance is the shortest edge-connected distance between nodes. In this case, the short distances are preserved and the large distances are found by hops along short distances. The use of graph distance changes the linear classical MDS to a nonlinear algorithm which has been demonstrated to locate curved manifolds for a number of data sets.

Related algorithms include locally linear embedding [94], Laplacian eigenmap [3], Hessian LLE [24], local tangent space alignment (LTSA) [135], maximum variance unfolding (MVU)[120], and the diffusion map [70]. LLE is also a variant of PCA and preserves local structure by using linear coefficients to represent a data point by its neighbor points with coefficients and then attempting to preserve coefficients from the higher-dimensional data space to the low dimensional manifold. LLE consists of two steps: finding coordinates for data points on each local patch and then deriving the global coordinates with the alignment of local patches by solving an elegant eigenproblem. LTSA is a modifications of LLE and the only difference of these two methods is local geometry. LTSA uses the local tangent plane as the local patch to be aligned rather than the neighbourhood ball directly from the input data points. Laplacian eigenmaps also attempt to preserve certain local geometric structure of the data by building an adjacency weight matrix from the data points and projecting the data onto the largest eigenvector of the resulting Laplacian matrix. Hessian LLE finds a low-dimensional configuration of points by using the estimated Hessian over a neighborhood as the Laplacian matrix. The diffusion map is a variant of the Laplacian eigenmaps and constructs the Laplacian matrix by using a kernel function. MVU constructs a neighborhood graph as the local geometry and “unfolds” the local geometry by using the semifdefinite programming instead of learning the output vectors directly as the LTSA. The use of semidefinite prgramming makes LTSA preserve local distances and maximize the pairwise distances between any two points that are not connected in the local neighborhood graph. As stated in [99], all these methods share a similar structure of establishing a neighborhood graph on the data points and using eigendecomposition to produce vectorial representation in a low-dimensional space from the input data in terms of dissimilarity matrix. These methods are known as manifold learning.

A systematic comparison and identification of the advantages and shortcomings of these methods is given in [99, 115]. By associating each of these techniques as an instance of kernel methods, a kernel interpretation is given by Ham et al [49]. Although they are all variants of the classical methods of PCA and MDS and share a common aim to capture local structure of the data, they are based on rather different geometric intuition and intermediate computations. For example, geometrically, LLE and Laplacian eigenmaps attempt to preserve certain local relationship of the input data, while MVU trys to preserve the local distances but maximize the total variance of data points. Computationally, Isomap and MVU build a Gram matrix and derive the low-dimensional configuration with its largest eigenvectors, while LLE, Laplacian eigenmaps and Hessian LLE build a Laplacian matrix and use its smallest eigenvectors. Moreover, for some data, the intrinsic dimension can be estimated by a noticeable gap between a few largest eigenvalues and the rest of the spectra for methods using the Gram matrix, while detecting the intrinsic dimensionality needs an additional step for the methods using the Laplacian matrix [16].

The target embedding space in manifold learning is Euclidean vector space. This is plausible if the manifold on which the input data points reside is developable, i.e. can be flattened onto a plane without distance distortion. In the case of undevelopable manifolds, the non-Euclidean distances are intrinsically unable to be represented with Euclidean distances. This limitation motivates an alternative embedding which uses a non-Euclidean, but metric embedding space [75, 18, 103, 56, 92, 123]. The geodesic distances on a curved Riemannian manifold is a natural representation for non-Euclidean dissimilarities, as they are metric but non-Euclidean. The earliest form of such embedding is due to Lindman and Caelli [75]. They restrict their work to manifolds with constant curvature and have investigated the elliptic and hyperbolic embedding on psychological data. They have defined the Euclidean distances in the ambient space in terms of the pairwise geodesic distances on a manifold with constant curvature and the curvature of the manifold. Cox and Cox

[18] extend MDS to find a configuration of data points on a spherical space by optimizing the stress on geodesic distances. Shavitt and Tankel have studied hyperbolic embeddings in the context of interpreting internet connectivity [103]. Hubert et al explored unidimensional scaling constrained to circles [56]. Robles-Kelly and Hancock embed the graph which represents the dissimilarity data in a manifold and deform the graph so that the data points conform either to elliptic or hyperbolic geometry [92]. Recently Wilson et al projected data points in terms of non-Euclidean dissimilarity onto a constant curvature manifold (a hypersphere) by solving an optimization problem for finding the smallest eigenvalues of the associated Gram matrix from the input data points [123].

Above all, all these nonlinear techniques of manifold learning preserves proximity relations better than the linear ones at increased computational cost. Proximity relations include local structure and global structure of input data. The ideal embedding maps nearby input datapoints to nearby output datapoints, and faraway input datapoints to faraway output datapoints. In addition to this, most of these methods do not explicitly consider the intrinsic structure of the manifold on which the object possibly resides. One extension of manifold learning is to embed a graph on a manifold. The graph can be characterized by geometric invariants and be deformed by applying geometric methods like altering the curvature of the manifold. Various robust invariants including sectional curvature on edges, Gaussian curvature on faces have been explored in the graph embedding community for characterizing graphs and their effectiveness of these geometric invariants are demonstrated by good clustering results for a number of image data sets [92, 33, 127]. Robles-Kelly and Hancock have explored the link between the edge curvature and the embedding coordinates of nodes and estimate the curvature of edges using the difference between geodesic and Euclidean distances. As an extension of the edge curvature, the Gaussian curvature associated with triangular faces of the graph is computed by applying the Gauss-Bonnet theorem [34, 33].

2.5.3 Imposing Geometricity

The emphasis of the above embedding methods is on finding a low-dimensional representation. As many traditional learning methods are designed in a Euclidean space, it is important to check whether a given distance measure has a Euclidean behavior. For Euclidean distance, an exact distance-preserving configuration in Euclidean space can be found by embedding. The problem with dissimilarity representations and their embeddings is that the distance measures cannot be used to construct a Euclidean vector space if the underlying Gram matrix contains negative eigenvalues. If this is the case, then the data can not be embedded into a real-valued Euclidean space without distance distortion, and must instead be embedded into a complex pseudo-Euclidean space called Krein space [40] or a postulated Euclidean space called dissimilarity space [84] for preserving distances. This section starts with these two general dissimilarity representation spaces, and moves onto embedding approaches as a means for correcting non-Euclidean behaviors.

When the Gram matrix (inner product) has negative eigenvalues then those dimensions of the embedding associated with negative eigenvalues are represented by imaginary numbers, and those associated with positive eigenvalues by real numbers. In other words, the data are embedded into a pseudo Euclidean space. Introduced by Goldfarb, a pseudo-Euclidean spaces [47, 87] is an indefinite inner product space which is flexible enough to allow embedding for any symmetric distance matrix. The pseudo-Euclidean space $R^{(p,q)}$ is composed of two Euclidean spaces, for which the inner product operation is positive definite on the first subspace R^p of dimension p and negative definite on the second one R^q of dimension q. The pseudo-Euclidean space is a Euclidean space when $q = 0$. The squared distances in pseudo-Euclidean space are interpreted as the difference of squared Euclidean distances from the "positive" subspace and the "negative" subspace $d^2_{R^{(p,q)}}(\mathbf{x}, \mathbf{y}) = d^2(\mathbf{x}_p, \mathbf{y}_p) - d^2(\mathbf{x}_q, \mathbf{y}_q)$, which can be negative. The pseudo-Euclidean space is completely isometric to the original distances as the distances between the embedded vectors are equal to the original ones. However, the pseudo-Euclidean space is nonmetric and points in the space can have negative distances to each other [123]. As a result, the traditional geometric learning techniques which are based on Euclidean space are not able to be used directly on data embedded in this space.

Pekalska and Duin introduce the concept of dissimilarity space, where features are dissimilarities to a representation set of objects (prototypes) [86]. In the dissimilarity space, a set of r prototypes $R = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_r\}$ covering all classes is selected from N objects, and the data is represented as a $n \times r$ distance matrix $D(T, R)$. Each object is described by a vector of some distance measure between the object and all the objects in the representation set. The dissimilarities to the selected prototypes work like features, thus any traditional classifier operating on feature-based data can be used. However this representation may lose information encoded in the original pairwise dissimilarity data.

Unlike vectorial data, learning from the dissimilarity data usually uses either the k-nearest neighbor classifiers (KNN) or kernel methods [87, 84]. KNN classifier puts a new object to the category of the nearest representation objects [17]. It is one of the few data analysis tools which requires only distance information and can be directly used for general dissimilarity data, either metric, Euclidean, non-metric or non-Euclidean. KNN generalizes well for dissimilarity data. However, in addition to being sensitive to noise examples, KNN needs high storage and high computational cost for new objects.

Another typical approach is to correct the distance matrix D so that the corresponding Gram matrix becomes psd. This correction procedure is equivalent to the kernel regularization methods for obtaining definite kernels from indefinite kernel when treating similarities (Gram matrix) as kernels in the kernel methods. The kernel embedding (or the positive part of the pseudo Euclidean space (PPES))[30] only considers the positive eigenvalues of the Gram matrix by ignoring the negative eigenvalues. Distances are overestimated compared to the original distances. Since the distances are positive, the larger the magnitude of the negative eigenvalues, the larger the resulting distances differ from the original ones. The resulting configuration is in the positive p dimension subspace. The embedding is reasonable on the basis that the sum of the positive eigenvalues is larger than the sum in magnitude of the negative ones or the negative eigenvalues result from the noise and can be disregarded. Hence some important information is possibly lost when the negative eigenvalues are relatively large. This correction technique is equivalent to spectrum clipping in the kernel methods.

The kernel embedding neglects the Euclidean violation (the negative eigenvalues of the Gram matrix), thus the information coded in the negative part of the pseudo Euclidean space is lost. In order to recover the information coded in the Euclidean violation, the Associated Euclidean space embedding (AES) [85, 29] includes the negative eigenvalues by using the magnitudes of the eigenvalues. The configuration of data points uses the positive eigenvalues and the absolute value of the negative eigenvalues. In this way, features hidden in the negative eigenvalues are preserved. In a pseudo Euclidean space, the distance is interpreted as the difference between squared Euclidean distances from the positive and negative subspace. Here the distance is regarded as the sum of squared Euclidean distances from the positive and negative subspace. However, the Euclidean distances from the configuration are usually highly distorted and highly over-estimated compared to the original distances. It might be reasonable to use this method if the distance measure is negative or the "negative" subspace contributes much more than the positive ones to distances and contains very useful information for classification. Otherwise, the configuration will obtain very bad results for locally sensitive classifiers as the local ranking order in the original dissimilarity is destroyed. If all eigenvalues are positive, the associated Euclidean space embedding obtains the same configuration as the kernel embedding. This correction technique is equivalent to spectrum flipping in the kernel methods

The constant shift embedding [72, 71] makes the Gram matrix psd by adding a suitable constant $c \geq -2\lambda_{\text{min}}$ to the off-diagonal elements of the squared dissimilarity matrix, where λ_{min} is the minimal eigenvalue of the gram matrix. It is equivalent to adding $|\lambda_{\text{min}}|$ to the non-zero eigenvalues. Since the square root is monotonically increasing, the object with smaller distance in the original dissimilarity matrix is still smaller in the resulting dissimilarity matrix. Compared kernel embedding and AES, the constant shift embedding does not change the order of similarities between any two different objects. Laub et al. [72] have studied the constant shift embedding for pairwise clustering on non-metric dissimilarity data and show that this method preserves the group structure of data. The constant shift embedding is distortion-free in terms of the data partition. Compared with the original distances, the corrected distances are over-estimated. The distortion is large for non-Euclidean dissimilarity data of which the negative eigenvalue has a big mangitude. This correction technique is equivalent to the spectrum shift in the kernel methods.

Duin et al. [29] identify many causes of non-Euclidean dissimilarity and conclude the non-Euclidean dissimilarity is either caused by measurement error, or the distance measures from the demand in applications are intrinsically non-Euclidean when any pairwise comparison is in different feature space. It demonstrates the non-Euclideanness of the dissimilarity data is informative by comparing classification errors of the linear SVM for the original non-Euclidean dissimilarity data and the resulting Euclidean data from the kernel embedding. However, the effects of the causes of non-Euclideanness on the correcting procedures are not explored. Pekalska et al. [30, 85] demonstrate that the discriminating power of the corrected measure is not as good as the original non-Euclidean distance measures by testing the above correcting approaches on five dissimilarity data sets with four classifiers. These research results put the necessity of imposing geometricity into doubt and emphasizes that the discriminating power of original dissimilarity measures are more important than the Euclidean property [29, 30, 85]. Hence, how to correct the dissimilarity such that the new dissimilarity in Euclidean space is not less discriminative than the non-Euclidean dissimilarity is a problem.

2.6 Pairwise Clustering

Given datapoints and a pairwise measure of distance between datapoints, pairwise clustering usually represent datasets with distances defined by a weighted undirected graph whose vertices correspond to datapoints and weights of edges between nearby points correspond to distance. A dissimilarity matrix whose elements correspond to distances between datapoints is usually built to represent such dataset. Pairwise clustering is to divide given datapoints in terms of a dissimilarity matrix to disjoint groups (clusters) such that similar datapoints are grouped together and dissimilar datapoints are in different groups. It is used to find the hidden structure of a dataset. A huge variety of different algorithms for clustering has been motivated by combinatorial search techniques, kernel techniques and spectral graph theory. Combinatorial search techniques-based clustering algorithms model the clustering problem as an optimization problem that optimize some criterion function by organizing the data points into certain groups. Herault and Horaud's figure-ground discrimination model [52], Hofmann and Buhmann's deterministic annealing algorithms [54] and Fisherer and Buhamann's path based approach [38] share this idea.

Since the general dissimilarities can be nonmetric or non-Euclidean, a loss-free embedding into a vector space is not possible. Most of the clustering algorithms are based on the Euclidean distances, hence unable to deal with non-Euclidean dissimilarity. The well-known k-means clustering which uses k prototype vectors is one of these techniques. Kernel-based clustering methods are developed to solve this limitation by transforming nonlinearly separable structure in the data into a higher-dimensional feature space where it is possible to sperate the data linearly. Well-known manifestations of this idea include kernel k-means [101, 13], NMF [130], kernel SOM [58] and kernel neural gas [91]. Kernel-based methods have many advantages [129]. For instance, it is more easily possible to obtain a linearly separable hyperplane in the high-dimensional or even infinite feature space. They are robust to noise and outliers. However, the computational complexity is high for large datasets.

Recently, spectral clustering methods have been very popular [2, 104, 116, 23, 22, 117, 118, 131]. They are based on spectral graph theory [15] and transform the clustering problem to a graph cut problem where a proper objective function has to be optimized. The idea is to build a weighted graph from the data in terms of a dissimilarity matrix and cut the graph with the leading vector from singular value decomposition of the Laplacian matrix of the weighted graph. Shi and Malik show the Fiedler eigenvector from the Laplacian matrix provides an approximate solution to minimize a normalized cut cost function [76, 104]. A comparison of some spectral methods is provided in [37, 116].

2.7 Summary

We may draw several conclusions based on the review of the related literature. Firstly, popular distances measures in computer vision and pattern recognition are mostly non-metric or non-Euclidean. The non-Euclidean behaviors of dissimilarity hinders the applicability of standard machine learning techniques and the optimization cost is high for indefinite kernel methods. Hence, more correction methods for eliminating the non-Euclidean artefacts are required so that many powerful traditional learning methods can be used to better analyse dissimilarity data. Moreover, informational content of non-Euclideanness is lost during the recently developed correction procedures, since the resulting Euclidean dissimilarity has less discriminative power than the original non-Euclidean data which is demonstrated by obtaining worse classification results from the resulting Euclidean data. In this thesis, we will not only propose new correcting procedures, but also try to preserve the underlying informational content in the ”negative” subspace.

Secondly, most Euclidean correcting procedures are plausible if the informational content of non-Euclideanness is limited or caused by accident. This trivial informational content is represented either as a small magnitude of smallest negative eigenvalues or no worse classification results after correction. Although numerous causes of non-Euclidean data are proposed, it lacks the mathematical formalization of these causes. These causes have not been thoroughly investigated or applied for the purpose of choosing proper correcting procedures. In this thesis, we will model the origination of non-Euclidean data and investigate the feasibility of extracting useful information from negative dissimilarity eigenvalues to identify the origination. Then based on the identified cause, we choose suitable correcting procedures. We will demonstrate a simple empirical test for identifying causes of non-Euclidean data.

Fourthly, Isomap and kernel embedding are two classical distance-preserving embeddings. The sectional curvature of a manifold with constant curvature has been defined in terms of geodesic distances on the manifold and Euclidean distances in the ambient space. By considering the objects of interest to be represented by points on a manifold with constant curvature in an ambient Euclidean space, and the given dissimilarities to be the geodesic distances on the manifold between these points, the Euclidean distances in the ambient space deduced from the Isomap and the kernel embedding can be used for the purpose of estimating curvatures of each edge in the curved manifold. On the other hand, the spherical embedding used in the manifold learning can be used for the purpose of estimating curvatures of local patches on a hypersphere.

Above all, the work in this thesis addresses the open problems in the research literature and aim to investigate the causes of non-Euclidean dissimilarity data and present effective embedding methods for transforming non-Euclidean dissimilarities into a set of Euclidean distances. In the subsequent chapters, we will compare our proposed methods with the state of art by using 1-nearest neighbour (1NN) classifier which is commonly used for non-Euclidean dissimilarity data and discuss in details our contribution to the dissmilarity-based learning.

Chapter 3

Determining the Causes of Negative Dissimilarity Eigenvalues

3.1 Introduction

Pairwise dissimilarity representations offer a powerful alternative to vectorial or feature-based characterisations of objects. Specifically, they provide a natural way of capturing the relationships between objects that are not characterised by ordinal measurements or feature vectors [87]. Given a symmetric pairwise dissimilarity matrix D, its similarity (Gram) matrix is defined as $-\frac{1}{2}D^2$ and the centered Gram matrix

$$G = -\frac{1}{2}JD^2J$$

where $J = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$is the centering matrix and **1** is the all-ones vector of length N. With the eigendecomposition of the Gram matrix $G = \Phi\Lambda\Phi^T$, a symmetric dissimilarity matrix is Euclidean if all eigenvalues of the corresponding Gram matrix are non-negative. One way to translate such data into a vector representation is to represent the similarity data using a kernel matrix, and to embed the data into a vector space using kernel principal component analysis. That is, the the eigendecomposition of the Gram matrix $G = \Phi\Lambda\Phi^T$, where Λ is Gram matrix's eigenvalues and Φ is the corresponding eigenvectors. In this way a vector representation is obtained by projecting the dissimilarity data into a vector space of fixed dimension.

However, one of the problems with dissimilarity representations and their embeddings is that the distance measures can not be used to construct a Euclidean vector space if the underlying Gram matrix contains any negative eigenvalue. If this is the case, then the data can not be embedded into a real-valued Euclidean space, and must instead be embedded into a complex valued or pseudo-Euclidean space [40]. Though projecting non-Euclidean dissimilarity into a pseudo-Euclidean space reproduces exactly the original distances without distortion, the pseudo-Euclidean embedding space is non-metric and thus the distances between some pairs of points in the space is negative [123]. As a result, the traditional geometric learning techniques can not be applied to this embedding data.

In order to analyse non-Euclidean dissimilarity data by means of traditional geometric machine learning or pattern recognition techniques based on vector space, we must first attempt to rectify the data for eliminating the non-Euclidean artefacts. Prior to these analysis, it is advisable to assess the degree and extent to which non-Euclidean artefacts affect the data-set. One measure that has proved useful in this respect is the negative eigenfraction [86] which is the total mass of negative eigenvalues as a fraction of the total mass of unsigned eigenvalues. However, in this chapter, we introduce a finer measure that assesses the contribution of each object to the mass of negative eigenvalues. In this way it is possible to determine whether the non-Euclidean artefacts are attributable to the outlying dissimilarities of a few objects or are uniformly distributed throughout the dataset.

One way of translating similarities into vector representation is based on the positive definite subspace of the distance embedding and manifold embedding (e.g. the spherical embedding in [123]. Each of these approaches is based on assumptions concerning the sources of the negative eigenvalues. The positive definite subspace embedding assumes that metric violations are an artefact produced by noise and that the “distances” in the negative sub-space do not carry any significant discriminative information. The manifold embedding assumes that the Euclidean violations are geodesic and that the data lies on a manifold. Recent studies [2,4] have showed that the negative eigenspace can provide with valuable information. Moreover, Euclidean correction can lead to poor classification performance.

Thus, before using any of the above approaches to attempt to rectify non-Euclidean data, the underlying causes should be analysed. The aim in this chapter is to study the occurrence of negative eigenvalues and investigate whether we can extract useful measures from the non-Euclidean dissimilarity matrix as a means of detecting the sources of negative eigenvalues. Models are developed to explain these spectra.

The outline of this chapter is as follows. We first model the distribution of non-Euclidean pairwise data in the following three situations: a) the objects lie on the surface of a sphere (a simple manifold) and that the pairwise similarities are geodesic distances across the manifold, b) extended objects where a non-metric dataset based on the distances between the surfaces of randomly positioned balls having different radii (Delfts balls data) and c) a noisy dataset with Gaussian noise added to the distance between points in Euclidean space. Then we explore whether the spectrum of the the negative eigenvalues of the Gram matrices can be used for the purpose of characterising these three sources. We examine the shape of the negative spectrum of non-Euclidean dissimilarity matrices under these three sources and find out that all the negative spectra follow an exponential decay but exhibit the tails differently, which is indicated by the slope and intercept of an exponential fitting. Moreover the dissimilarity value over the sphere is metric. Thus the three sources can be characterized by the non-metricity, the slope and incept extracted from an exponential fit. Our idea is to use these three measures as features for detecting the sources of negative eigenvalues.

For our experiments, we use the above models to analyze a set of public domain dissimilarity data provide by the EU SIMBAD project consortium. Our analysis provides insight into the non-Euclidean behaviour of dissimilarity datasets and can be used to select appropriate embedding methods suitable for the non-Euclidean data in hand.

3.2 Characterizing Non-metric or Non-Euclidean Data

In this thesis we are concerned with embedding data represented in terms of pairwise dissimilarities or distances in the area of pattern recognition, and in particular in the case in which the data is non-Euclidean. Our overall aim is to explore how non-Euclidean dissimilarity data can be used to analyse object and learn object categories. Given a set of N objects and a dissimilarity measure that compares pairs of objects, a symmetric $N \times N$ dissimilarity matrix D in which the elements are the pairwise dissimilarities. If the distance measure is asymmetric, the dissimilarity matrix is usually transformed into a symmetric one by averaging $d_{ij} = \frac{d_{ij}+d_{ji}}{2}$. In shape analysis, the variants of weighted edit distances and the Hausdorff distances are used as dissimilarity measures. These measures are derived from graph matching process and rarely metric or Euclidean, though more intuitive and effective than their Euclidean counterparts [85].

Our starting point is to study the non-Euclidean aspects of the dissimilarity matrix. Given a symmetric pairwise dissimilarity matrix D, its similarity (Gram) matrix is defined as $-\frac{1}{2}D^2$ and the centered Gram matrix

$$G = -\frac{1}{2}JD^2J$$

where $J = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$is the centering matrix and $\mathbf{1}$ is the all-ones vector of length N. With the eigendecomposition of the Gram matrix $G = \Phi\Lambda\Phi^T$, a symmetric dissimilarity matrix is Euclidean if all eigenvalues of the corresponding Gram matrix are non-negative [42]. One way to gauge the degree to which a pairwise distance matrix exhibits non-Euclidean artefacts is to analyze the properties of its centered Gram matrix. The degree to which the distance matrix departs from being Euclidean can be measured by using the relative mass of the negative eigenvalues [86]:

$$NEF = \sum_{\lambda_i < 0} |\lambda_i| / \sum_{i=1}^{N} |\lambda_i|$$

. This measure is called the negative eigefraction (NEF). Its value is zero when the distances are Euclidean and increases as the distance becomes increasingly non-Euclidean. Another measure of the non-Euclidean behavior is the ratio of the absolute values of the smallest negative eigenvalues (the largest negative eigenvalues in magnitude) to the largest positive eigenvalues. This is called negative eigenratio (NER) and it is zero when the distances are Euclidean [86]:

$$NER = \frac{\max_{\lambda_i < 0} |\lambda_i|}{\max_{\lambda_i > 0} \lambda_i}$$

.

A distance measure is considered to be non-metric if it is either non-symmetric, negative or violates the triangle inequality. The non-metricity of the data is usually defined by counting the number of metric violations. A distance measure is rarely negative and an asymmetric measure can be made symmetric directly by averaging, so we will assume that the distances are all positive and symmetric. As dissimilarity matrix usually does not satisfies the triangle inequality. There are two measures for the triangle inequality violation. One is computed as the fraction of three-point sets which violate the triangle inequality. This is called Triangle Ratio (TR). The other is computed as the constant value that makes the dissimilarity matrix satisfy the triangle inequality by adding it to the off-diagonal elements of D [86]. This is called Triangle Constant (TC) here.

$$TC = \max_{i,j,k} |d_{ij} + d_{ik} - d_{jk}|$$

If the Triangle constant is zero, the pairwise dissimilarity is considered to be metric but not necessarily Euclidean.

However, these four test measures do not reveal the origins of non-Euclidean or non-metric artifacts in the data. We develop a new measure which gauges the extent to which individual data gives rise to departures from being Euclidean. This allows us to assess whether the non-Euclidean artifacts in a dataset can be attributed to individual objects or are distributed uniformly. If the non-Euclidean artefacts are contributed solely by the set of distances to a few "outlier" objects, it is possible to restore the data to a Euclidean state by editing (i.e. removing) these objects from the dataset. Based on this idea we introduce the notion of measuring the contribution of each object to the negative eigenfraction of a dissimilarity matrix. That is, the fraction given by the sum of the negative distances originating from an individual object to all the remaining objects, divided by the total.

The matrix of kernel embedding co-ordinates is given by $Y = \Lambda^{1/2}\Phi = (\mathbf{y}_1, ..., \mathbf{y}_N)$, where $\Lambda = \text{diag}(\lambda_1, ..., \lambda_N)$ is the diagonal matrix with the ordered eigenvalues of centered Gram matrix as elements and $\Phi = (\phi_1|...|\phi_N)$ is the eigenvector matrix with the ordered eigenvectors ϕ_1, ..., ϕ_N as columns. When the centered Gram matrix has negative eigenvalues then those dimensions of the embedding associated with negative eigenvalues are represented by imaginary numbers, and those associated with positive eigenvalues by real numbers. In other words, the data are embedded into a pseudoEuclidean space [40].

Under the embedding,the coordinate vector of point j is $\mathbf{y}_j = (\sqrt{\lambda_1}\Phi_{1j}, ..., \sqrt{\lambda_i}\Phi_{ij}, \sqrt{\lambda_N}\Phi_{Nj})^T$. The contribution to the squared distance between two points k and e is

$$d_{ke}^2 = \sum_i (\mathbf{y}_k(i) - \mathbf{y}_e(i))^2 = \sum_i \lambda_i(\phi_{ik} - \phi_{ie})^2 = \sum_{\lambda_i<0} \lambda_i(\phi_{ik} - \phi_{ie})^2 + \sum_{\lambda_i>0} \lambda_i(\phi_{ik} - \phi_{ie})^2 \tag{3.1}$$

The sum of the squared distance consists of the negative squared distance associated with the negative eigenvalues, and the positive squared distance associated with the positive eigenvalues. Hence the sum of the negative squared distances from point k to all the remaining points is

$$d_{k-}^2 = \sum_{e \neq k} \sum_{\lambda_i < 0} \lambda_i (\phi_{ik} - \phi_{ie})^2, \tag{3.2}$$

and the sum of positive squared distances from point k to all the remaining points is:

$$d_{k+}^2 = \sum_{e \neq k} \sum_{\lambda_i > 0} \lambda_i (\phi_{ik} - \phi_{ie})^2 \tag{3.3}$$

We define the fraction of negative squared distances from point k as

$$NEC = \frac{|d_{k-}^2|}{|d_{k-}^2| + |d_{k+}^2|} \tag{3.4}$$

We call it negative eigenvector contribution (NEC) here. Its value is zero for all objects (data points) when the distances are Euclidean and bigger for outlier objects. Thus the measure can be useful to identify whether the non-Euclidean behavior is caused by a few outlier objects.

3.3 The Informativeness of Negative Eigenvalues

As a non-metric or a non-Euclidean dissimilarity can not be embedded into a Euclidean space without distortion, any procedures for enforcing Euclideanness will distort the data. Most correction procedures involve neglecting the contribution of negative eigenvalues like the kernel embedding (positive space embedding) and the associated Euclidean embedding. The question arises of whether negative eigenvalues are informative and whether the removal of negative eigenvalues leads to information loss. It is commonly assumed that the negative eigenvalues are associated to noise and cutting the negative eigenvalues would improve or not change the results, if the magnitude of the negative eigenvalues is small. Laub et al. [71, 72] categorized negative eigenvalues into two cases based on its shape once they are sorted. The first is flat negative spectrum if the negative eigenvalues falls slowly and appear linearly downward trailing trend compared to all the eigenvalues. The second is the non-trivial spectra if there are strongly decreasing negative eigenvalues in the end. Duin et al. [30] showed that the removal of negative eigenvalues deteriorates classification performance on some datasets . Laub et al [72, 71] further demonstrated that negative eigenvalues contain relevant features for classification tasks on some datasets. However, they focus on the non-trivial spectra of pairwise data and consider the dissimilarity data with flat negative eigenvalues is commonly associated with noise and the negative eigenvalues can be cut off in the correction procedure.

Dataset	size	NEF	NER	TC	Original D	Positive D_p
		(%)	(%)		(%)	(%)
CatCortex	65	20.83	27.21	1.97	12.31± 4.07	**6.15 ± 2.98**
Chickenpieces-5	446	21.64	1.73	0	**34.53± 2.25**	46.19± 2.36
Chickenpieces-10	446	25.72	2.34	3.86	**16.14± 1.74**	34.5 ± 2.25
Chickenpieces-15	446	28.61	2.80	4.26	**7.40 ± 1.24**	24.0± 2.02
Chickenpieces-20	446	30.75	3.42	4.22	**6.28 ± 1.15**	17.04 ± 1.78
Chickenpieces-25	446	31.99	3.71	4.10	**4.26 ± 0.96**	14.13 ± 1.65
Chickenpieces-30	446	33.07	4.30	4.01	**4.48 ± 0.98**	13.00 ± 1.59
Chickenpieces-35	446	33.94	4.59	3.98	**6.28 ± 1.15**	15.02 ± 1.69
Chickenpieces-40	446	34.46	5.28	4.10	**8.74 ± 1.34**	15.25 ± 1.70
CoilDelftDiff	288	12.77	5.10	2.54	47.22± 2.94	47.57 ± 2.94
CoilDelftSame	288	2.73	18.08	0	64.58 ± 2.82	60.76 ± 2.88
CoilYork	288	25.76	4.61	4.63	**23.26 ± 2.49**	33.68 ± 2.78
FlowCyto-3	612	27.08	18.62	4.86	36.11± 1.94	41.01 ± 1.99
Newsgroups	600	20.15	4.90	2.07	24.83± 1.76	28.15 ± 1.84
ProDom	2604	4.31	1.09	3.18	0.23 ± 0.01	**0.19 ± 0.01**
Protein	213	0.07	0.24	0	1.88 ± 0.93	1.88 ± 0.93
WoodyPlants	791	22.94	5.64	4.17	9.99± 1.07	10.49 ± 1.09
Zongker	2000	41.94	35.35	4.54	43.95± 1.11	**16.70 ± 0.83**

TABLE 3.1: Classification errors of the 1NN for the original dissimilarity and positive subspace dissimilarity using leave-one-out cross validation. Better classification results are in bold.

For most dissimilarity datasets, it is hard to interpret the meaning of the negative eigenvalues. For classification tasks, it is natural to think that the negative eigenvalues contain relevant information for classification if the classification results with non-Euclidean data are better than those with corrected data. Duin et al present classification results on dissimilarity space representation with linear SVM and some density classifiers [28, 29] and show that the negative eigenvalues can be informative. Since all dissimilarity matrices, Euclidean or non-Euclidean, can be computed directly on nearest neighbour classifier without any preprocessing steps like dimensionality reduction or prototypes selection, we restrict our experiments to 1 Nearest Neighbour (1NN) classifier to make the comparisons with Euclidean corrections without preprocessing steps. Here we compute the 1NN based on the original and positive space dissimilarities on a set of public domain dissimilarity matrices used in various applications from the SIMBAD project [29].

3.3.1 Dataset Description

The Catcortex dataset contains dissimilarities based on the connection strengths between 65 cortical areas of the cat brain from four regions. Chickenpieces contains a set of dissimilarity matrices whose elements are weighted edit distance between two contours of chicken pieces images represented by strings. CoilDelftDiff, CoilDelftSame and CoilYork are three dissimilarity datasets extracted from feature points detected in the COIL image database computed using different graph edit distances. The CoilYork data is obtained from a set of dissimilarity measurements between four objects from the COIL database. There are 72 equally space views for each object. A sample view of each object is illustrated in Figure 5.4. A graph for each image is constructed by extracting feature points from the object images and then performing Delaunay triangulation from the feature points [126]. The distance between a pair of images is the graph distance which is computed by the graph matching algorithm of Gold and Ranguranjan [39]. FlowCyto contains four histogram dissimilarities for samples of breast cancer tissue. Newsgroups contains dissimilarities for messages in four classes of newsgroups. Protein contains the dissimilarities of protein sequences based on an evolutionary measure of distance. ProDom contains dissimilarities between a protein sequences from the protein domain families Corpet computed by structural alignments. Woodyplants50 contains the shape dissimilarities between leaves of woody plants. Zongker contains the dissimilarities between handwritten digits based on deformable templates.

Chickenpieces-cost45 contains 8 dissimilarity matrices from a weighed edit distance. Chicken pieces [86, 30] data contains 446 binary image in five classes illustrated in Figure 3.1 : breast (96 examples), back (76 examples), thigh and back (61 examples), wing (117 examples) and drumstick (96 examples). It generates different distance matrix with straight line segment of a fixed length L and the angles between the neighbouring segments and editing cost C. Our experimental results are computed from the data with cost $= 45$ and $L = \{5, 10, 15, 20, 25, 30, 40\}$. The originally asymmetric dissimilarities are made symmetric by averaging [86, 30]. The Chickenpieces data is a useful set for the study of non-Euclidean dissimilarities, because there is a set of parameters which can be varied to change the level of non-Euclidean artefacts. Duin et al [86, 30] showed that the dissimilarity becomes increasingly non-Euclidean as both the negative eigenfraction and the negative ratio grow with increasing L.

To characterize the dissimilarity matrix, the size of the matrix, the metric measure TC and Euclidean measures (NEF, NER) are computed and shown in Table 3.1. The last two columns are the classification errors using leave-one-out crossvalidation of the 1NN for the original dissimilarity matrix (D) and the Euclidean matrix (D_p) obtained by removing the negative eigenvalues . Similar to the results in [29], in almost all dissimilarity datasets, we see the original, uncorrected, non-Euclidean dissimilarities similarly perform or better. For datasets like chickenpieces and the Coilyork, the corrected Euclidean dissimilarities obviously deteriorates the classification performance. This means the negative eigenvalues in such datasets are informative for classification using 1NN classifier.

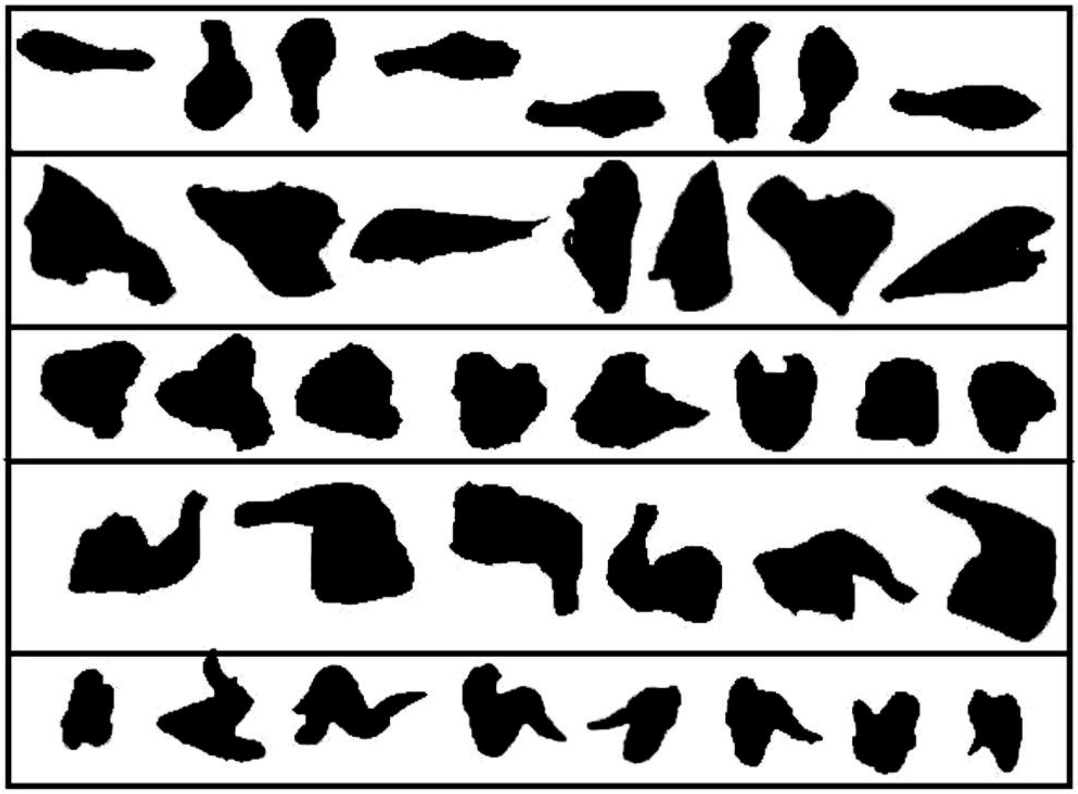


FIGURE 3.1: Chickenpieces image

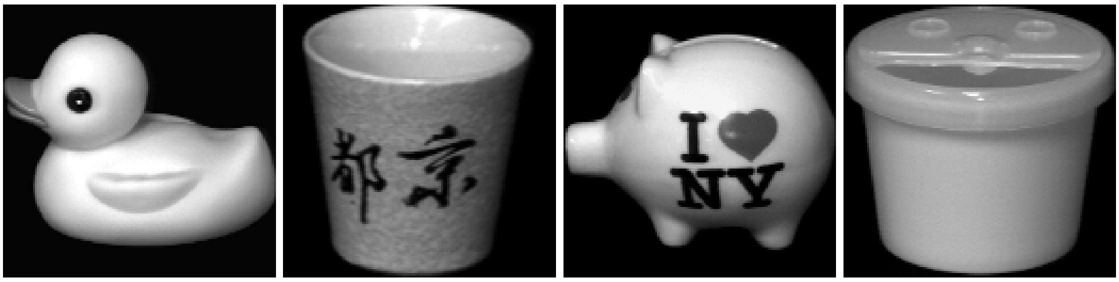


FIGURE 3.2: CoilYork objects image

Correction procedures that ignore the negative eigenvalues cause information loss. While for some datasets such as Newsgroups and Zongker, the resulting dissimilarities obtained by truncating negative eigenvalues give better classification performance. For such data, the negative eigenvalues might be noisy. Note the information loss is measured by the classifier performance before and after correction and is subject to the used classifier.

In addition to this, we have also explored the shape of the negative eigenvalues of the informative datasets and found out that the informative, non-Euclidean dissimilarities have a flat negative spectrum as we can see in Figure 3.3 and 3.4. In the literature [71], a flat negative spectrum where the negative eigenvalues appear linear downward trailing trend, is usually considered as noise and ignoring the negative eigenvalues is used for correction. As a result, the negative eigenvalues from the Chickenpieces and the CoilYork data might be caused by noise. But Table 3.1 shows the removal of negative eigenvalues cause information loss for classification. This indicates that ignoring the negative eigenvalues should not be generally used as a proper procedure for correcting and denoising non-Euclidean dissimilarity data even if the negative spectrum is flat. In addition to this, in order to prove that ignoring negative eigenvalues is not a proper way for denoising Gaussian noise if the the negative eigenvalues is caused by Gaussian noise, we experimentally add Gaussian noise to a set of Euclidean distances and investigate the effects of the removing negative eigenvalues on distances.

EXAMPLE Let D be a Euclidean dissimilarity matrix of four objects A through D

$$D = \begin{pmatrix} 0 & 1.5 & 3.0 & 1.0 \\ 1.5 & 0 & 3.0 & 2.0 \\ 3.0 & 3.0 & 0 & 3.0 \\ 1.0 & 2.0 & 3.0 & 0 \end{pmatrix} \tag{3.5}$$

we generate a non-Euclidean matrix $D1$ by adding gaussian noise to D

$$D_1 = \begin{pmatrix} 0 & 0.9555 & 3.0163 & 1.5503 \\ 0.9555 & 0 & 3.2763 & 2.7721 \\ 3.0163 & 3.2763 & 0 & 3.0430 \\ 1.5503 & 2.7721 & 3.0430 & 0 \end{pmatrix} \tag{3.6}$$

with negative eigenfraction $NEF = 0.032$

Now we apply kernel embedding which removes the negative eigenvalues to obtain a Euclidean matrix $D2$

$$D_2 = \begin{pmatrix} 0 & 1.1462 & 3.0385 & 1.6482 \\ 1.1426 & 0 & 3.2865 & 2.7729 \\ 3.0385 & 3.2865 & 0 & 3.0490 \\ 1.6482 & 2.7729 & 3.0490 & 0 \end{pmatrix} \tag{3.7}$$

Compared to the original Euclidean dissimilarity matrix D, the new Euclidean dissimilarity matrix in which negative eigenvalues are removed is distorted and the original ranking pattern is destroyed, especially in the case of the distances from object A to object B and object C which object C is the closer to C than to B in the original dissimilarity matrix. This illustrates that the non-Euclidean dissimilarity caused by noise can not be reversed back to the original Euclidean distances by simply removing the negative eigenvalues, as negative eigenvalues are not correspondingly exactly to the noise. Note this is a particular example to show that removing the negative eigenvalues is not a proper way to denoise non-Euclidean dissimilarities even if the non-Euclidean dissimilarities are caused by noise. The next section gives statistical results on the spectrum of non-Euclidean dissimilarities with additions of noise. Further more, the negative eigenvalues are informative if the classification results become worse by removing them. This also leads to the question of what are the sources of the negative values, as different correction procedures are based on the assumption of the causes of the negative eigenvalues.

3.4 The Causes of Non-Euclidean Data

We begin by identifying three reasons for non-Euclidean behaviour [29]. Several models are developed to explain these spectra and simple measures are presented to distinguish the three sources of negative eigenvalues displayed in Figure 3.5.

Manifold If the data points reside on a curved manifold, then the distances between them are intrinsically non-Euclidean (but still metric). This is one possible source of non-Euclidean distances. Here we model such data as points on the surface of a sphere, a simple surface where distances are easy to compute. It is simple to simulate patches with various degrees of curvature that depart from Euclidean behavior by changing the curvature of the patch. The dissimilarity measurements on the sphere are metric but non-Euclidean.

Extended objects If objects are not point-like but rather are extended in space, then the distances between them are measured between the closest points on their surface. As a result the distances will be non-Euclidean and possibly non-metric. Delft's balls data [29, 28] is a typical example. Randomly positioned balls are generated with varying radius. The pairwise dissimilarities are the surface distances between the balls. As a result only the pairwise distances between balls with zero radius are Euclidean. It is also simple to modify the degree of non-Euclidean behaviour by adjusting the radii of the balls.

Gaussian noise The final source is Gaussian noise added to the original Euclidean dissimilarities. This will generate data that is both non-Euclidean and non-metric.

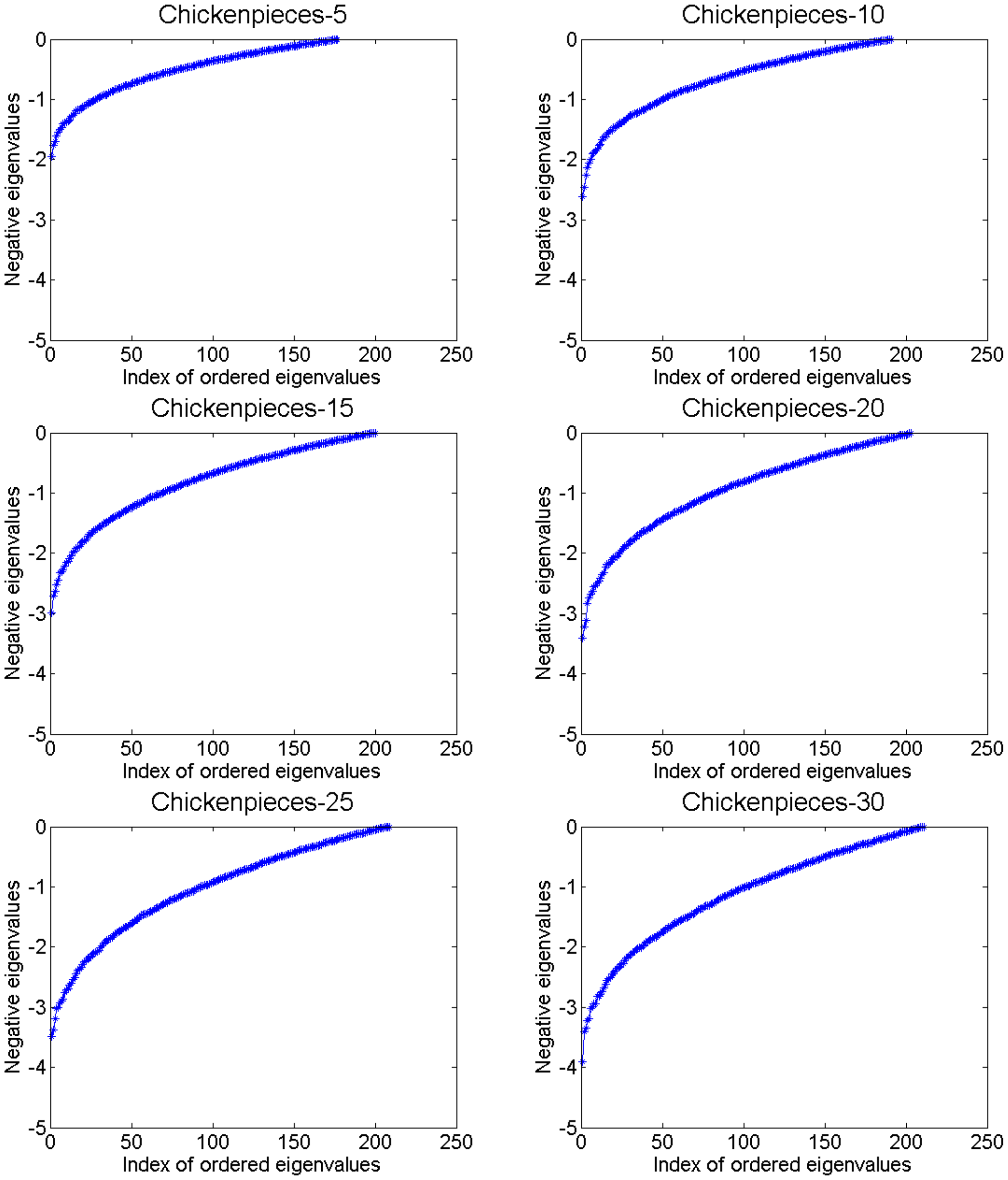


FIGURE 3.3: The spectrum of the negative eigenvalues of Chickenpieces datasets.

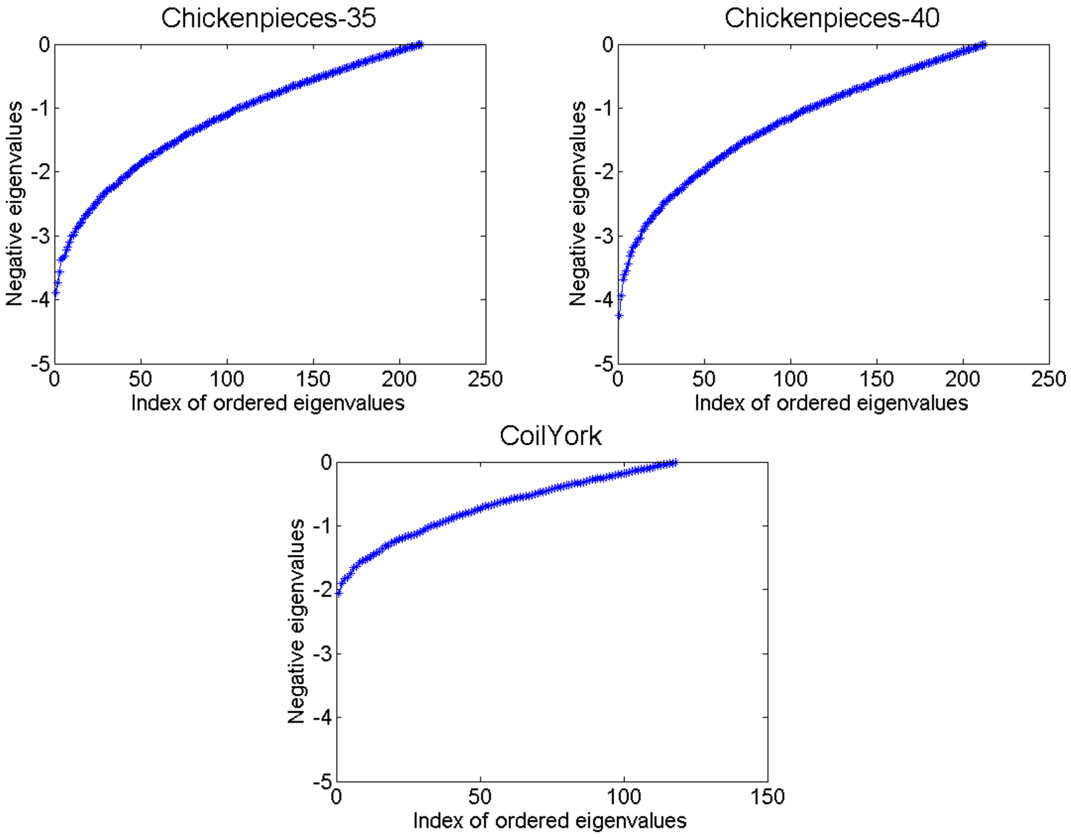


FIGURE 3.4: The spectrum of the negative eigenvalues of Chickenpieces-35, Chickenpieces-40, CoilYork.

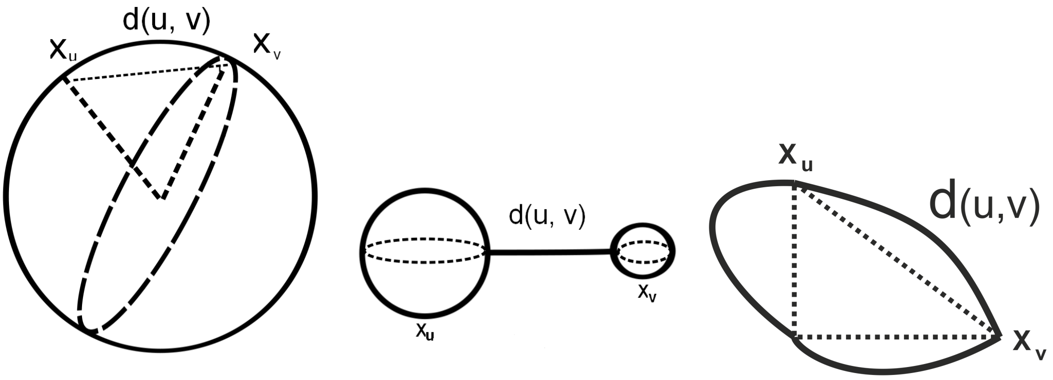


FIGURE 3.5: Three sources of negative eigenvalues: (1) Points on manifold (sphere); (2) Extended objects; (3) Gaussian noise.

3.5 Investigation

To model distances sampled from a manifold, we commence with 100 points uniformly distributed on the surface of a 3D sphere with unit radius. The spherical coordinates of an object are $x = (r \sin\theta \cos\phi, r \sin\theta \sin\phi, r \cos\theta)^T$ where r is the radius of the sphere, θ is the elevation angle($[0, \pi]$) and $\phi([0, 2\pi])$ is azimuth angle. The pairwise geodesic distances are computed as the lengths of great circle arcs between pairs of objects. We can change the range of the elevation angle to control the extent to which the patches deviate from a Euclidean surface, i.e. the degree of non-Euclideanness in the dissimilarity matrix. In total 100 initial configurations of points are used.

To model the extended objects, we pick 100 randomly positioned points in a 7D hypercube with length 100, and we take each point as the center of a ball with radius $r(r \geq 0)$. The balls do not overlap. The pairwise distance is the Euclidean distance between the centers of two balls minus the radii of the two balls. We regard the balls with radius greater than 0 as non-Euclidean balls. We vary the fraction of non-Euclidean balls, and take the fraction to be 0.1, 0,3, 0.5, 0.7 or 0.9 in our experiments. The radii of the non-Euclidean balls are 2, 3 or 4. We also generate 100 balls with uniformly distributed radii ranging from 0 to 4.

To model Gaussian noise, we commence with 100 randomly positioned points in a 3D Euclidean space and calculate the Euclidean dissimilarity matrix. Then we add Gaussian noise with zero mean and several standard deviation values to the off-diagonal elements of the dissimilarity matrix to generate a non-Euclidean dissimilarity matrix. The value of the standard deviation of Gaussian noise is 0.1, 0.3, 0.5, 0.7 and 0.9.

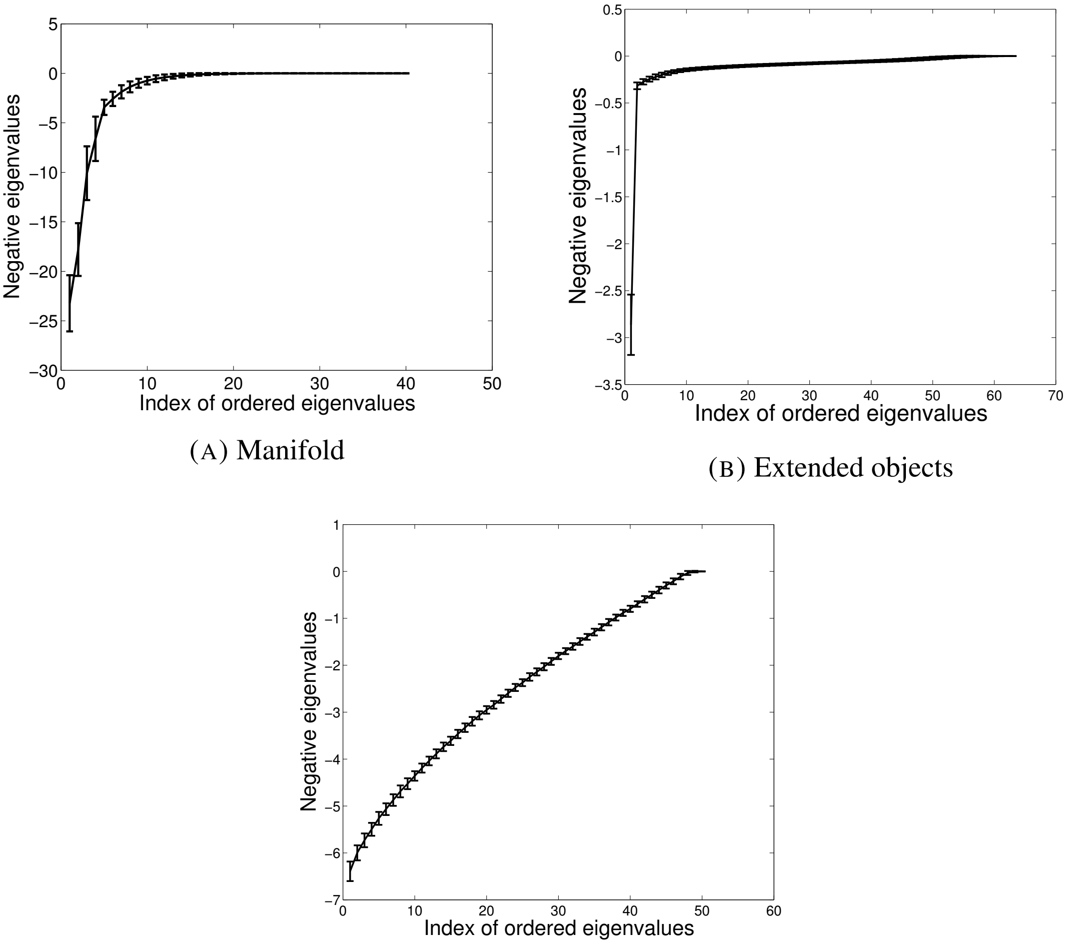
To ensure the results are comparable over the dissimilarity data in various ranges and scales, all of the dissimilarity metrics are scaled such that the average dissimilarity is unity. We calculate the negative eigenvalues of each dissimilarity matrix and fit the average negative spectrum by an exponential curve to obtain the slope b, the intercept a and the average triangle constant C. The whole process is repeated for a sample sizes of

500 and 1000 points.

3.5.1 Negative Spectrum

We analyse the three simple modes, which simulate the occurrence of non-Euclidean pairwise data. We commence by examining the negative spectrum of the Gram matrix under the three models. Figure 3.6 shows that the non-Euclidean dissimilarities from the sphere and balls data-sets have spectrum which contain a strong negative component, with a concentration towards the low end of the spectrum. The non-Euclidean dissimilarities from Gaussian noise have a more slowly decreasing negative spectrum. Each of these negative spectrum appear to follow an exponential decay. Thus the slope and the intercept from an exponential fit should be able to discriminate at least the Guassian noise model from the remaining two models. An exponential curve of the form $y = ae^{bx}$ is fitted to the data, with b the slope and a the intercept. These two parameters are used as measures to characterize the negative spectrum.

Figure 3.7 shows the slope b as a function of the metric constant value C from the non-Euclidean dissimilarities with a sample size of 100, 500 and 1000 respectively on the sphere, the "balls" data and Gaussian noise. As the negative spectrum of the Gram matrix from the Euclidean points with Gaussian noise appears to be in a flat and linear in shape, so the value of slope b is very small with a value around -0.04. For the dissimilarities from the extended objects, the negative spectrum has a very sharp decreasing negative tail (just few significant negative eigenvalue), so the value for the slope b has a larger magnitude. Comparing the points on sphere and the ball data, there are several negative eigenvalues in the tail and the decrease is less sharp. This may explain why the slope of the non-Euclidean dissimilarities on the sphere is intermediate between that of the Gaussian noise and the non-Euclidean balls data. Another interesting finding is that the number of objects is not correlated with the slope, especially for points on the sphere and Gaussian noise. In terms of the parameters, the three sources of negative eigenvalues are



(A) Manifold

(B) Extended objects

(C) Gaussian noise

FIGURE 3.6: (A)(B)(C) show the negative eigenvalues of the resulting Gram matrix of 100 points on the sphere, from extended objects and Gaussian noise as a function to the index of ordered negative eigenvalues.

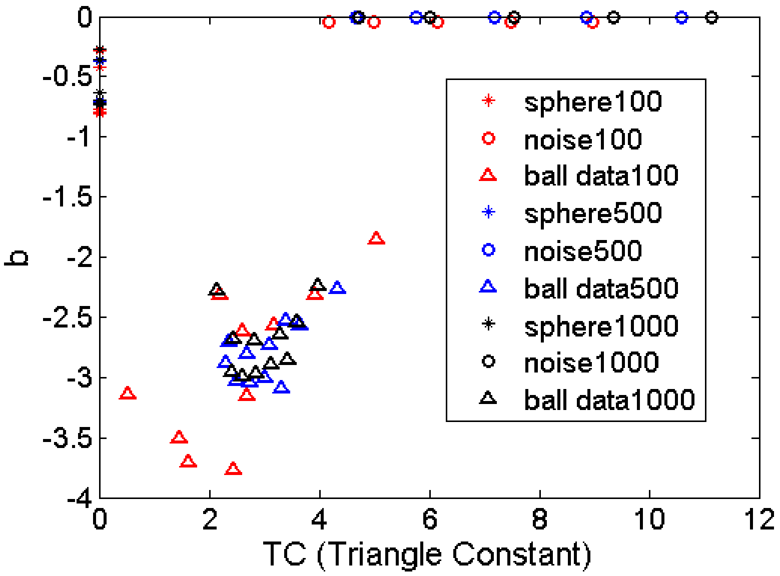


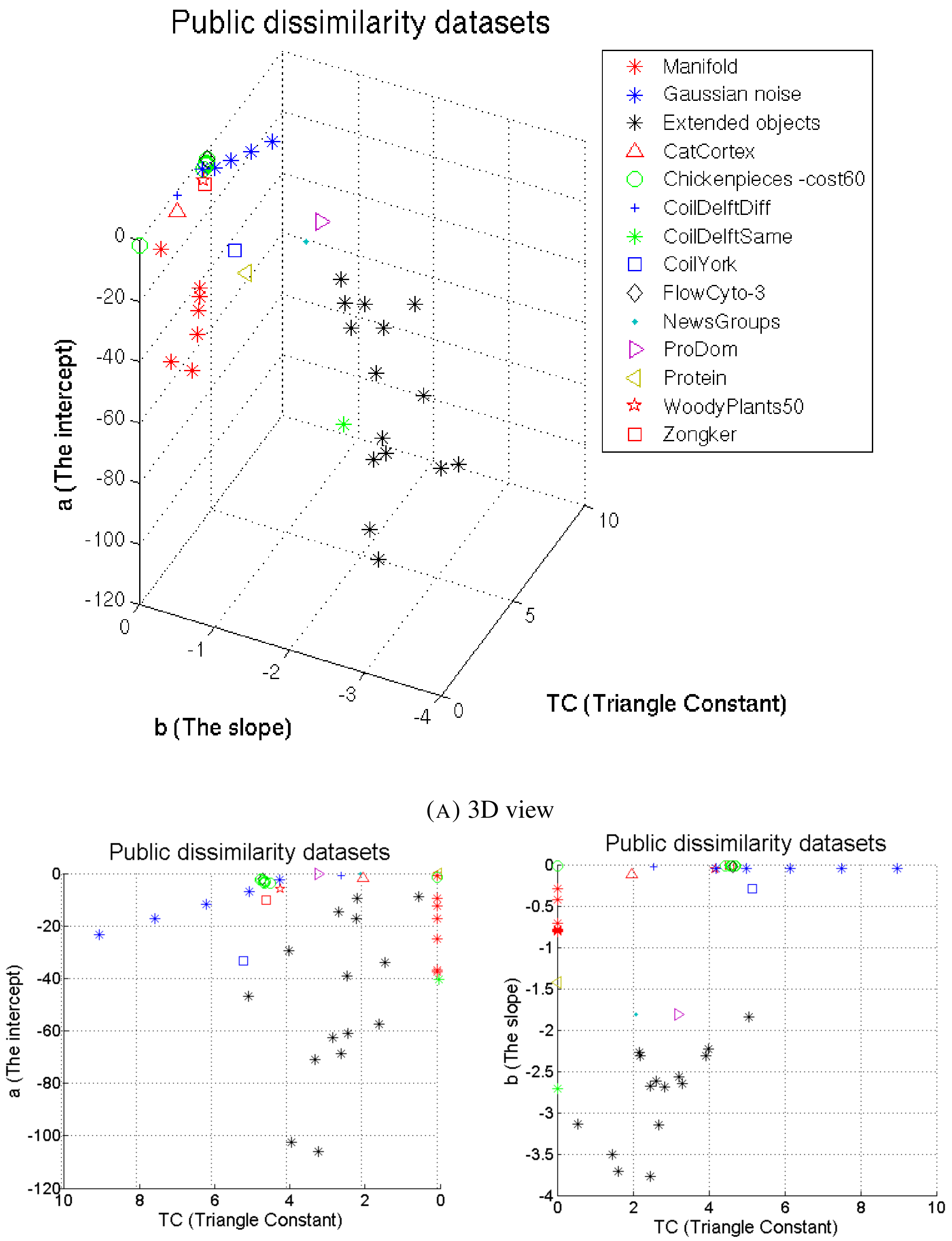
FIGURE 3.7: The artificial non-Euclidean dissimilarity data caused by the manifold the data resides on, the extended objects and Gaussian noise respectively with a sample size of 100, 500 and 1000.

well separated from each other.

In this section we present the analysis of three possible sources of the non-Euclidean behavior: manifold, extended objects and Gaussian noise. As each of the negative spectrum appear to follow an exponential decay, an exponential curve of the form $y = ae^{bx}$ is fitted to the negative spectrum. The triangle constant C is computed and added so as to increase the amount of data that satisfies triangle equality.In terms of the parameters the three sources of negative eigenvalues from the artificial dataset are well separated from each other. As a result, we can identify the three modeled sources of non-Euclidean behavior by using a, b and C, the three measures.

3.5.2 Investigation on Public Dissimilarity Datasets

We now use the three models to analyze a set of public dissimilarities used in various applications. The top figure in Figure 3.8 shows the intercept, the slope b and the Trian-



(A) 3D view

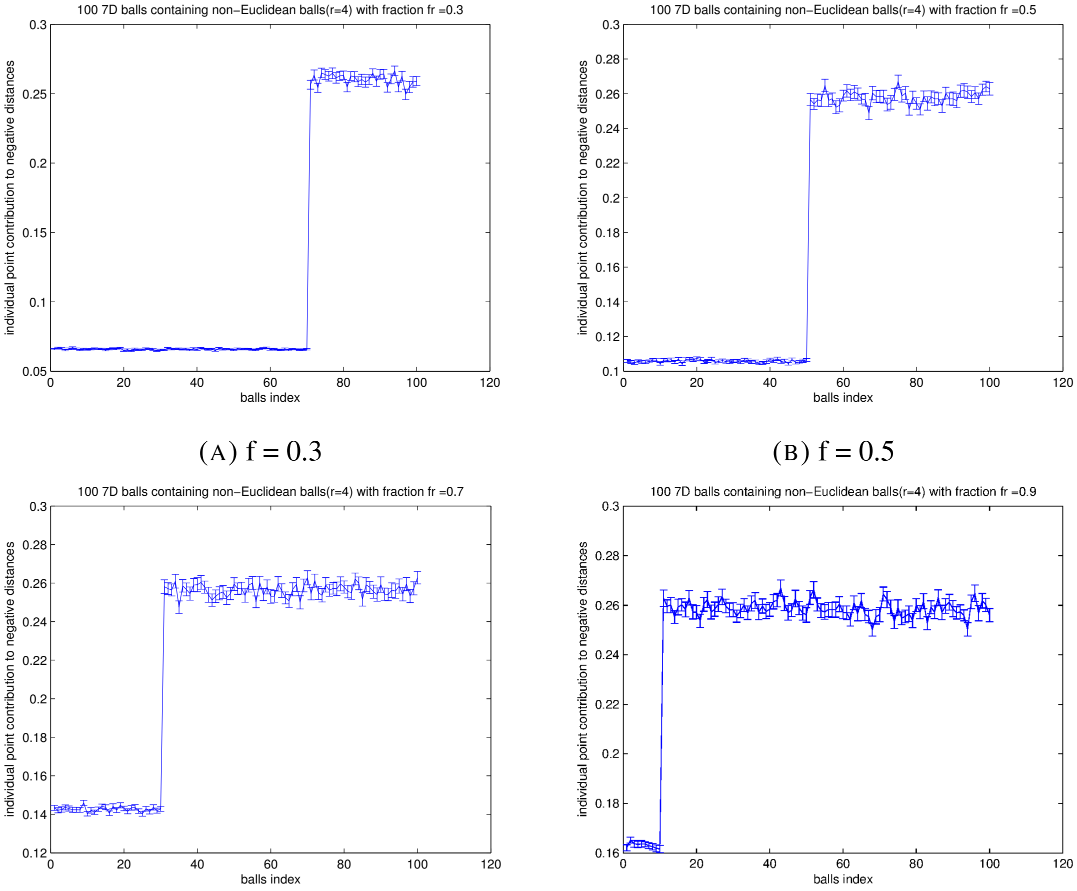
(B) 2D view

(C) 2D view

FIGURE 3.8: (A) shows the three measures for public dissimilarity data, (B) shows the intercept a as a function of the Triangle Constant TC, (C) shows the slope b as a function of the Triangle Constant TC for visualisation.

gle Constant (TC) for the samples of 100 objects, and a set of public dissimilarity data. The bottom plots show the top plot's 2D view for visualisation. The plots indicate that the non-Euclidean behaviour of Catcortex, Chicken pieces, CoilDelftDiff and FlowCyto-3,Woodyplants50, Zongker are likely to arise from Gaussian noise. On the other hand, the non-Euclidean behaviour of the Newgroups, ProDom and CoilDelftSame datasets is likely to arise the non-Euclidean distances of a few outlying objects. CoilYork, the cause may be a combination of data residing on a manifold and the Gaussian noise. For the Protein dataset it may be a combination of data on the manifold and extended objects.

Figure 3.9 shows the objects' negative eigenvector contribution in dissimilarities from 100-ball data with various fraction of non-Euclidean balls. It shows that non-Euclidean balls have high contribution. As the fraction of non-Euclidean balls increases, the contribution measure of Euclidean balls increases as well. It shows that objects' negative eigenvector is useful to identify the outlier objects which cause non-Euclidean behavior. We plot each object's negative eigenvector contribution for the Protein dataset in Figure 3.10. This shows that the negative eigenvalues are caused by the non-Euclidean distances between just a few objects. The protein data is almost Euclidean with a very small negative eigenfraction value of 0.001. We have explored the effect of applying a leave one out nearest neighbor classifier to the dataset. When we edit out the effect of the outlier objects distances by adding a constant to the squared distances to the remaining objects, we obtain only a slightly smaller error rate of 0.47% compared to 1.9% for the original distances. For the other datasets, each object contributes almost equally to the non-Euclideanness of dissimilarities. There is no significant difference among each object's negative eigenvector contribution.



(A) f = 0.3

(B) f = 0.5

(C) f = 0.7

(D) f = 0.9

FIGURE 3.9: (A)(B)(C)(D) show the individual object's contribution to the negative distances, measured by the negative eigenvector contribution (NEC), in the balls data with various fractions of non-Euclidean balls.

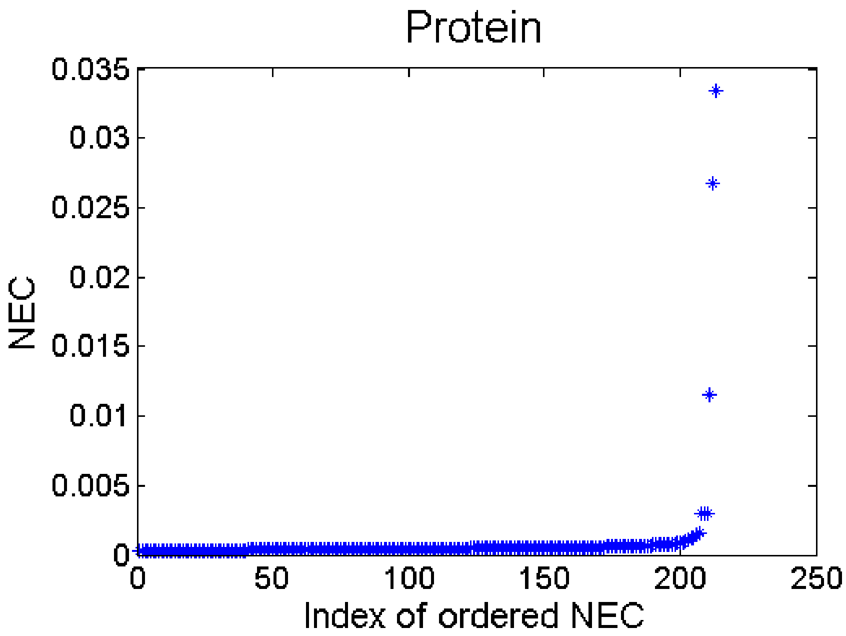


FIGURE 3.10: Ordered negative eigenvector contribution (NEC) for objects in the Protein dataset.

3.6 Summary

Negative eigenvalues might be informative for classification. This can be determined by checking the performance before and after removing the negative eigenvalues. We experimentally demonstrate that the flat spectrum can be informative though it can caused by Gaussian noise illustrated in the CoilYork dataset and the Chickenpieces dataset, thus we can not use simple procedures to ignor them. We have studied the occurrence of negative eigenvalues by modeling dissimilarity data from three sources. These three sources are manifold, extended objects and Gaussian noise. We investigated the shape of the negative eigenvalues and found out that all the shapes under the three sources follow an exponential decay but with different tails. We found out the negative eigenvalues caused by Gaussian noise appear flat, which is coherent with the idea that the flat spectrum are considered to be noise in the literature. The difference in the shape of negative eigenvalues can be characterized by the intercept and slope of an exponential function. Further more, the triangle constant is zero for metric dissimilarity resulted from manifold. Hence we extract three useful measures: a) the slope b) the intercept of an exponential curve c)the triangle costant as a means of detecting the sources of negative eigenvalues. By using the three measures, the three sources of negative eigenvalues are well separated from each other for artificial data. We presented experimental results to identify the origin of the negative eigenvalues on various dissimilarity datasets. The proposed three measures can be used as the preprocessing step in transforming the non-Euclidean dissimilarity into Euclidean space.

Chapter 4

Ricci Flow Embedding

4.1 Introduction

In the previous chapter we have shown that we can identify those three sources of the negative dissimilarity eigenvalues. In this chapter we will further explore the objects which lie on a manifold and introduce a way to rectify the non-Euclidean distances (make them more Euclidean) originated from a curved manifold. To make the non-Euclidean distances more Euclidean, we utilise the idea of Ricci flow on a constant curvature Riemannian manifold for the embedded data. We present the first application of Ricci flow in dissimilarity based learning. We provide implementation details for Ricci flow on the constant curvature manifold of either elliptic or hyperbolic background geometry. Our Ricci flow-based method can convert non-Euclidean dissimilarity problems into the manifold evolution process and offers a general framework for the non-Euclidean dissimilarity rectification. Instead of approximating the original dissimilarities by Euclidean distances, our approach is based on correcting the non-Euclidean dissimilarity matrix of objects to a set of Euclidean dissimilarities and try to preserve the underlying group structure of the data. In other words, we are seeking a mapping from the objects of a dissimilarity (or similarity) matrix to point-set in a vector space with more or similar discriminating power. In this chapter, we will introduce two Ricci flow based methods for correcting the non-Euclidean distances.

Our idea is based on considering the objects of interest to be represented by points on a manifold with local constant curvature, and the given dissimilarities to be the geodesic distances on the manifold between these points. In other words, we can regard a small portion of the space as a manifold of constant curvature (hypersphere). Manifolds can be considered to be embedded in some higher-dimensional Euclidean space, which is the ambient space. For example, manifold learning assumes the data is sampled from a low-dimensional manifold embedded in a high-dimensional ambient space. For the purpose of clarity, in this thesis, Euclidean distances refer to the distances in the ambient space, and geodesic distances refer to the distances on the manifold.

For an arbitrary set of non-Euclidean dissimilarities, the manifold will be curved. In contrast, a Euclidean surface will be flat. And the geodesic on the manifold and Euclidean distances in the ambient space are identical. Thus our idea is that we can obtain a set of Euclidean distances by transforming a curved manifold to a manifold with zero curvature. This process of evolving a manifold is called manifold flattening. To apply this technique, we require two tools: One tool is used to compute the initial curvature of local patches, and the other is used to update the curvature.

In the field of differential geometry, Ricci flow provides a way to evolve the manifold. Given the initial curvature of a manifold, Ricci flow is used to reduce its sectional curvature. The evolved manifold is less curved than the initial one. The process is repeated until the manifold has zero curvature. That is, the manifold becomes flat. The geodesic distance on the flat manifold is Euclidean. We model the manifold where the objects of interest lie as consisting of a set of local patches with individual constant curvatures. Then we use Ricci flow to obtain the rectified Euclidean distances. We call this technique Ricci flow embedding. This is a framework that allows data specified in terms of non-Euclidean distances rather than feature vectors to be transformed in a vector space. This helps us to apply machine learning techniques directly to the data. Hence given a method to accurately compute the sectional curvatures of local patches, we can flatten the manifold by using Ricci flow to correct a set of non-Euclidean distances. We use two ways to estimate the curvature of local patches and develop two corresponding Ricci flow embedding methods.

In our first method, we consider each edge to be a local patch (i.e a piecewise manifold), of which the weights are the given pairwise distances, and use the Ricci flow process to locate a Euclidean distance matrix. We call this process piecewise manifold embedding. Each local patch is regarded as a simple manifold with a constant curvature. Proposed by Lindman and Caelli [75], the sectional curvature of a manifold with constant curvature is determined by the Euclidean distance in the ambient space and geodesic distance on the manifold. To compute the curvature, we require a method to locate the Euclidean distance given the geodesic distance while preserving the geodesic distance as much as possible. Kernel embedding and Isomap embedding are two of these methods. Once the curvature is approximated, we use the Ricci flow process to flatten the local patches. The updated geodesic distances on the flat manifold is the Euclidean distance we aim for.

However, the method can prove unstable due to local fluctuations in edge curvatures. To overcome this problem, we develop a way to stabilise the first method by using a graph regularisation technique. The edge curvature is regularized using heat kernel regularisation and the Ricci flow process is used to locate a Euclidean distance matrix. In our second method, in addition to considering individual edge as a local patch, we also consider the neighborhood structure of the data by adding a curvature regularisation step. We consider each edge as a local patch, regularise the curvature over neighbourhoods and update the regularised curvature on the edge using Ricci flow to locate a set of new Euclidean distances. We call this process regularised Ricci flow embedding.

For our experiments we use the publically available SIMBAD project datasets, namely the CoilYork and Chickenpieces databases. We show how the manifold learning methods and differential geometry tool can be combined to correct non-Euclidean dissimilarity into a set of Euclidean dissimilarity.

4.2 Riemannian Space with Constant Curvature

Our aim is to rectify a set of non-Euclidean distances to make them more Euclidean. We consider the objects of interest to be represented by points on a manifold which is embedded into some Euclidean space, and the given dissimilarities to be the geodesic distances on the manifold between these points. For an arbitrary set of non-Euclidean similarities the manifold will be curved. In contrast, a Euclidean space will be flat and the geodesic and Euclidean distances will be identical. Our task is then to remove the curvature from the manifold to create a corrected set of Euclidean distances. We achieve this by evolving the manifold using Ricci flow. In this section we introduce specific theoretic backgrounds and motivations behind the above general idea. Please refer to [73, 67, 75, 123] for an overview of general Riemannian manifolds, the elliptical space and hyperbolic space.

We use Riemannian manifolds to represent the structure of a set of objects. A Riemannian manifold is a manifold designed by metric tensor g $= (g_{ij})$. The metric tensor is the first fundamental form of a manifold. It is positive definite and defines an inner product as a distance measure for the tangent spaces of the manifold in terms of coordinate system $x_1, x_2, \cdots, x_n$, where n is the dimensionality of the manifold.

$$ds^2 = \sum_{ij} g_{ij} dx_i dx_j \tag{4.1}$$

A manifold is viewed as a subspace of a larger ambient space (Euclidean or pseudo-Euclidean). Extrinsically, a two-dimensional spherical surface is a subset of a three-dimensional space. The intrinsic view only considers how the points in the surface stand in relation with each other, but not in relation to any external space. The geodesic distances are intrinsic, defined as the shortest length (distance) of curves connecting two points on the manifold. As finding the geodesic distances between two points on the manifold involves the complex task of solving a set of coupled second-order differential equations [123, 73, 67] which is not easy, we use manifolds on which it is simpler to find the geodesics. Riemannian manifolds with constant sectional curvature are such spaces. As an important parameter of a constant curvature Riemannian space, sectional curvature is a geometric description of the curvature of Riemannian manifold. It is the curvature of two-dimensional sections of the manifold. It is the Gaussian curvature of the locally defined surface which has the two-dimensional plane as a tangent plane at the point of interest. The Gaussian curvature is the product of the maximum and minimum curvatures of all geodesics passing through the point [75]. The curvature of a curve at a given point is $\frac{1}{r}$ for elliptic space and $\frac{1}{r}i$ for hyperbolic space where r is the radius of the circle that best fits the curve at that point. Gaussian curvature is defined in two dimensional manifold, and the sectional curvature is the analogue of Gaussian curvature for high dimensional manifold [124]. To understand the sectional curvature in terms of Gaussian curvature, considering all curves passing the point of interest p that are tangent to a two dimensional plane ω in the tangent space T_pM, a two-dimensional surface which belongs to the manifold is consisting of these geodesic curves. The sectional curvature defined on the two dimensional plane ω is the Gaussian curvature of the two dimensional surface at the point of interest. Thus the sectional curvature is constant everywhere for a Riemannian space. The constant curvature is positive in elliptic space, negative in hyperbolic space and zero in Euclidean space.

For any dimension n, there are exactly three types of Riemannian manifold with constant sectional curvature. Elliptic spherical space is an n dimensional hypersphere manifold with positive sectional curvature $K = 1/r^2$ everywhere, embedded in an $n+1$ dimensional Euclidean space. Euclidean space is an n dimensional space with sectional curvature $K = 0$. Hyperbolic space is manifold with negative sectional curvature $K = -1/r^2$. Since curved manifolds have non-Euclidean geometry and can produce non-Euclidean dissimilarity, elliptic space and hyperbolic space are two choices for embedding the non-Euclidean dissimilarities.

Elliptical Space The simplest model of elliptical space is a two dimensional surface of a sphere of radius r embedded in a three dimensional Euclidean space with coordinate system x_1, x_2, x_3. The distances from all the points on the sphere to its centre equal to the radius. A zero centered sphere is defined by

$$< \mathbf{x}, \mathbf{x} >= x_1^2 + x_2^2 + x_3^2 = r^2. \tag{4.2}$$

The points on the surface with spherical parameterisation are given by:

$$\mathbf{x} = (r \sin\theta_1 \sin\phi, r\cos\theta \sin\phi, r\cos\phi)^T \tag{4.3}$$

The infinitsimal distance element on the sphere is

$$\begin{aligned} ds^2 &= dx_1^2 + dx_2^2 + dx_3^2 \\ &= r^2 \sin^2\theta d\phi^2 + r^2 d\theta^2 \end{aligned} \tag{4.4}$$

where θ is the angle measured from the z axis, ϕ is the angle from the x axis in the xy plane. So the metric tensor is

$$g = r^2 \begin{bmatrix} \sin^2\theta & 0 \\ 0 & 1 \end{bmatrix} \tag{4.5}$$

Extending a two dimensional sphere to an (n) dimensional sphere in an $n+1$ dimensional space, the hypersphere surface can be defined by the constraints:

$$\sum_{i}^{n+1} x_i^2 = r^2 = 1/K \tag{4.6}$$

the points on the hypersphere are

$$\begin{aligned}\mathbf{x} &= (x_1, x_2, \cdots, x_{n+1}) \\ &= (r\cos\theta_1\cos\theta_2\cdots\cos\theta_n, r\sin\theta_1\cos\theta_2\cdots\cos\theta_n, \cdots, r\sin\theta_n)^T \end{aligned} \tag{4.7}$$

where $x_k = r\sin\theta_{k-1}\prod_{j=k}^{n+1}\cos\theta_j, \sin\theta_0 = \cos\theta_{n+1} = 1$. Similar to the two dimensional sphere, the metric tensor element on a hypersphere is the product of squared radius and a function determined by the the angular coordinates, which is derived by local derivatives on the n angular coordinates $\theta_1, \theta_2, \cdots, \theta n$

$$g_{ij} = \begin{cases} F_1(\theta_1, \theta_2, \cdots, \theta_n)/K & i = j, \\ 0 & i \neq j. \end{cases} \tag{4.8}$$

Hyperbolic Space For a hyperbolic space to be Riemannian and to retain its metric properties on distances, there is exactly one negative dimension. For example, the two dimensional hyperbolic space in three dimensions is

$$< \mathbf{x}, \mathbf{x} >= x_1^2 + x_2^2 - x_3^2 = -r^2. \tag{4.9}$$

The points on the surface with spherical parameterizations are given by:

$$\mathbf{x} = (r\sin\theta_1\sinh\phi, r\cos\theta\sinh\phi, r\cosh\phi)^T \tag{4.10}$$

The infinitsimal distance element on the sphere is

$$\begin{aligned} ds^2 &= dx_1^2 + dx_2^2 - dx_3^2 \\ &= r^2 \sinh^2 \theta d\phi^2 + r^2 d\theta^2 \end{aligned} \tag{4.11}$$

So the metric tensor is

$$g = r^2 \begin{bmatrix} \sinh^2 \theta & 0 \\ 0 & 1 \end{bmatrix} \tag{4.12}$$

Extending a two dimensional sphere to an (n) dimensional sphere in $n+1$ dimensional space, the surface can be defined by the constrains:

$$\sum_i^n x_i^2 - x_{n+1}^2 = -r^2 = 1/K \tag{4.13}$$

Similar to the sphere, retaining the spherical parameterisation by changing the last dimension (or angular coordinates) from $\sin$ to $\sinh$ as $\sin(i\theta) = i \sinh(\theta)$. The points on the hyperbolic space are

$$\begin{aligned} \mathbf{x} &= (x_1, x_2, \cdots, x_{n+1}) \\ &= (r \cos\theta_1 \cos\theta_2 \cdots \cosh\theta_n, r \sin\theta_1 \cos\theta_2 \cdots \cosh\theta_n, \cdots, r \sinh\theta_n)^T \end{aligned} \tag{4.14}$$

where $x_k = r \sin\theta_{k-1} \cosh\theta_n \prod_{j=k}^{n+1} \cos\theta_j, x_(n+1) = \sinh\theta_n, \sin\theta_0 = \cos\theta_{n+1} = 1$. Similar to the two dimensional sphere, the metric tensor element on a hypersphere is the product of squared radius and a function determined by the the angular coordinates, which is derived by local derivates on the n angular coordinates $\theta_1, \theta_2, \cdots, \theta n$

$$g_{ij} = \begin{cases} F_2(\theta_1, \theta_2, \cdots, \theta_n)/K & i = j, \\ 0 & i \neq j. \end{cases} \tag{4.15}$$

Since our objective is to transform non-Euclidean dissimilarities into Euclidean space and the points of non-Euclidean dissimilarities can be modelled on curved constant curvature manifold, our task becomes to transform a curved manifolds to a flat Euclidean manifold. The next section summarises how we achieve this by means of Ricci flow process.

4.3 Ricci Flow in Constant Curvature Riemannian Space

In geometric analysis, Ricci flow is an intrinsic curvature flow method [63], as it is not related to an extrinsic ambient space. Introduced by Richard Hamilton for topological classification of three-dimensional smooth manifolds, Ricci flow evolves a manifold so that the process to change the metric tensor is controlled by the Ricci curvature [14]. Essentially, this is an analogue of a diffusion process for a manifold. The geometric evolution equation is:

$$\frac{dg_{ij}}{dt} = -2K_{ij}, \tag{4.16}$$

where g_{ij} is the metric tensor of the manifold and K_{ij} is the sectional curvature.

Ricci flow is a famous form of intrinsic curvature flow in differential geometry. Hamilton [50] introduced the Ricci flow for Riemannian manifolds of any dimension in his seminal work, which deforms a given Riemmannian metric according to its curvature. Perelman has applied it to prove the 3-dimensional Poincaré conjecture [88]. Hamilton has proved the uniformization theorem for surfaces of positive genus by using the 2-dimensional Ricci flow [51]. Chow and Luo have studied the intrinsic relations between the circle packing metric and surface Ricci flow and established the theoretic foundation for discrete Ricci flow by proving the existence and convergence of the discrete Ricci flow [51]. This leads to a wide application of surface Ricci flow in surface parameterizations, shape analysis and geometric graphics [65, 45, 64, 132, 133, 97, 98]. All these works share the feature of representing surfaces as piecewise linear triangle meshes and using the Ricci flow to deform edge distances by discrete curvature. They discretise the Riemannian metric and the Gaussian curvature as the edge lengths and the angle deficits.

However, they are based on different geometric intuition. For example, both Jin et al. [65] and Gu et al. [45] extend the work from Luo and Chow on the combinatorial structures of triangular meshes by improving the gradient descent Ricci flow algorithm with Newton's method, the former focussing on hyperbolic space while the later works on Euclidean space. Zeng has investigated the Ricci flow on both Euclidean and hyperbolic background geometry in the context of characterizing 3D shapes [132, 133]. The Euclidean Ricci flow [65, 133] finds a flat metric that gives zero Gaussian curvature for all the interior vertices on a manifold of triangular meshes. In other words, it flattens the surface onto a plane. Figure 4.1 illustrates the simple example of Ricci flow process on a curved line. The curved line L_1 is flattered to L_2, the flattered curve L_2 has smaller curvature after Ricci flow process.

Generally the Ricci flow tends to expand the manifold if the manifold has negative curvature, and contract the manifold if it has positive curvature. The concepts of expanding and contracting means that Ricci flow increases or decreases the distances between points along the direction of sectional curvatures. Moreover, the stronger the curvature is, the faster is the expanding or contracting of the distances [109].

Prior discrete Ricci flow works are mostly based on surface Ricci flow defined in the two dimensional space, though Ricci flow can be defined for Riemannian manifolds of any dimensionality. Without assumptions on curvature, the behaviour of the metric evolving by Ricci flow is not straightforward. In our framework, the input manifold is a manifold with constant sectional curvature within a high dimensional space. Then we change the distances(metric) on the manifold as the curvature evolves. In elliptic space, we replace the metric tensor in 4.16 with 4.8. For simplicity, we assume only the sectional curvature changes with time. In this case, the metric tensor discretise as the sectional curvature. As a result, the Ricci flow in terms of the sectional curvature K is simplified by removing the function $F1$:

$$\frac{d(1/K)}{dt} = -2 \tag{4.17}$$

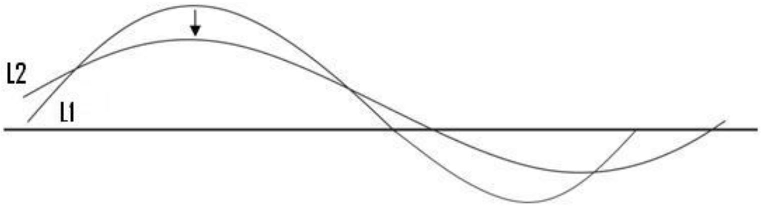


FIGURE 4.1: Ricci flow change the curvature: flatten a curve, the curvature $K \to 0$.

This equation shows that sectional curvature increases with time, since we have to decrease the curvature in order to evolve the curved manifold to the Euclidean space, we reverse the time by changing the sign of the time parameter. The Ricci flow for elliptic space is defined as:

$$\frac{d(1/K)}{dt} = 2 \tag{4.18}$$

Similarly, the Ricci flow for hyperbolic space is defined as:

$$\frac{d(1/K)}{dt} = -2 \tag{4.19}$$

We model the embedding manifold as consisting of a set of local patches with individual constant Ricci curvatures. These patches can be either elliptic (of positive sectional curvature) or hyperbolic (of negative sectional curvature). It is straightforward to re-express the Ricci flow in terms of the sectional curvature K:

$$\frac{dK}{dt} = \begin{cases} -2K^2 & \text{elliptic hypersphere,} \\ 2K^2 & \text{hyperbolic space.} \end{cases} \tag{4.20}$$

Under this evolution, the curvature moves towards zero for both types of patches, flattening the manifold. The solution of the differential equation is straightforward. Starting from the initial conditions $K = K_0$ at time $t = 0$, then at time t we have

$$K_t = \begin{cases} \frac{K_0}{1+2K_0 t} & \text{elliptic hypersphere,} \\ \frac{K_0}{1-2K_0 t} & \text{hyperbolic space.} \end{cases} \tag{4.21}$$

In this section, we review the basics of Ricci flow and constant curvature Riemannian geometry and develop a technique for evolving the curved manifold with constant curvature to Euclidean space. In fact, Ricci flow in constant curvature manifold provides us with a way to update the section curvature. The novel contribution here, hence, is to apply Ricci flow for correcting the non-Euclidean dissimilarities. Thus we can model the non-Euclidean dissimilarities using curved manifold with local constant sectional curvature and flatten the local manifolds via Ricci flow. In order to locate a set of Euclidean distances in the Euclidean space evolved from the curved manifolds, we need to know how the distances change during the manifold evolution. We also need a way to compute the sectional curvatures of the local patches where the data is located. The following section presents two algorithms and describes corresponding techniques to estimate the sectional curvatures and update the distances during the manifold evolutions.

4.4 Piecewise Ricci Flow Embedding

Our aim is to transform a non-Euclidean dissimilarity measure into a Euclidean one using the Ricci flow described in the previous section. We commence by representing the dissimilarity data using a weighted graph $G = (V, E, D)$, where the node set V represents the set of objects and the edges E are weighted with the pairwise dissimilarities D. We embed the graph onto a manifold which is embedded in some Euclidean space, so that the geodesic distance $d_g(u, v), (u, v) \in E$ between the positions of the nodes (points) u and v is equal to the dissimilarity on the edges. Let $\mathbf{y}_u$ be the embedding co-ordinates of the node (point) $u \in V$ in the Euclidean space and $Y = (\mathbf{y}_1|...|\mathbf{y}_{|V|})$ be the matrix with the embedding co-ordinates as columns. Under this embedding the Euclidean distance is

$d_E(u,v) = \sqrt{(\mathbf{y}_u - \mathbf{y}_v)^T(\mathbf{y}_u - \mathbf{y}_v)}$.

The embedding manifold is assumed to contain a set of local patches with individual constant sectional curvatures. Here we model each edge as a local patch, flatten the manifolds via the Ricci flow and evolve the distances by fixing the embedded Euclidean distances to get a set of Euclidean distances. We call this process piecewise manifold embedding. To apply this technique, we require a method to assign a sectional curvature to the edges. The sectional curvature of a local patch is the sectional curvature of the curve (edge) connecting two points. The patches can be elliptic with positive sectional curvature or hyperbolic with negative sectional curvature. We commence by showing how the Euclidean distances in some ambient space are estimated from the kernel embedding and the Isomap embedding, and then how the geodesic distances given by the non-Euclidean dissimilarities and the estimated Euclidean distances can be used to associate a sectional curvature with the edges [32, 75]. Next, we turn our attention to updating the geodesic distances by evolving the sectional curvatures and fixing the estimated Euclidean distances. Finally we summarise the steps in the algorithm for correcting non-Euclidean dissimilarities.

There are many ways to find a vector representation from a set of dissimilarities on manifold in the literature. The Euclidean distances are the inner product of vectors. In this thesis, we use the Euclidean distances deduced from the kernel embedding and Isomap embedding, as both the kernel embedding and Isomap embedding aims to preserve the embedded distances. During the piecewise Ricci flow embedding, we update the geodesic distances by fixing the Euclidean distances, and using the curvature to recompute the geodesic distances, and finally the geodesic distances are equal to the Euclidean distances in Euclidean space. Our objective is to obtain a set of Euclidean distances which preserve the original grouping structure. Thus we want the deduced Euclidean distances as close as possible to the geodesic distances, not just the ranking order. Both the embedding processes are performed on the initial non-Euclidean distance matrix and the Euclidean distances are calculated from the embedding vectors.

4.4.1 Euclidean distance from Kernel Embedding

The embedding procedure is straightforward. We project objects in positive space of the pesudo-Euclidean space and get the coordinates $\mathbf{y}$ and the corresponding Euclidean distances. The kernel embedding is equivalent to the classical MDS which finds an embedding that preserves the distances between points [110] when the given distances are Euclidean. Here is the procedure:

1. Given the initial distance matrix D, compute the centered Gram matrix $G = -\frac{1}{2}JD^2J$, where $J = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, D_0^2 is the elementwise squared distance matrix, so that the embedded coordinates have zero mean.

2. Take the $P(P < N)$ positive eigenvalues of the Gram matrix to form the P by P diagonal matrix Λ_P and project objects on the eigenvectors with positive eigenvalues using Young-Householder decomposition for preserving distance, where Φ_P is P by N matrix with each column is the positive eigenvectors. Y is the $P \times N$ matrix with the vectors of co-ordinates as columns,

$$G = \Phi\Lambda\Phi^T$$

$$Y = {\Lambda_P}^{1/2}\Phi_P = (\mathbf{y}_1|\mathbf{y}_2|\mathbf{y}_3|\dots|\mathbf{y}_N)$$

The Euclidean distance between objects $\mathbf{y}_u$ and $\mathbf{y}_v$ is:

$$d_E(u,v) = \sqrt[2]{(\mathbf{y}_u - \mathbf{y}_v)^T(\mathbf{y}_u - \mathbf{y}_v)} = \sqrt[2]{\sum_{i=1}^{N} \lambda_i(\phi_i(u) - \phi_i(v))^2}$$

The above equation shows that the embedded Euclidean distance decreases if some positive eigenvalue part is taken away, and increases if some negative eigenvalues part is taken away. In this case, the geodesic distance is smaller than the Euclidean distance, the curvature is negative, and the space is assumed to be hyperbolic space.

4.4.2 Euclidean Distance from Isomap Embedding

Isomap [110] is used to find a low dimensional representation with low distortion on geodesic distance on a manifold. The idea behind Isomap is to apply the MDS on the shortest distance matrix computed from the input distance data rather than directly on the input distances. The method was originally devised for dimensionality reduction like PCA and MDS. Here we use it for preserveing distances. For non-Euclidean dissimilarities, we first construct a nearest neighour graph from the input data, then compute the shortest distance (geodesic distance) by adding up connected edges on the shortest path between point and apply the MDS on the shortest distance matrix. Here is the procedure:

1. Construct the k nearest neighbour graph over the available dissimilarity data. Node u and v are connected by an edge if u is among k nearest dissimilarity neighbors of v or v is among k nearest dissimilarity neighbours of u.

2. Estimate the geodesic distances by computing the shortest path distances for all pairs in the neighbour graph by using the Dijkstra's algorithm, and get the matrix of geodesic distance D_G.

3. Apply the kernel embedding on the matrix of neighbour graph distance D_G and get the Euclidean matrix.

4.4.3 Sectional Curvature Computation

In the previous section we obtained the Euclidean distance in the ambient space deduced from the two embedding methods. In this section we show that both the Euclidean distances in the ambient Euclidean space and the geodesic distances between the objects of

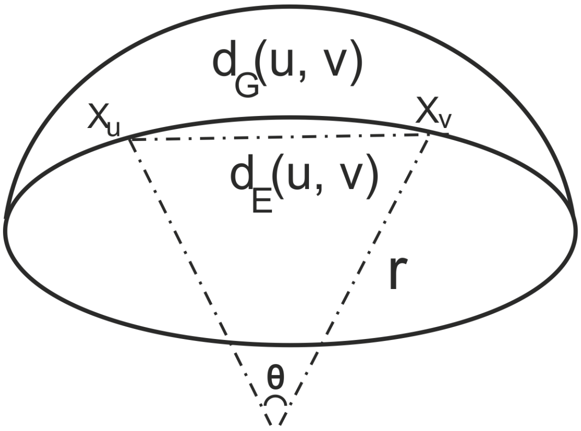


FIGURE 4.2: Illustration of the relationship between the geodesic distance, the Euclidean distance and the sectional curvature.

interest on the manifold can be used to make a numeric estimation of the sectional curvature associated with each edge. Since the radius of curvature is determined by the degree to which the geodesic bends away from the Euclidean chord and the sectional curvature is constant everywhere, the sectional curvature can be estimated easily by squaring the radius of curvature if the geodesic distances on the manifold and the Euclidean distances in the Euclidean space are known. For a curved manifold, the geodesic distance between two points is the length of the shortest curve connecting the two points. Take the elliptic space for example, the geodesic distance between two points is the length of arc of the great circle which joins the two points on the hypersphere illustrated in Figure 4.2. Let the radius of the hypersphere (also called radius of the curvature) be r and $\theta_{u,v}$ be the angle subtended by two points at the center of the hypersphere, then the distance between them is:

$$d_G(u,v) = 2r\theta_{u,v} \tag{4.22}$$

And the Euclidean distance is given by:

$$d_E(u,v) = 2r\sin\theta = 2r\sin\frac{d_G(u,v)}{2r} \tag{4.23}$$

This is exactly equivalent to the equation for constant curvature Riemannian scaling in elliptic space by Lindman and Caelli [75]. Lindman and Caelli give the relationship between the two distances on elliptic, hyperbolic and Euclidean constant curvature manifolds as

$$d_E = \begin{cases} \frac{2}{K^{\frac{1}{2}}} \sin(\frac{K^{\frac{1}{2}}}{2} d_G) & \text{Elliptic,} \\ \frac{2}{|K|^{\frac{1}{2}}} \sinh(\frac{|K|^{\frac{1}{2}}}{2} d_G) & \text{Hyperbolic,} \\ d_G & \text{Euclidean.} \end{cases} \quad (4.24)$$

The Euclidean distances are usually approximated by means of the Maclaurin series [92, 33, 127]. The sectional curvature in elliptic space is:

$$K(u,v) = \frac{1}{r^2(u,v)} = \begin{cases} \frac{24(d_G(u,v) - d_E(u,v))}{d_G^3(u,v)} & \text{Elliptic,} \\ -\frac{24(d_G(u,v) - d_E(u,v))}{d_G^3(u,v)} & \text{Hyperbolic.} \end{cases} \quad (4.25)$$

However, the above curvature approximations only hold for small curvatures. In the data under study here, we find that the curvatures for some edges are too large for these approximations to hold. We therefore use it as the initialisation value and estimate curvature from Equation 4.24 using Newton's method.

We use the Newton method to find the sectional curvature in terms of the Euclidean distances and geodesic distances. For elliptic space, the Newton iteration is:

$$K_{m+1} = K_m - \frac{2(d_E - 2K_m^{-\frac{1}{2}} \sin(\frac{1}{2} K_m^{\frac{1}{2}} d_G))}{2K_m^{-\frac{3}{2}} \sin(\frac{1}{2} K_m^{\frac{1}{2}} d_G) - d_G K_m^{-1} \cos(\frac{1}{2} K_m^{\frac{1}{2}} d_G)} \quad (4.26)$$

For hyperbolic space, the Newton iteration is:

$$K_{m+1} = K_m - \frac{2(d_E - 2|K_m|^{-\frac{1}{2}} \sinh(\frac{1}{2}|K_m|^{\frac{1}{2}} d_G))}{2|K_m|^{-\frac{3}{2}} \sinh(\frac{1}{2}|K_m|^{\frac{1}{2}} d_G) - d_G |K_m|^{-1} \cosh(\frac{1}{2}|K_m|^{\frac{1}{2}} d_G)} \quad (4.27)$$

4.4.4 Updating Distances Based on Ricci flow

This is our final step during the Ricci flow piecewise embedding. We can compute new geodesic distances between points lying on a new manifold, which has the updated curvature. We keep the Euclidean distance between the points fixed, while updating the curvature. Based on Equation 4.24, the new geodesic distance under the new sectional curvature can be represented in terms of the Euclidean distances in the ambient space and the new sectional curvatures. The update equation for the geodesic distance is:

$$d_{G_{i+1}} = \begin{cases} \frac{2}{K_{i+1}^{\frac{1}{2}}} \sin^{-1}\left(\frac{K_{i+1}^{\frac{1}{2}}}{2} d_{E_i}\right) & \text{elliptic hypersphere} \\ \frac{2}{|K_{i+1}|^{\frac{1}{2}}} \sinh^{-1}\left(\frac{|K_{i+1}|^{\frac{1}{2}}}{2} d_{E_i}\right) & \text{hyperbolic space} \end{cases} \tag{4.28}$$

These above equations can be applied to each element of the dissimilarity matrix in turn.

4.4.5 The Algorithm

We can transform the curved manifold by updating the sectional curvatures with small time steps and compute the new geodesic distances on the less curved manifold with the Euclidean distances fixed both before and after deforming the manifold. After this process, the geodesic distances come closer to Euclidean distances as the sectional curvatures move to zero. After a number of iterations, the geodesic distances get equal to the Euclidean distances and the space where the objects reside on are smoothed to be Euclidean space from original non-Euclidean space. Figure 4.3 shows the algorithmic steps.

The idea underpinning the algorithm is to embed objects represented by the non-Euclidean dissimilarity on a curved manifold and apply the Ricci flow process to transform the manifold to a flat Euclidean space and update the geodesic distances on the manifold during manifold evolution. The novel contribution here is to apply Ricci flow on non-Euclidean dissimilarities and develop a way to update the geodesic distances dur-

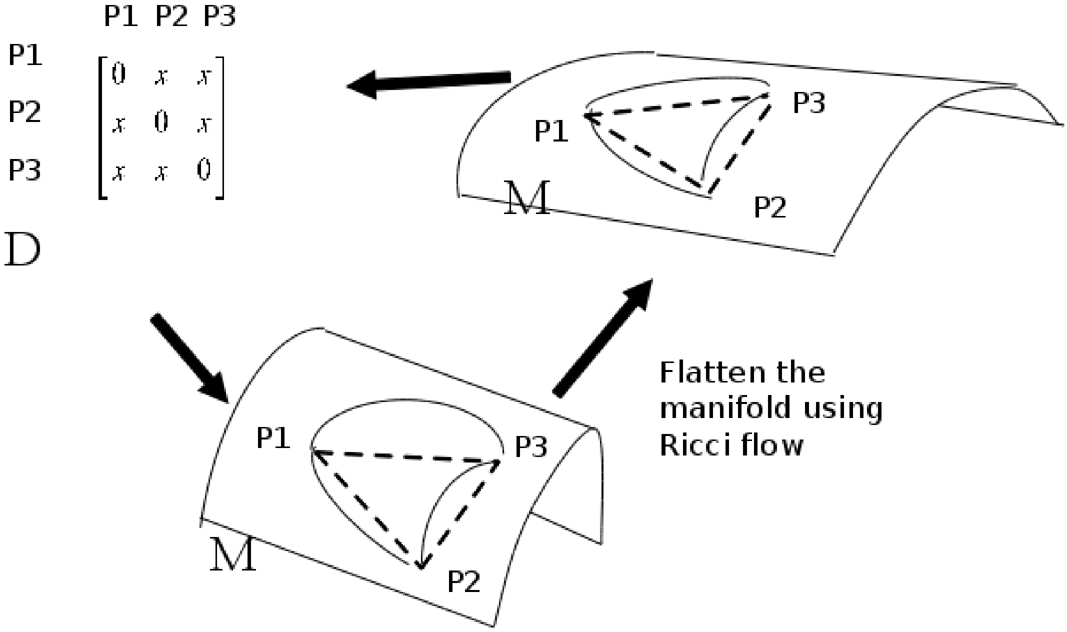

P1 P2 P3
P1
P2
P3
$$\begin{bmatrix} 0 & x & x \\ x & 0 & x \\ x & x & 0 \end{bmatrix}$$
D
P1
P3
M
P2
Flatten the manifold using Ricci flow
P1
P3
M
P2

FIGURE 4.3: Illustration of the piecewise Ricci flow embedding.

ing the manifold evolution. Thus this rectifies a given set of non-Euclidean dissimilarity data so as to make them more Euclidean.

Given a set of N objects and a dissimilarity measure d, a dissimilarity representation is an $N \times N$ matrix D_G, with the elements $d_G(u, v)$ representing the pairwise geodesic distance between objects u and v. The following implementation steps shows how to rectify the distance matrix from being non-Euclidean to Euclidean.

Begin with a $N \times N$ pairwise distance matrix D_G, the iteration number $i = 1$,

1. Embed the objects in a Euclidean space using either Isomap or the kernel embedding to obtain Euclidean distances d_{E_i}.

2. From geodesic distance d_{G_i} and Euclidean distance d_{E_i}, find the constant curvature space with curvature K_i for a pair of objects using Newtons method iteratively until the change is smaller than $1e - 5$ based on Equation 4.26 or Equation 4.27

3. Obtain new geodesic distance $d_{G_{i+1}}$ from previous geodesic distance and curvatures with fixed Euclidean distance based on Equation 4.28.

4. Get the new distance matrix $D_{G_{i+1}}$ composed of new geodesic distances between objects, and repeat from step 1 until $D_{G_{i+1}}$ is Euclidean, that is, there is no negative eigenvalues from its centered Gram matrix.

This method uses Ricci flow on a constant curvature Riemannian manifold to evolve the distance measures. This is implemented by updating the curvatures on the edges of the graph representing the data. Since we consider each edge as individual patch with constant curvature, our current approach updates curvature independently on each edge and ignores the relation among connected edges in the neighbourhood. This method can prove experimentally unstable due to local fluctuations in edge curvatures. To overcome this problem, the next section shows how to stabilise this method by regularising the curvature of the embedded graph.

4.5 Regularised Ricci Flow Embedding

As posed above, the piecewise Ricci flow embedding updates the sectional curvature separately for each individual edge. This places no constraint on the smoothness of the manifold, and this fact can lead to numerical instability in the embedding. Graph regularization provides a way to smooth data samples over a graph and overcome the numerical stability problems. One such regularization process is a graph diffusion. This section shows how the sectional curvature is smoothed out by the diffusion kernel for stabilising the method.

4.5.1 The Laplacian on Graph

Before we introduce the diffusion kernel in the next section, we first review the basic concepts about graphs and the Laplacian on a graph that are necessary to define our curvature regularisation process. Firstly, suppose that an undirected unweighted graph is denoted by $G = (V, E)$ where V is the set of nodes and $E \subseteq V \times V$ is the set of edges. The elements of the adjacency matrix A for the graph are:

$$A(u,v) = \begin{cases} 1 & \text{if } (u,v) \in E \\ 0 & \text{otherwise} \end{cases} \tag{4.29}$$

From the adjacency matrix we construct the diagonal degree matrix D, whose elements are defined as the degree of the nodes :

$$D(u,v) = \begin{cases} \deg(u) = \sum_{v \in V} A(u,v) & \text{if } u = v \\ 0 & \text{otherwise} \end{cases} \tag{4.30}$$

From the degree matrix and the adjacency matrix, we construct the Laplacian matrix $L = D - A$, which is the degree matrix minus the adjacency matrix. Thus the elements of the Laplacian matrix are defined as:

$$L(u,v) = \begin{cases} \deg(u) & \text{if } u = v \\ -1 & \text{if } u \text{ and } v \text{ are aadjacent} \\ 0 & \text{otherwise} \end{cases} \tag{4.31}$$

The elements of the normalized Laplacian $\hat{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ are defined as

$$\hat{L}(u,v) = \begin{cases} 1 & \text{if } u = v \text{ and } \deg(v) \neq 0 \\ -\frac{1}{\sqrt{\deg(u)\deg(v)}} & \text{if } u \text{ and } v \text{ are aadjacent} \\ 0 & \text{otherwise} \end{cases} \tag{4.32}$$

The spectral decomposition of the normalised Lapalcain matrix is $\hat{L} = \Phi \Lambda \Phi^T$, where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{|V|})$ is the diagonal matrix with the ordered eigenvalues $(\lambda_1 < \lambda_2 < \cdots < \lambda_{|V|})$ as elements and $\Phi = (\phi_1 | \phi_2 | \cdots | \phi_{|V|})$ is the matrix with the ordered eigenvectors as columns. Recently spectral graph theory has been applied to graph clustering, as the the spectral graph methods use only part of the information from the spectrum (eigenvalues of the spectral decomposition) of the Laplacian matrix and avoid the nodes correspondence problems [15]. Since $\hat{L}$ is symmetric and positive semi-definite, the eigenvalues of the normalized Laplacian are all non-negative. The number of zero eigenvalues is the number of isolated components in the graph. Hence for a connected graph, there is only one eigenvalue equal to zero. von Luxburg et al. [118] show that the normalised Laplacian leads to more robust semi definite Laplacian. Hence, we use the normalised Laplacian matrix for calculating the heat kernel.

4.5.2 Curvature Regularisation Using Heat Kernel

We are interested in the heat kernel which is the fundamental solution of the heat equation. The heat equation associated with the Laplacian is defined as

$$\frac{\partial H(t)}{\partial t} = -\hat{L}H(t) \tag{4.33}$$

where Ht is the heat kernel and t is time. It is a diffusion process and is analogous to the flow of heat, which flows from high to low concentrations, and over time creates a smooth distribution of heat. In a similar way, a diffusion of a function on the graph will create a smoother function. The diffusion is equivalent to a random walk on the edges of the graph [69], and is represented by the diffusion (or heat) kernel:

$$H(t) = \exp(-\hat{L}t) \tag{4.34}$$

where $\exp(-\hat{L}t)$ is the exponential of the matrix, $-\hat{L}t$. To compute the heat kernel, we use the eigen-decomposition of the Laplacian $\hat{L}$. Based on [15] we can proceed to compute the heat kernel on a graph by exponentiating the Laplacian eigenspectrum:

$$H(t) = \sum_{i=1}^{|V|} \exp(-\lambda_i t)\phi_i\phi_i^T = \Phi\exp(-\Lambda t)\Phi^T \tag{4.35}$$

Xiao et al [128] shows that the heat kernel depends on the local connectivity structure of the graph when t tend to zero, $H_t \simeq I - \hat{L}t$ where I is the identity matrix; the heat kernel is controlled by the global structure of the graph when t is large, $H_t \simeq I - \exp(-\lambda_2 t)\phi_2\phi_2^T$ where λ_2 is the smallest non-zero eigenvalue and ϕ_2 is the associated eigenvector.

The exponential of a matrix M is defined by the power series:

$$\exp M = \sum_{k=0}^{\infty} \frac{1}{k!}M^k \tag{4.36}$$

If a matrix is diagonal, then its exponential can be obtained by just exponentiating every entry on the main diagonal. Hence we only need to exponentiate the eigenvalues of the normalised Laplacian matrix for computing the heat kernel. The evolution of a function under this kernel is simply

$$f(t) = H(t)f(0) \tag{4.37}$$

The evolution is 'mass-preserving' in the sense that the sum of the values of the function over vertices is preserved. We can use this process for smoothing curvatures before the application of the Ricci flow, to remove extreme values.

However, our curvatures are defined on pairs of objects and heat kernel diffusion only works on vertices. Therefore we need to build a graph which has vertices corresponding to object-pairs and edges describing a neighbourhood structure of these pairs. We construct this graph as follows. Firstly, we build the nearest-neighbours graph of the objects $G = \{V, E\}$. Each vertex u represents an object; and an undirected edge E_{uv} exists if u is in the k nearest neighbours of object v or if v is in the k nearest neighbours of u. We then construct its dual graph $G_D = \{V_D, E_D\}$; each edge of the original graph becomes a vertex V_{uv} and an edges exist between two vertices if they share a common vertex from the original graph. In the dual graph, each vertex represents a pair of objects and the edges reflect the neighbourhood structure of the pairs. We can then define the curvature between object pairs as a function over the vertices of this dual graph and apply the diffusion kernel.

We therefore add an additional step in which we smooth the sectional curvatures over the dual of the nearest neighbour graph prior to performing the Ricci flow for updating the sectional curvatures. All of the remaining steps of the algorithm remain as the piecewise Ricci flow embedding in Section 4.4.5.

4.5.3 The Algorithm

The following steps shows how to smooth sectional curvatures over the nearest neighbour edges.

Start from the initial sectional curvatures K from step 2 of the piecewise Ricci flow embedding in Section 4.4.5,

1. Construct the k nearest neighbour graph over the available dissimilarity data. Node u and v are connected by an edge if u is among k nearest dissimilarity neighbors of v or v is among k nearest dissimilarity neighbors of u.

2. Construct the dual graph of the nearest neighbour graph. Each edge in the nearest neighbour graph is a vertex of the dual graph. If two edges in the nearest neighbour graph share a one common vertex, then the corresponding two vertices in the dual graph are connected by an edge.

3. Obtain the updated and regularised curvature K. Suppose that $\hat{L}$ is the normalised Laplacian of the dual nearest neighbour graph, then the heat-kernel of the dual graph is $\exp[-\hat{L}t]$. If V_D is the node-set of the dual graph, then we construct a vector **K** of Gaussian curvatures $\mathbf{K} = (K_1, ..., K_{|V_D|})^T$. The vector of regularised Gaussian curvatures after heat kernel smoothing is $\mathbf{K}_{\text{reg}} = \exp[-\hat{L}t]\mathbf{K}$. Since our objective is to preserve the local structure of the dual graph, in the experiment we choose small value for t from a set of values $10.0, 1.0, 0.1, 0.01$. We experimentally found that the smoothing does not have much effect when we use value of t which is greater than 1.0, so we use 0.1.

In summary, the above approach starts from a nearest neighbour graph over the the dissimilarity matrix, and then constructs the dual graph where a node corresponds to an edge in the original graph. The heat kernel on the dual graph smooths the curvatures on the original nearest neighbour graph.

4.6 Experiments on Ricci Flow Embedding

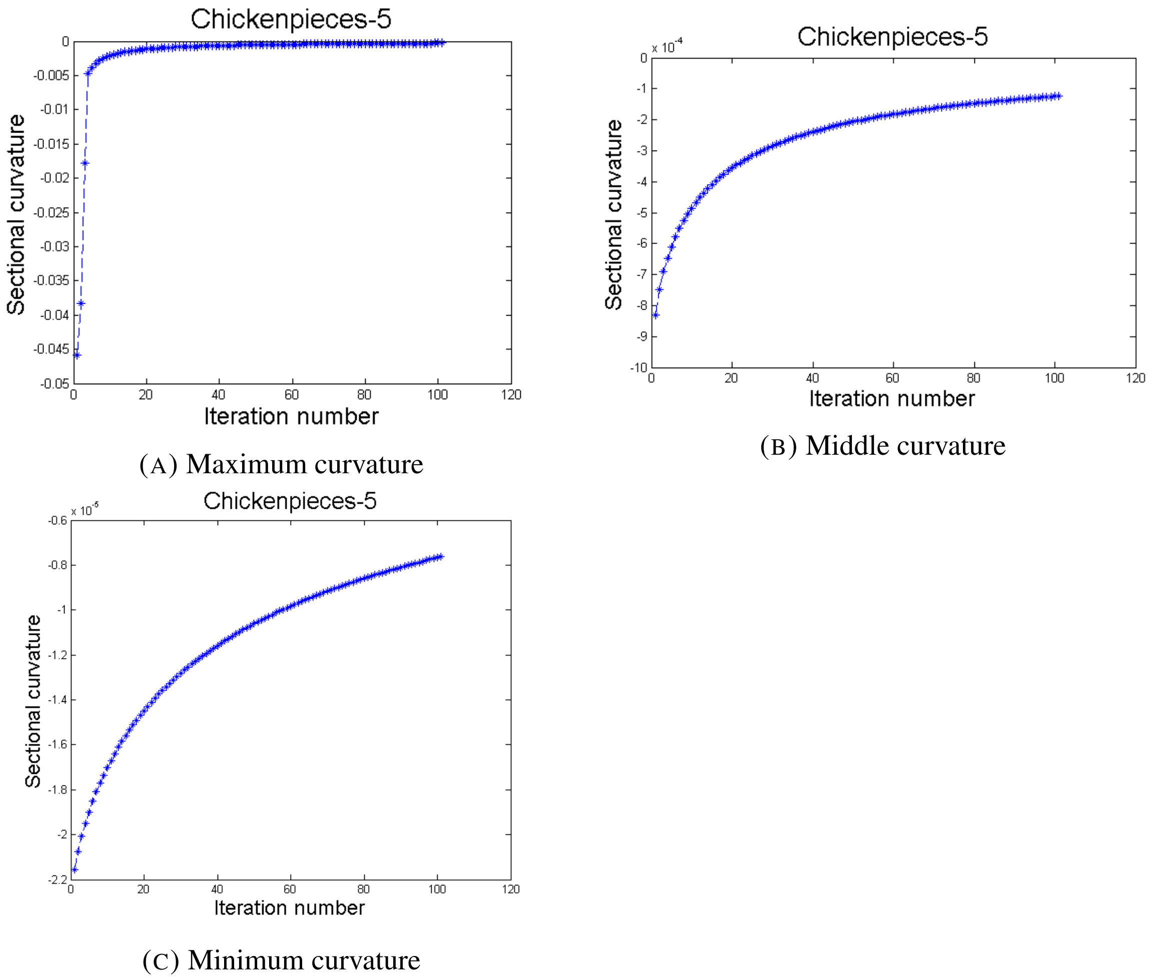
In the previous section, we introduce the implementation details for the Ricci flow on constant curvature of either elliptic and hyperbolic background geometry for correcting non-Euclidean dissimilarities. We demonstrate the applicability of this intrinsic curvature flow method though dissimilarity based learning problems, especially non-Euclidean dissimilarities. In this section, we applied our two Ricci flow embedding techniques to two different datasets: Chickenpieces and CoilYork. These datasets give rise to non-metric or non-Euclidean dissimilarities. They are chosen from a variety of datasets we have analysed in Chapter 3. Considering the Chickenpieces dataset, the 1NN classification errors of the eight dissimilarity data are lower than those of the eight corresponding positive subspace dissimilarity illustrated in Table 3.1 in Chapter 3. The table shows the negative eigenvalues of the dissimilarities are informative for classification. Considering the CoilYork dataset, the negative eigenfraction (NEF) is 0.2576, the classification error rate of the original dissimilarity is 0.2326, which is lower than 0.3368, the classification error of the positive subspace dissimilarity. This shows the negative eigenvalues of the dissimilarity is informative for classification. Since the negative eigenvalues from these two datasets are informative for classification, these datasets can be used to demonstrate whether the embedding techniques preserve the grouping structure of data contained in the negative eigenvalues. These are the Chickenpieces data and the CoilYork data.

The objective in this thesis is to rectify the non-Euclidean dissimilarity data into a set of Euclidean distances. The results of these experiments for the two Ricci flow embedding techniques fall into two main categories and will be discussed in turn. The first is the negative eigenfraction of the dissimilarity data during the Ricci flow process. Then negative eigenfraction is zero when the distances are Euclidean and increases as the distance becomes increasingly non-Euclidean. In fact, rectifying non-Euclidean dissimilarity data into a set of Euclidean distances is equivalent to decrease the negative eigenfraction so that it moves toward zero finally. The second is the 1NN classification error rate of the corrected dissimilarity, relative to the original non-Euclidean dissimilarity and to other Euclidean embedding. We use the average of leave-one-out-cross-validation error rates by applying the 1NN classifier to measure the classification accuracy on the data. The 1NN classifier is chosen to give a comparison because it is one of very few classifiers which can perform directly on generally any dissimilarity without any preprocessing step. Firstly, for analytic purposes we show results during the embedding iterations. Secondly, for comparative purpose, we also show results against three Euclidean embeddings: the positive subspace which is the kernel embedding discarding negative eigenvalues, the associative Euclidean embedding which is the kernel embedding where negative eigenvalues are taken their absolute values, the Isomap embedding which is kernel embedding on geodesic distances of neighbourhood graph. Please note that we use the optimal parameter of the size of neighbouhood k and the dimensionality d, at which the curve of the residual variance stops to decrease significantly. The residual variance is used to evaluate the fits of Isomap, which is defined as one minus the linear correlation coefficient taken over all entries of resulting dissimilarity and original dissimilarity [110]. For CoilYork data, $k = 8, d = 10$. For the chickenpieces data, $k = 20, d = 20$.

4.6.1 Experiments on Piecewise Ricci Flow Embedding

The sectional curvatures for the edge with initial biggest curvature, the edge with median value of curvature and the edge with minimum curvature on the Chickenpieces data with $L = 5$, as the manifold evolves using the Ricci flow with kernel embedding, are shown in Figure 4.4. Each of the curvatures moves towards zero, indicating that the evolution process transforms the hyperbolic space (negative sectional curvature) to a Euclidean space (zero sectional curvature).

Figure 4.5 shows the curvatures as a function of distances obtained using the kernel embedding and the Isomap embedding for the randomly chosen 500 edges from the chickenpieces $L = 15$ over a single iteration. It demonstrates how the piecewise Ricci

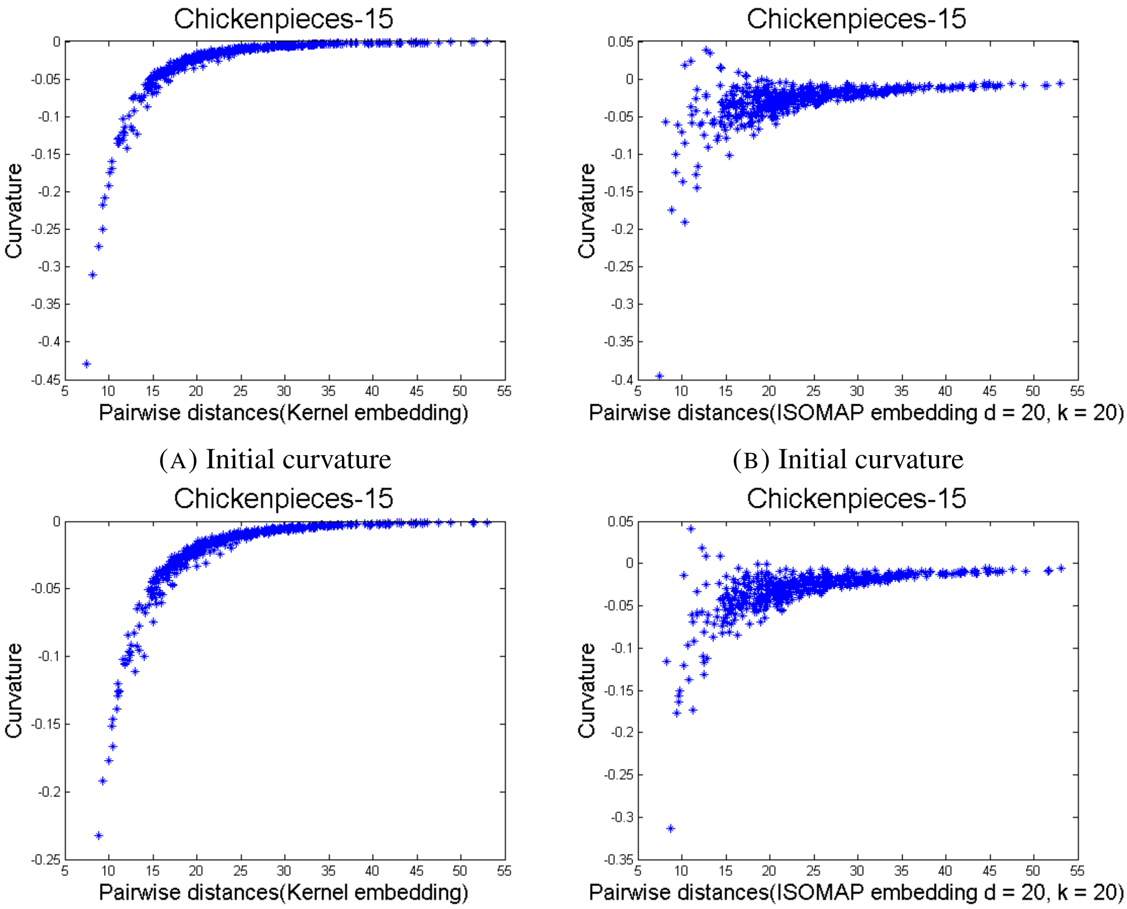


(A) Maximum curvature

(B) Middle curvature

(C) Minimum curvature

FIGURE 4.4: (A)(B)(C) show the individual edge's sectional curvatures during the piecewise Ricci flow with kernel embedding. All the sectional curvatures move to zero.



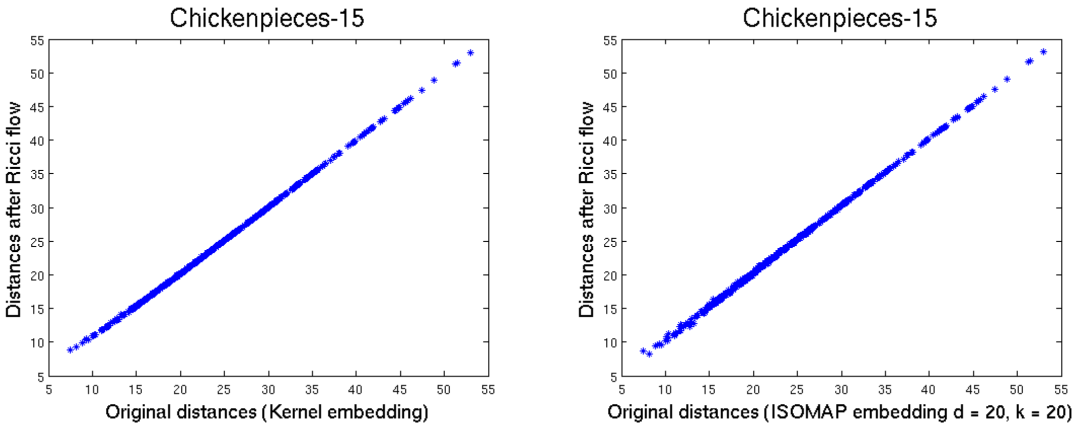
(A) Initial curvature

(B) Initial curvature

(C) Curvature after Ricci fow

(D) Curvature after Ricci flow

FIGURE 4.5: (A)(B) show the initial curvatures of 500 randomly selected edges from kernel embedding and Isomap embedding. (C)(D)show the edge curvatures after the piecewise Ricci flow with kernel embedding and the piecewise Ricci flow with Isomap embedding.



(A) Kernel embedding

(B) Isomap embedding

FIGURE 4.6: (A)(B) show the distances of 500 randomly selected edges before and after a single iteration of Ricci flow embedding with kernel embedding and Ricci flow with Isomap embedding.

flow process affects distances commencing from the two embedding methods. Figure 4.6 shows the original distances and updated distances obtained using Ricci flow with kernel embedding and Ricci flow with Isomap embedding for the randomly chosen 500 edges from the chickenpieces $L = 15$ over a single iteration. Both the resulting distances appear almost linear to the distance before embedding, it shows these two embedding methods preserve distances, especially the global distances because of the effects of kernel embedding. The change of the local distances demonstrates the difference of these embedding methods. Figure 4.6(A) demonstrates that the local distances expand faster than the global distances under kernel embedding, while Figure 4.6(B) shows under Isomap embedding the local distances are preserved better than that under kernel embedding.

Specifically, we choose two pairwise distances. The distance with a larger value is represented the distance of the distant points, and the smaller distance for the local points. Please note these two distances represent only part of the dissimilarity data. The two distances are selected to find out how the distant point and local points evolve during the iteration. Two pairwise distance with value 7.44 and 52.94 are used here. Firstly, the initial curvatures of the patches consisting of an edge is compared. Under the kernel embedding, the curvature of the distant points is -11.26×10^{-4}, which has much smaller magnitude than that of the local points, whose curvature is -0.43. Figure 4.5 shows that the kernel embedding preserves the global distances. This shows that the larger are the distances, the smaller is the magnitude of the curvatures. From our Ricci flow curvature updating process, the larger the magnitude of the original curvatures, the larger the curvature reduction in the updated process. That is, the distant edges are less curved than the local edge. Moreover, during the Ricci flow process, the stronger the magnitude of the negative curvature is, the faster the expanding of the distances. As a result, the local edges increases more rapidly than those distant edges. For example, the local edge is expanded from 7.44 to 9.34, while the distant edge is slightly expanded from 52.94 to 52.95. In terms of distances, the locations with some smaller distances expand more rapidly than those with larger distances demonstrated in Figure 4.6, which is caused by the combined effects of Ricci flow and the kernel embedding. This effect can be observed from Figure 4.5 (A) and 4.5 (C). As a result, the local pattern of distances will be destroyed after a few iterations of the Ricci flow with kernel embedding.

In the case of Isomap embedding, the curvature of the distant points is -0.0061, which has much smaller magnitude than that of the local points, whose edge is -0.39. After the Ricci flow process, the local edge is increased from 7.44 to 8.28, while the distant edge is slightly increased from 52.94 to 53.11. Although the expansion is smaller those obtained with the Ricci flow with kernel embedding, the Ricci flow with Isomap embedding exhibits similar patterns to the Ricci flow with kernel embedding. The Isomap embedding is equivalent to the kernel embedding on geodesic distances of neighbourhood graph. In some sense, the Isomap embedding preserves global patterns of distances as the kernel embedding. Hence, the locations associated with initial negative eigenvalues are evolved as if they are affected by the Ricci flow with kernel embedding.

As some positive curvatures are obtained with Isomap embedding, we also need to investigate how the piecewise Ricci flow affects distances of edges with positive curvatures.

Here we also choose two pairwise distances. One is 12.81 for slightly closer points, and the other is 17.28 for more distant points. The initial curvature for closer points is 0.039 which is larger than 0.0052, the curvature for distant points. That is, the locations associated with distant points are less curved than that of the local edge. During the Ricci flow process, the stronger the positive curvature, the faster the contraction of the distances. As a result, the initial closer point get more closer and the local edges with positive curvatures contract more rapidly that of distant points. In other words, the locations associated with smaller distances contract more rapidly than those locations associated with bigger distances. For example, he distance between the distant points is reduced from 17.28 to 17.27, while the distance between the local points is reduced from 12.81 to 12.24. This effect can be observed from Figure 4.5 (B) and 4.5 (D). Overall, the Ricci flow with Isomap embedding preserves the ranking of most local distances with positive curvatures. Compared to the Ricci flow with kernel embedding, the evolution of distances under the Ricci flow with Isomap embedding is more complex, since it has both ecliptic and hyperbolic geometry. And it preserves some of the local patterns and global patterns of distances. In other words, the Ricci flow with Isomap embedding preserves some of the local distances while disrupting some from the effects of the kernel embedding.

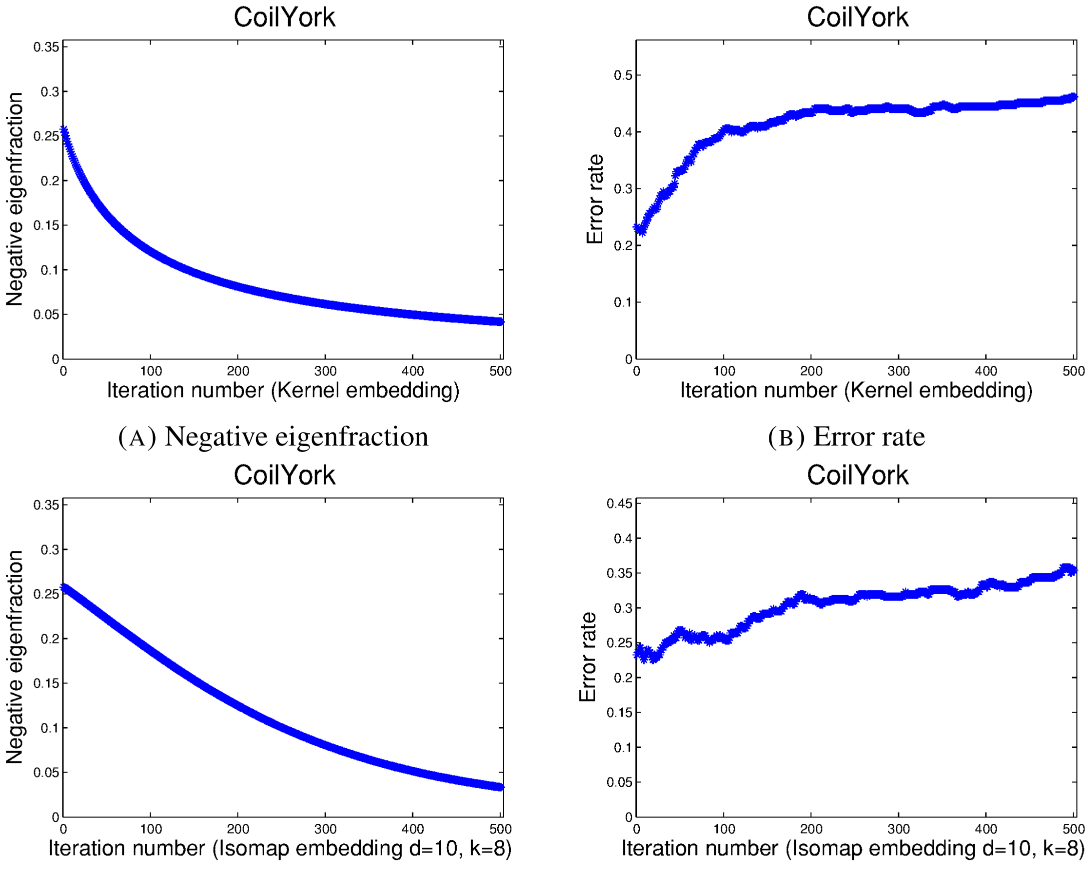
Figure 4.5 indicates that the embedding methods affects the magnitude of the curvatures. The magnitude of the curvatures affects how the distance is updated during Ricci flow. The sign of the curvature (hyperoblic geometry or elliptic geomtry) determines the direction for transforming distances, expanding for hyperbolic geometry and contracting for elliptic geometry.

Dataset	NEF_0 (%)	NEF_1 (%)	NEF_2 (%)	$Error_0$ (%)	$Error_1$ (%)	$Error_2$ %)
Chickenpieces-5	21.6	4.61	3.4	34.53	51.59	44.25
Chickenpieces-10	25.7	6.28	2.54	16.14	48.17	35.42
Chickenpieces-15	28.6	4.44	3.66	7.40	44.81	21.44
Chickenpieces-20	30.7	2.91	1.57	6.28	41.30	22.73
Chickenpieces-25	31.99	4.95	0.56	4.26	36.95	18.80
Chickenpieces-30	33.07	3.77	0.25	4.48	35.56	15.35
Chickenpieces-35	33.94	5.50	1.86	6.28	36.32	16.37
Chickenpieces-40	34.46	4.50	0.38	8.74	32.29	20.18
CoilYork	25.76	4.61	4.63	23.26	46.15	25.31

TABLE 4.1: Negative eigenvalues and classification errors of the original dissimilarity(NEF_0, $Error_0$), of the final dissimilarity of the piecewise (unregularized) Ricci flow with kernel (NEF_1, $Error_1$) embedding, and the piecewise (unregularized) Ricci flow with Isomap embedding NEF_2, $Error_2$).

Next we turn our attention to the choice of embedding on negative eigenfractions of rectified distances and results of classification. From Figure 4.7 to Figure 4.15, we show the negative eigenfraction and the 1NN error rates obtained with the piecewise Ricci flow on the two embedding schemes for the CoilYork data and the Chickenpieces data as the manifold evolves. In Table 4.1, we compare the 1NN error rate of the final dissimilarities obtain with the piecewise Ricci flow on the two embedding schemes. Error$_0$ refers to the error rate of the original non-Euclidean dissimilarity data, Error$_1$ refers to the error rate of final Euclidean dissimilarity obtained by the Ricci flow with kernel embedding, and Error$_2$ refers to the error rate of final Euclidean dissimilarity obtained by the Ricci flow with Isomap embedding. The first point to note is that for both the kernel embedding and the Isomap embedding, the negative eigenfraction decrease as the curvatures are updated, indicating that the dissimilarity measure becomes increasingly Euclidean. Another point to note is that the choice of embedding scheme strongly affects the rate of decrease of the negative eigenfraction, with Isomap embedding giving a faster rate of decrease with iteration number than the kernel embedding. However, for both embedding schemes, worse classification results are given than the original non-Euclidean dissimilarities. We obtain better classification results with Isomap embedding scheme than those with the kernel embedding. We believe that this is because the positive curvature decreases more rapidly for shorter distances in the Isomap embedding. As a result, the Isomap embedding preserves some of local pattern of distances compared to the kernel embedding during the Ricci flow process.

Finally, some classification experiments were performed to explore whether the evolution preserves the class structure in data. We compared these results with those obtained using some alternative Euclidean correction procedures. The methods explored were a) projecting onto the positive subspace b) projecting onto the associated Euclidean space c) using the Isomap embedding d) using the original distances. Our classification performance results on the chickenpieces data are shown in Figure 4.16. Each of the embedding methods distorts the data to some extent. The Isomap embedding gives the smallest degradation. The Ricci flow with Isomap embedding gives comparable results to the positive space embedding. By comparison, the results from the Ricci flow with kernel embedding is worse than those from the kernel embedding (positive subspaces). We believe that this is because the curvature increases much more rapidly for shorter distances in the kernel embedding. The Ricci flow with kernel embedding destroys local pattern of data. During the Ricci flow with kernel embedding, the Ricci flow is performed on local patches of hyperbolic geometry. On the other hand, during the process of applying the Ricci flow with Isomap embedding to dissimilarity data, Ricci flow is performed on a more complex geometry composed of both local hyperbolic geometry and local elliptic geometry. The Ricci flow performed on the local hyperpbolic space destroys the local structure of data as the Ricci flow with kernel embedding. This also explains why the results from the Ricci flow with Isomap embedding are worse than those from the purely Isomap embedding. Since the curvature increases much more rapidly for shorter distances in the kernel embedding, the change in distance induced by Ricci flow is rapid and potentially unstable. We demonstrate that our correction procedure is able to rectify non-Euclidean



(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

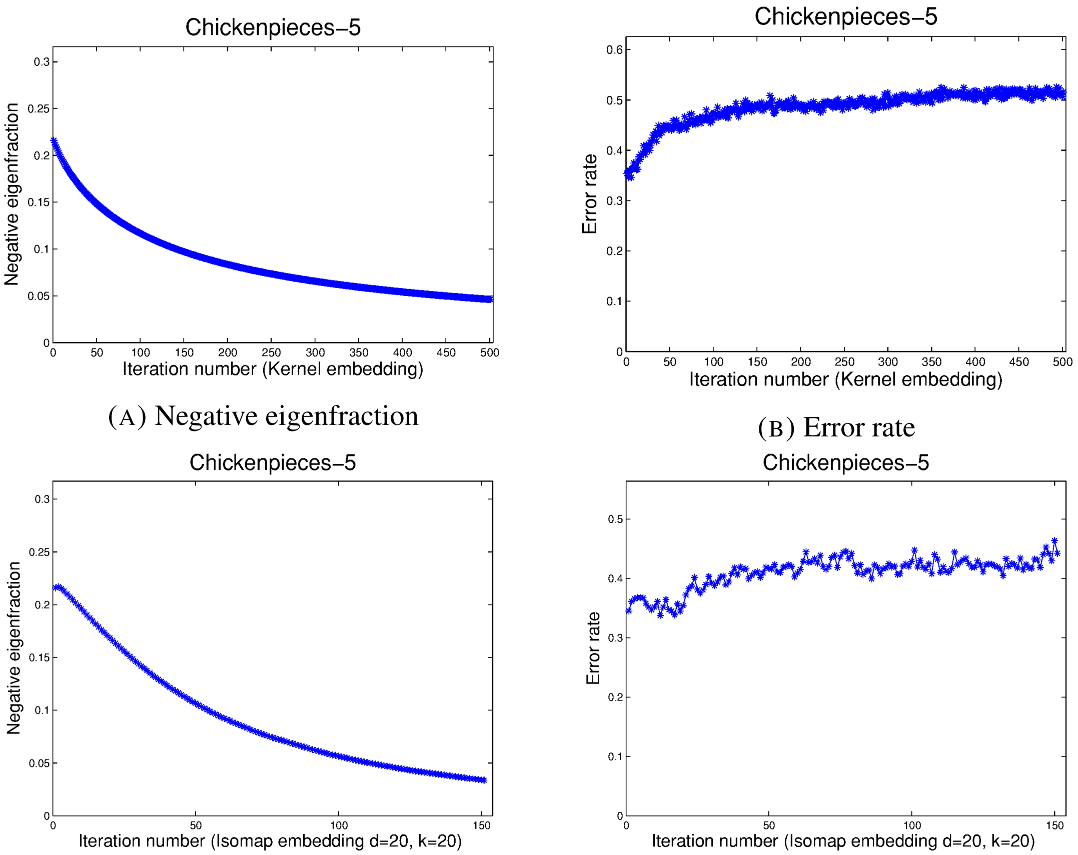
(D) Error rate

FIGURE 4.7: (A)(B) show the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with kernel embedding for the CoilYork dataset. (C)(D) are the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with Isomap embedding for the CoilYork dataset.

dissimilarity, but with some loss of discriminating power.

4.6.2 Experiments on Regularised Ricci Flow Embedding

Figure 4.17 and 4.18 show the curvature as a function of distances obtained using regularised Ricci flow with kernel embedding and Isomap embedding. During the regularisation step, the curvatures are smoothed over nearest neighbour edges. The result is to reduce local curvature fluctuations, and this may reduce some locally large curvature



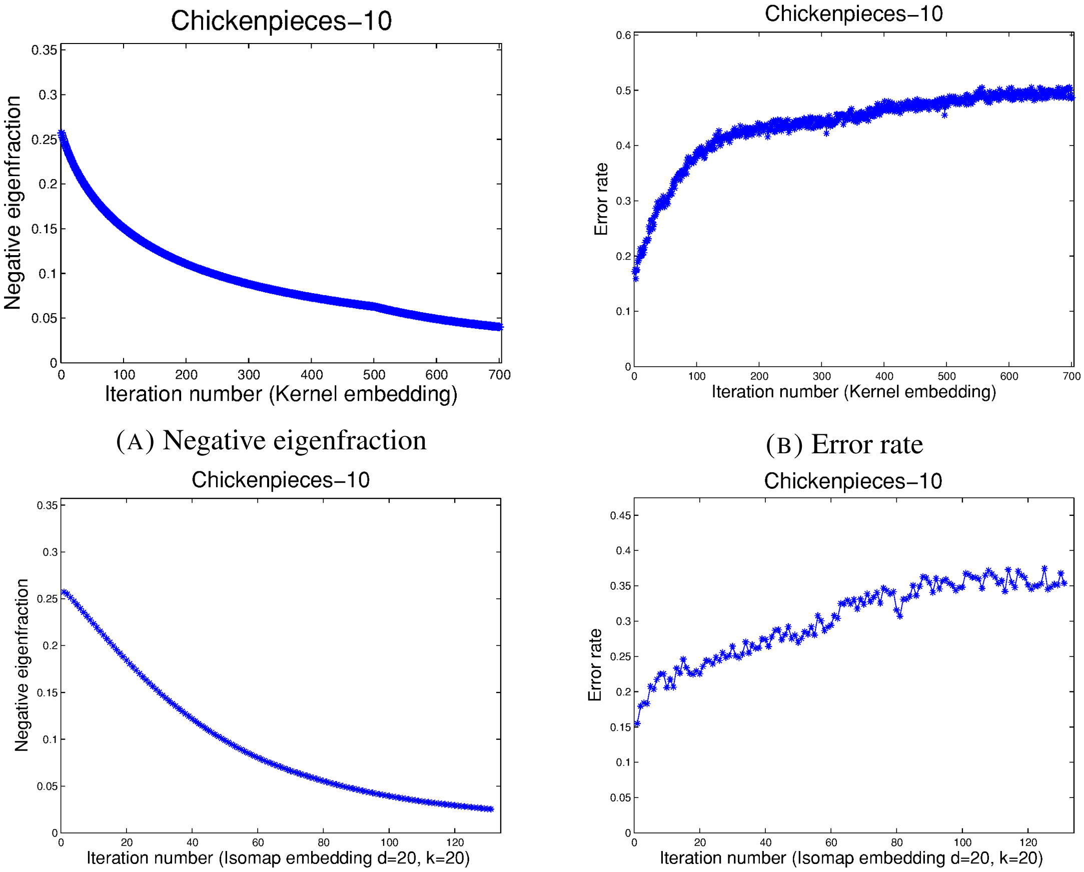
(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.8: (A)(B) show the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with kernel embedding for the Chickenpieces-5 dataset; (C)(D) are the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with Isomap embedding for the Chickenpieces-5 dataset.



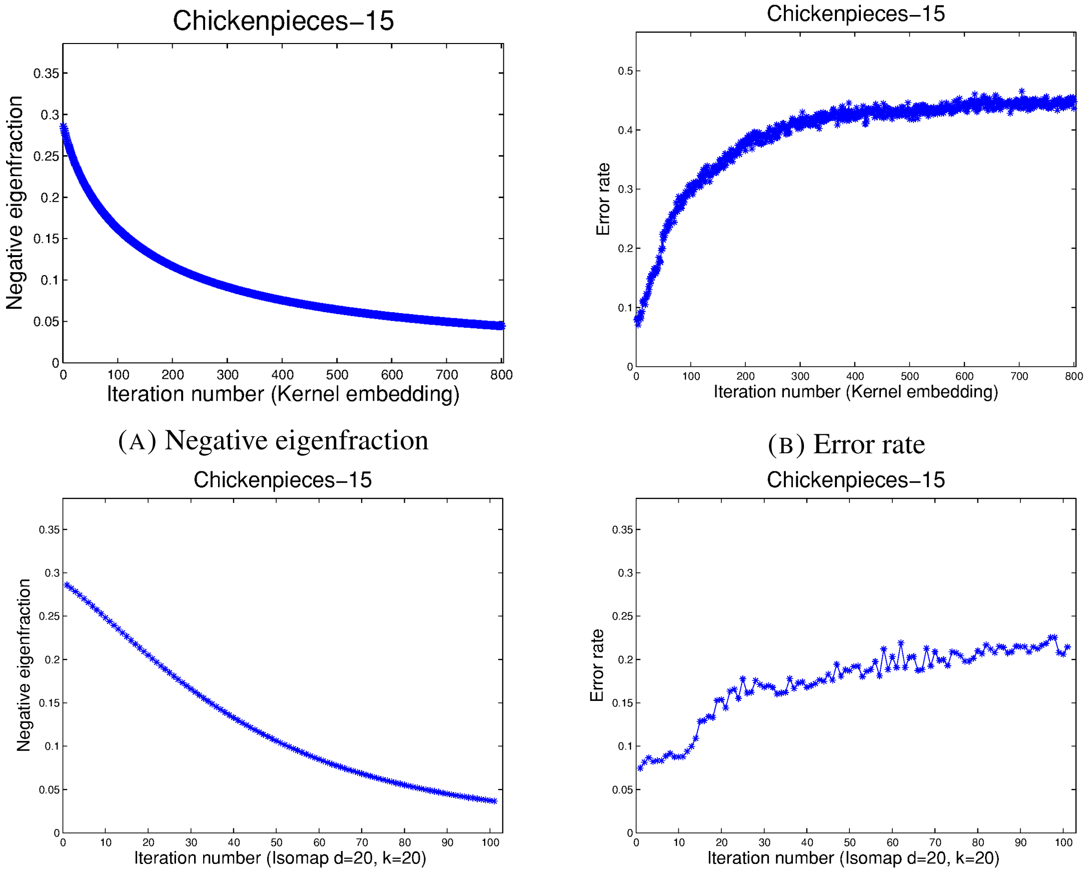
(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.9: (A)(B) show the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with kernel embedding for the Chickenpieces-10 dataset; (C)(D) show the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with Isomap embedding for the Chickenpieces-10 dataset.



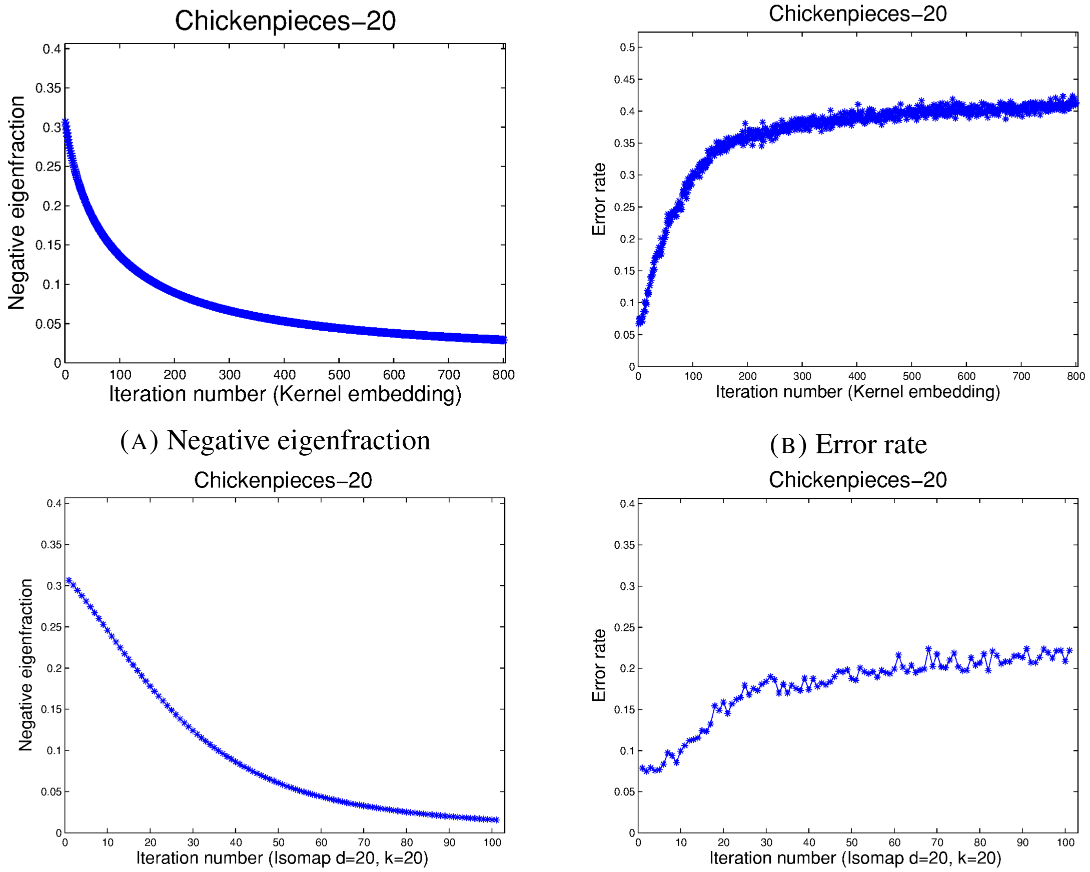
(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.10: (A)(B) show the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with kernel embedding for the Chickenpieces-15 dataset; (C)(D) show the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with Isomap embedding for the Chickenpieces-15 dataset.



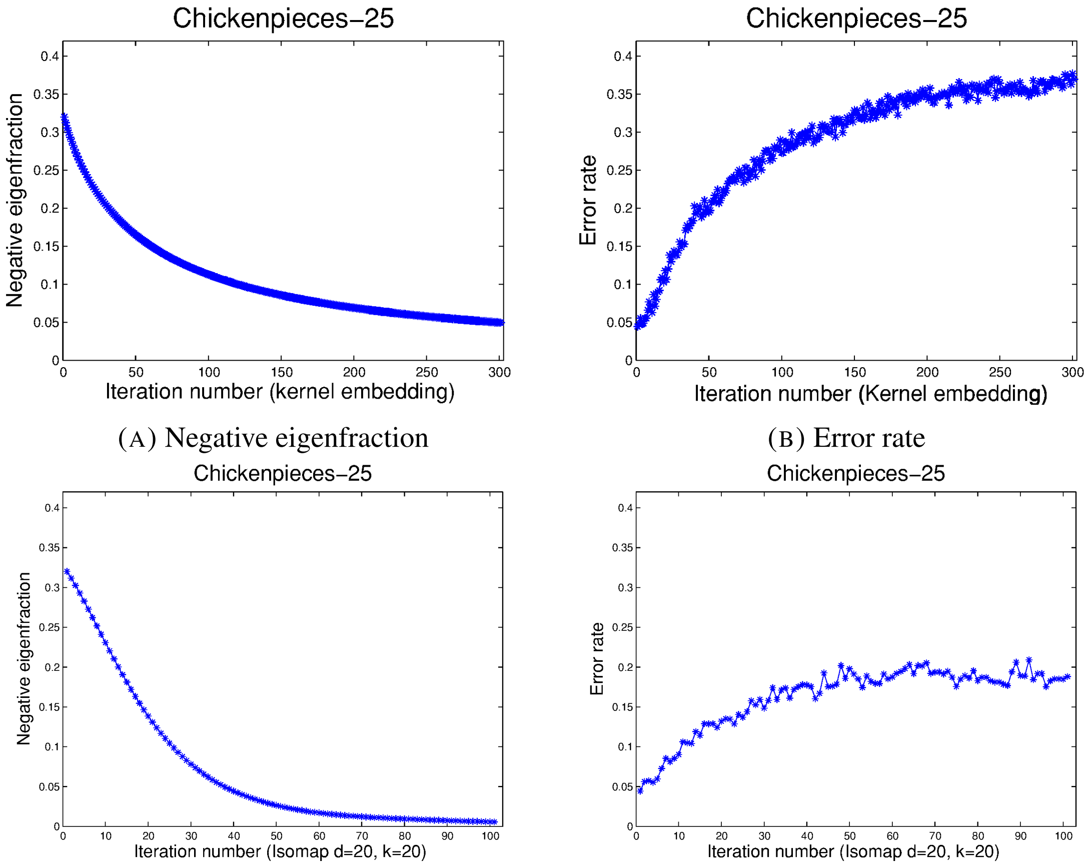
(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.11: (A)(B) show the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with kernel embedding for the Chickenpieces-20 dataset. (C)(D) show the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with Isomap embedding for the Chickenpieces-20 dataset.



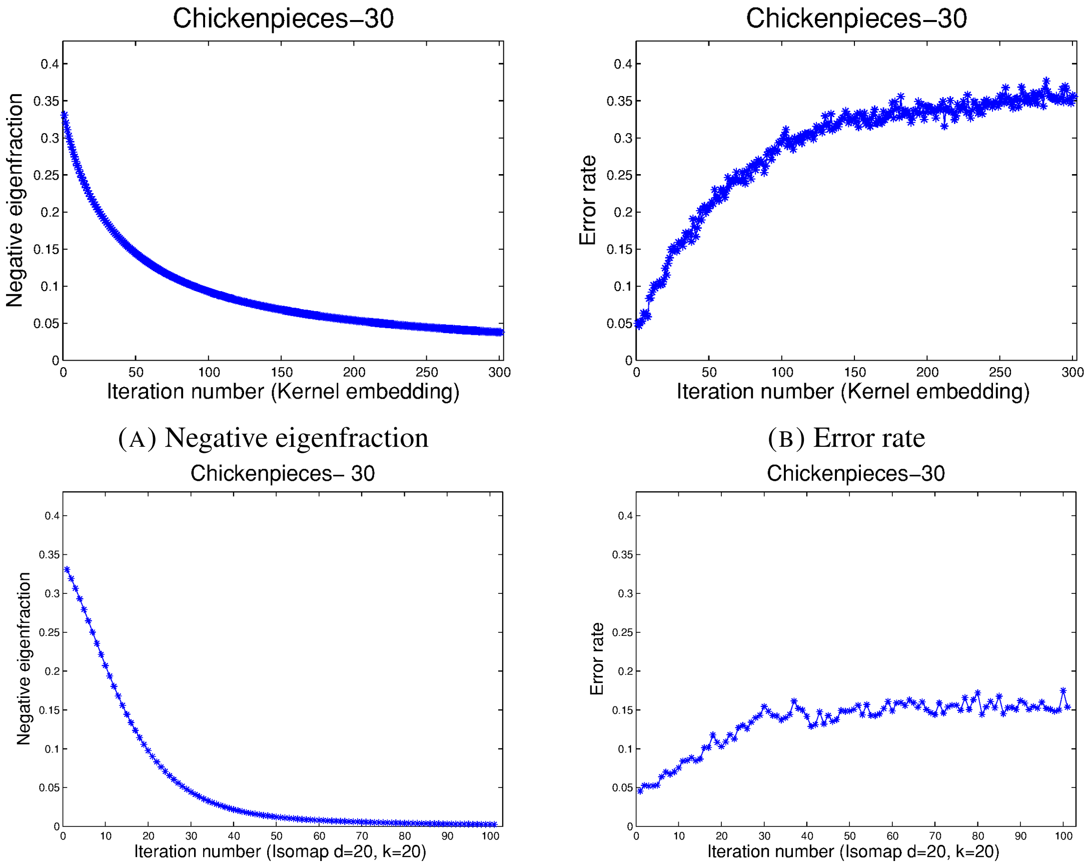
(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.12: (A)(B) show the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with kernel embedding for the Chickenpieces-25 dataset. (C)(D) show the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with Isomap embedding for the Chickenpieces-25 dataset.



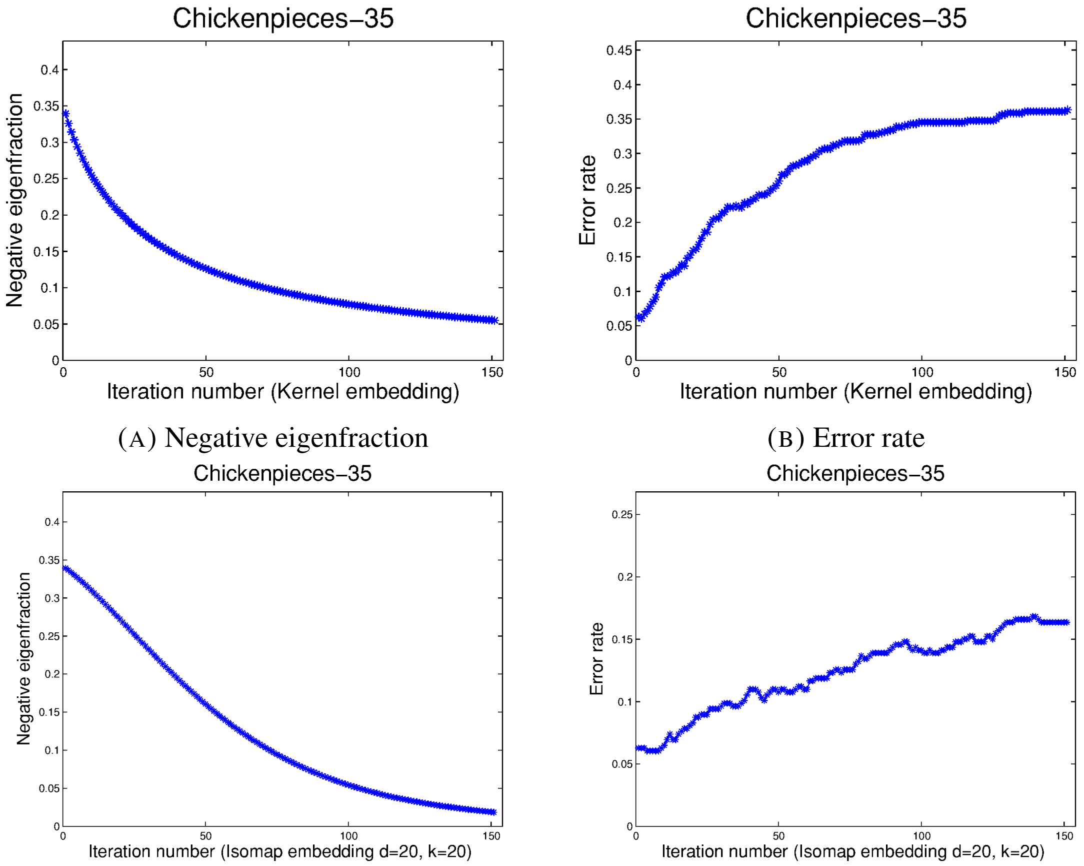
(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.13: (A)(B) show the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with kernel embedding for the Chickenpieces-30 dataset. (C)(D) show the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with Isomap embedding for the Chickenpieces-30 dataset.



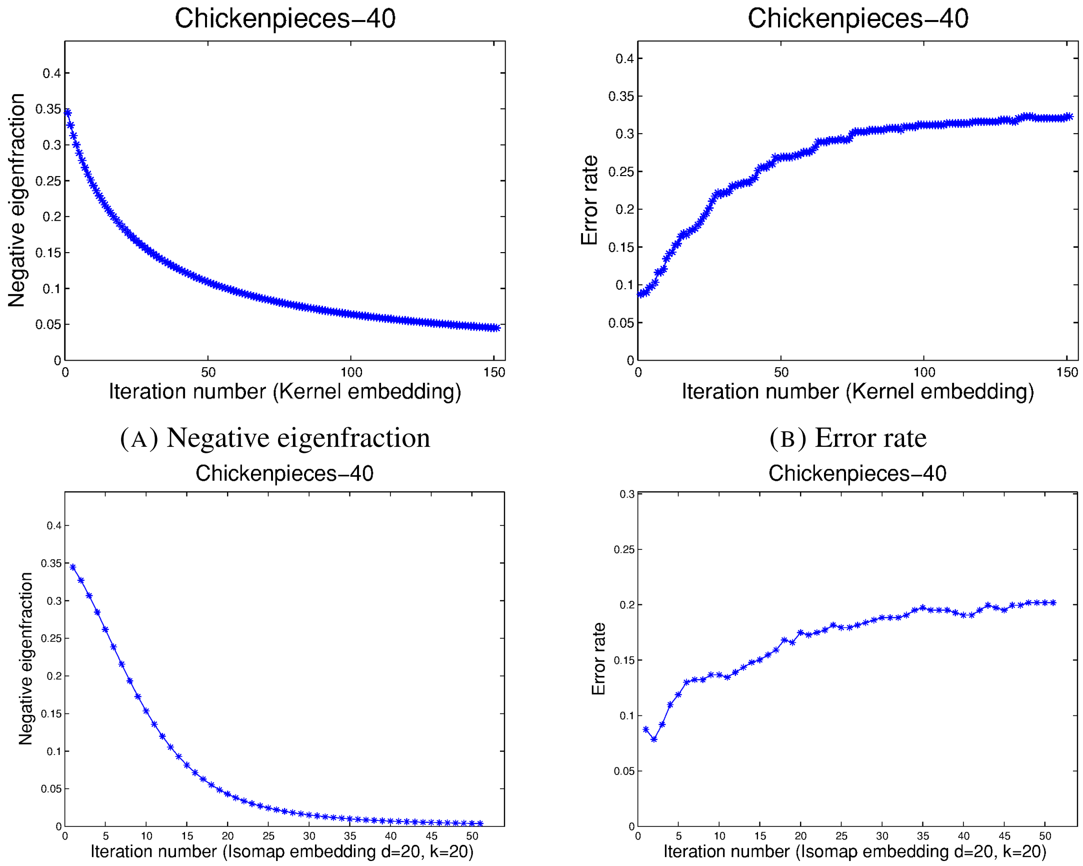
(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.14: (A)(B) show the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with kernel embedding for the Chickenpieces-35 dataset. (C)(D) show the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with Isomap embedding for the Chickenpieces-35 dataset.



(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.15: (A)(B) show the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with kernel embedding for the Chickenpieces-40 dataset. (C)(D) are the negative eigenfraction and 1NN error rate during the iteration of the piecewise Ricci flow with Isomap embedding for the Chickenpieces-40 dataset.

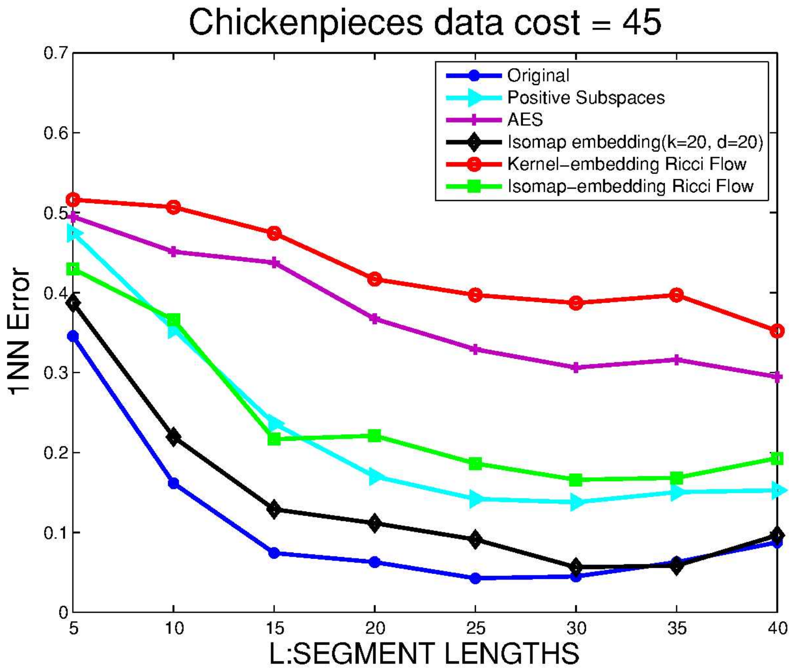


FIGURE 4.16: Error rate from 1NN for Chickenpieces dataset cos=45.

values. Figure 4.17 (B) and Figure 4.18 (B) show that when regularisation is used, the curvatures are smoothed over local distance scales compare to the initial curvature in Figure 4.17 (A) and Figure 4.18 (A). Hence the local distance structure is preserved under the embedding, and this is demonstrated in Figure 4.17 (D) and Figure 4.18 (D). As a result, the regularisation step preserves local distances and stabilises the local structure.

Next, we turn our attention to the effect of regularization and the choice of embedding on results of classification. From Figure 4.20 to Figure 4.28, we compare the negative eigenfraction and the 1NN error rates obtained with regularised and unregularised versions of Ricci flow on the two embedding schemes. The error rates of the final dissimilarities obtain with the regularised and unregularised Ricci flow are summarised in Table 4.2. Error_0, Error_1, RError_1, Error_2, RError_2 individually refer to the error rate of original non-Euclidean dissimilarity, of the final dissimilarity obtained from the unregularised Ricci flow with kernel embedding, of the final dissimilarity obtained from the regularised Ricci flow with kernel embedding, of the final dissimilarity obtained from the unregularised Ricci flow with Isomap embedding, of the final dissimilarity obtained from the regularised Ricci flow with Isomap embedding respectively.

Dataset	$Error_0$ (%)	$Error_1$ (%)	$RError_1$ (%)	$Error_2$ (%)	$RError_2$ (%)
Chickenpieces-5	34.53	51.59	21.82	44.25	38.22
Chickenpieces-10	16.14	48.17	8.67	35.42	17.68
Chickenpieces-15	7.40	44.81	5.84	21.44	5.19
Chickenpieces-20	6.28	41.30	4.63	22.73	6.44
Chickenpieces-25	4.26	36.95	4.01	18.80	5.40
Chickenpieces-30	4.48	35.56	6.05	15.35	5.38
Chickenpieces-35	6.28	36.32	6.05	16.37	4.71
Chickenpieces-40	8.74	32.29	9.19	20.18	8.07
CoilYork	23.26	46.15	24.13	36.57	25.86

TABLE 4.2: 1NN Classification errors of the original dissimilarity ($Error_0$) , of the final dissimilarity of the unregularised Ricci flow with kernel embedding ($Error_1$) and the regularized Ricci flow with kernel embedding ($RError_1$), the Ricci flow with Isomap embedding($Error_2$) and the unregularised Ricci flow with Isomap embedding($RError_2$).

Dataset	Pos (%)	$Error_0$ (%)	$Error_1$ (%)	$RError_1$ (%)	$Error_2$ (%)	$RError_2$ (%)
CoilYork (3NN)	34.38	29.17	44.44	28.12	35.76	27.78
CoilYork (5NN)	35.42	27.08	45.49	33.68	32.64	25.69
CoilYork (libSVM)	37.16		45.14	39.93	30.91	26.39

TABLE 4.3: CoilYork Classification errors of the positive subspace dissimilarity (Pos), the original dissimilarity ($Error_0$) , the final dissimilarity of the unregularised Ricci flow with kernel embedding ($Error_1$) and the regularised Ricci flow with kernel embedding ($RError_1$), the Ricci flow with Isomap embedding($Error_2$) and the unregularised Ricci flow with Isomap embedding($RError_2$).

The first point to note is that for both the kernel embedding and Isomap embedding methods, we obtain better classification results when heat kernel regularisation is used. In the unregularised versions of Ricci flow, the application of the Ricci flow scheme caused the classification error to increase with respect to iteration number. The regularised version of the Ricci flow on both embedding preserves the grouping structure of data, as the classification error almost stay the same. Overall, the results from the regularised kernel embedding is slightly better than those from the regularisd Isomap embedding. Finally, the choice of embedding scheme strongly affects the rate of decrease of the negative eigenfraction, with Isomap giving a faster rate of decrease with iteration number than the kernel embedding. However, for the kernel embedding, the use of regularisation has little effect on the rate of decrease. For the Isomap embedding, the rate of decrease is similar to each other when the Ricci flow process starts. And in the middle of process, the rate of decrease for the regularised Ricci flow with Isomap embedding is smaller than that of the unregularised version of Ricci flow with Isomap. Hence the negative eigenfraction for the unregularised version converges earlier and converges to a slight higher value than that of the regularised one illustrated in the (C) subplots from Figure 4.20 to Figure 4.28.

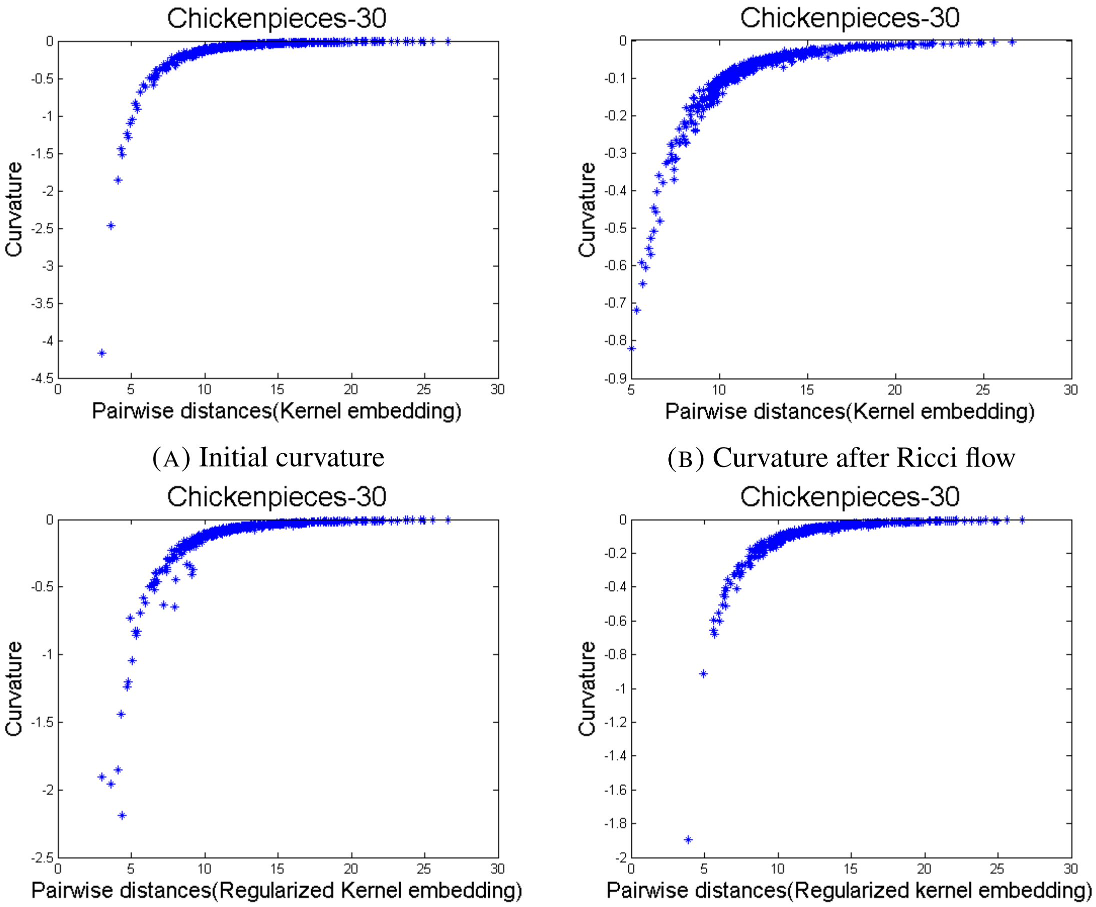
Since our objective is to correct the non-Euclidean artefacts and also to preserve the original distance as much as possible, Figure 4.19 plots the comparison of original non-Euclidean distances and resulting Euclidean distances after 500 iterations of Ricci flow with kernel embedding (A), Ricci flow with Isomap embedding (B), Ricci flow with regularised kernel embedding (C), Ricci flow with regularised Isomap embedding (D). (A) and (C) shows that the Ricci flow embedding preserves global distances and expand local distances. The expansion is the effects of kernel embedding and Ricci flow process. Since Ricci flow with Isomap embedding is a combination of Isomap embedding and kernel embedding, some distances are expanded and some are contracted. The global distances under Isomap embedding are expanded more than those under kernel embedding, because the global distances are approximated from edge-connected path which is large than the original distances over iterations. The regularised version under both schemes slow down the expansion by smoothing out extreme values of curvatures illustrated by lowest point in (C) and (D).

Finally we have compared our results with the known manifold embedding technique

Isomap, those obtained with positive subspaces and those with original distances. Figure 4.29 shows the 1NN error rate as function of the shape parameter L (the segment length). The best results are obtained with the regularized Ricci flow on kernel embedding. We have shown in Figure 4.16 that the unregularised version of Ricci flow embedding on both embedding scheme gives worse results than the purely Isomap embedding, so the regularized Ricci flow embedding performs better than the unregularised version and the results of the unregularised version are not repeated here. All of the remaining methods give worse results than applying the classifier to the original distance data. Our regularised Ricci flow embedding is a potentially good way to transform the non-Euclidean dissimilarity measure to be Euclidean, as the discriminating power of the evolved Euclidean distance is near to the original non-Euclidean distances.

The above results are evaluated by the 1NN classifier only. Compared to KNN classifier, the decision is determined by the closest neighbour, the locality of estimation is limited and the decision regions less smoother. We test the results on CoilYork by 3NN classifier and 5NN classifier using leave one out cross-validation in Table 4.3. The regularised Ricci flow under Isomap embedding obtain the similarities which get closet classification results to the original non-Euclidean dissimilarities, better than the dissimilarities from positive subspace and other Ricci flow embedding emthods. It demonstrates that Ricci flow with Isomap embedding preserve the local structure of CoilYork data. KNN uses local information only. Our objective is to correct non-Euclidean dissimilarities so that the resulting dissimilarities can be analysed by sophisticated geometric machine learning techniques. Finally We use linear SVM (Support Vector Machine) classifier with default parameter $C = 1$ under two folder cross-validation to compare the positive subspace embedding and the four Ricci flow embedding. The classification errors are displayed in Table 4.3. Since original distances are non-Euclidean and SVM is based on vectors, so we leave $Error_0$ blank and compare the distances from positive subspace of original distances and the resulting Euclidean distances from our four embedding methods. It shows

Ricci flow with Isomap embedding and the regularised version under Isomap embedding performs the better than the positive subspace. Overall, the experiments on CoilYork data shows that Ricci flow under Isomap embedding performs better than Ricci flow under kernel embedding.



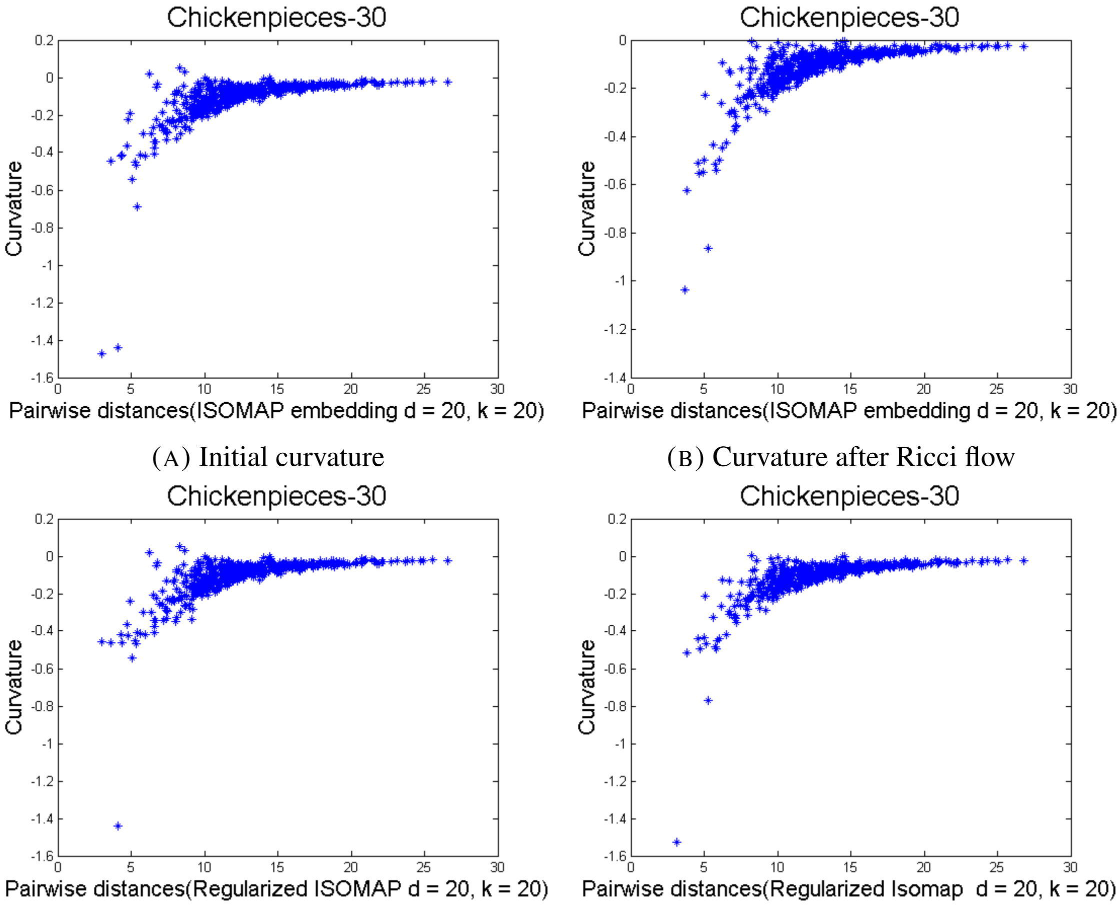
(A) Initial curvature

(B) Curvature after Ricci flow

(C) Initial regularized curvature

(D) Curvature after regularized Ricci flow

FIGURE 4.17: (A)(B)(C)(D) plots initial curvatures from the kernel embedding, the curvatures after Ricci flow with kernel embedding, initial regularised edge curvatures from heat diffusion and the kernel embedding, the regularised curvatures after Ricci flow with kernel embedding



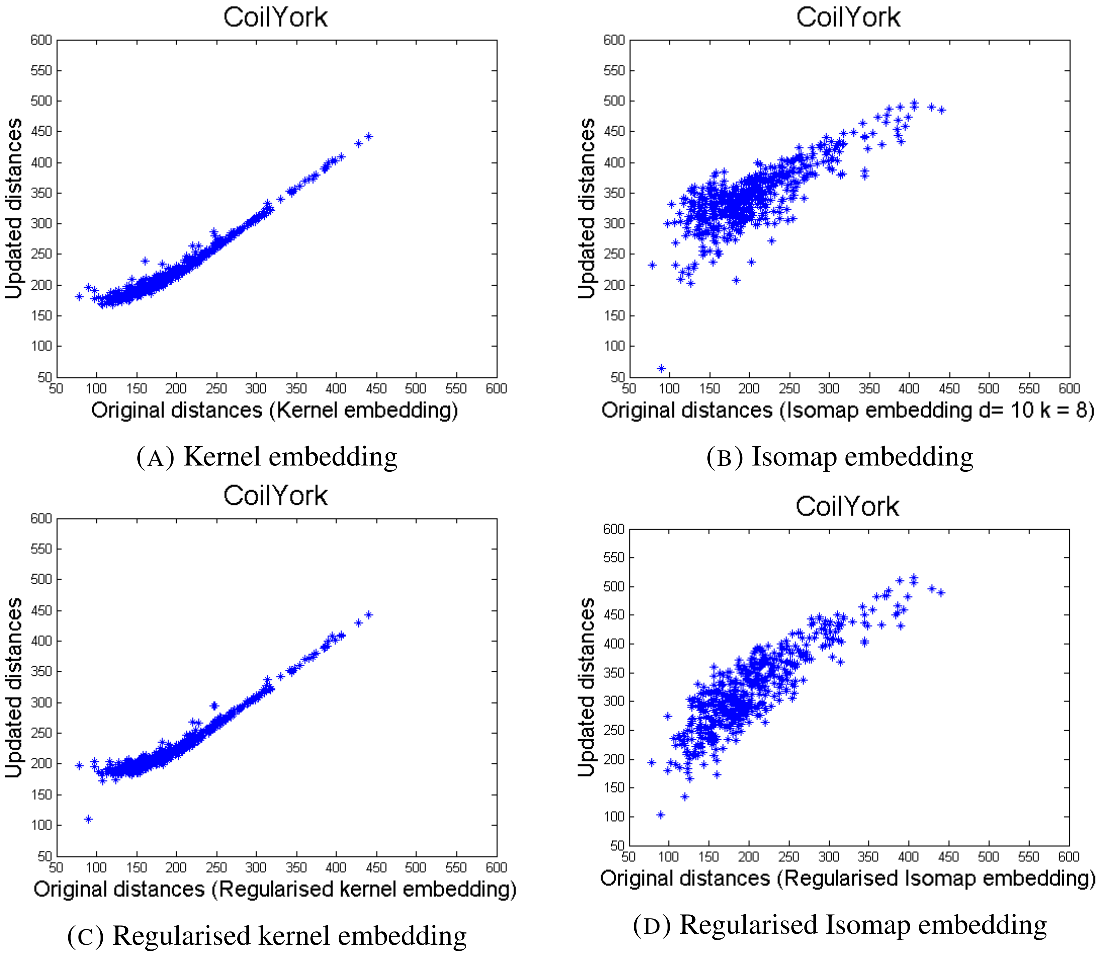
(A) Initial curvature

(B) Curvature after Ricci flow

(C) Initial regularized curvature

(D) Curvature after regularized Ricci flow

FIGURE 4.18: (A)(B)(C)(D) plots initial curvatures from Isomap embedding, the curvatures after Ricci flow with Isomap embedding, the initial regularised edge curvatures from heat diffusion and Isomap embedding, the regularised curvatures after Ricci flow with Isomap embedding



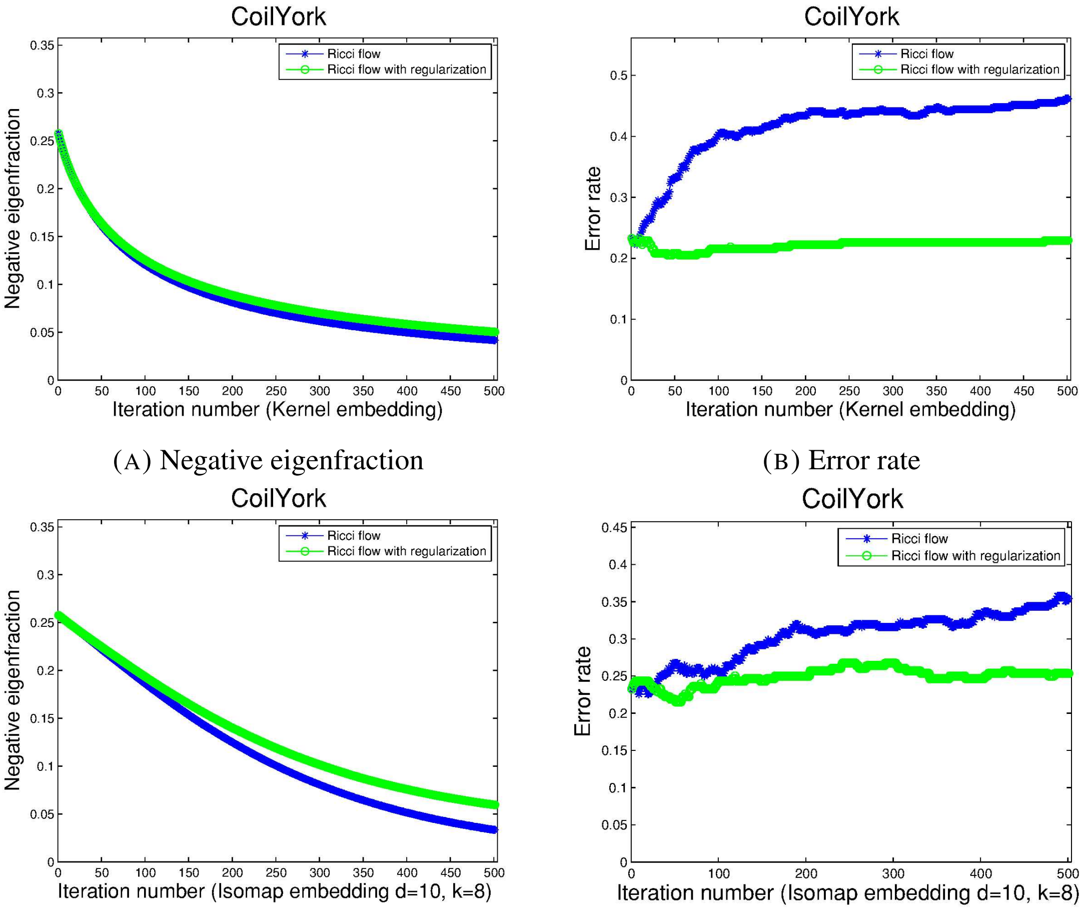
(A) Kernel embedding

(B) Isomap embedding

(C) Regularised kernel embedding

(D) Regularised Isomap embedding

FIGURE 4.19: (A)(B)(C)(D) plots 500 random pairs of distances before and after 500 iterations of Ricci flow with kernel embedding, Ricci flow with Isomap embedding, Regularised Ricci flow embedding with kernel embedding and Regularised Ricci flow embeddign with Isomap embedding



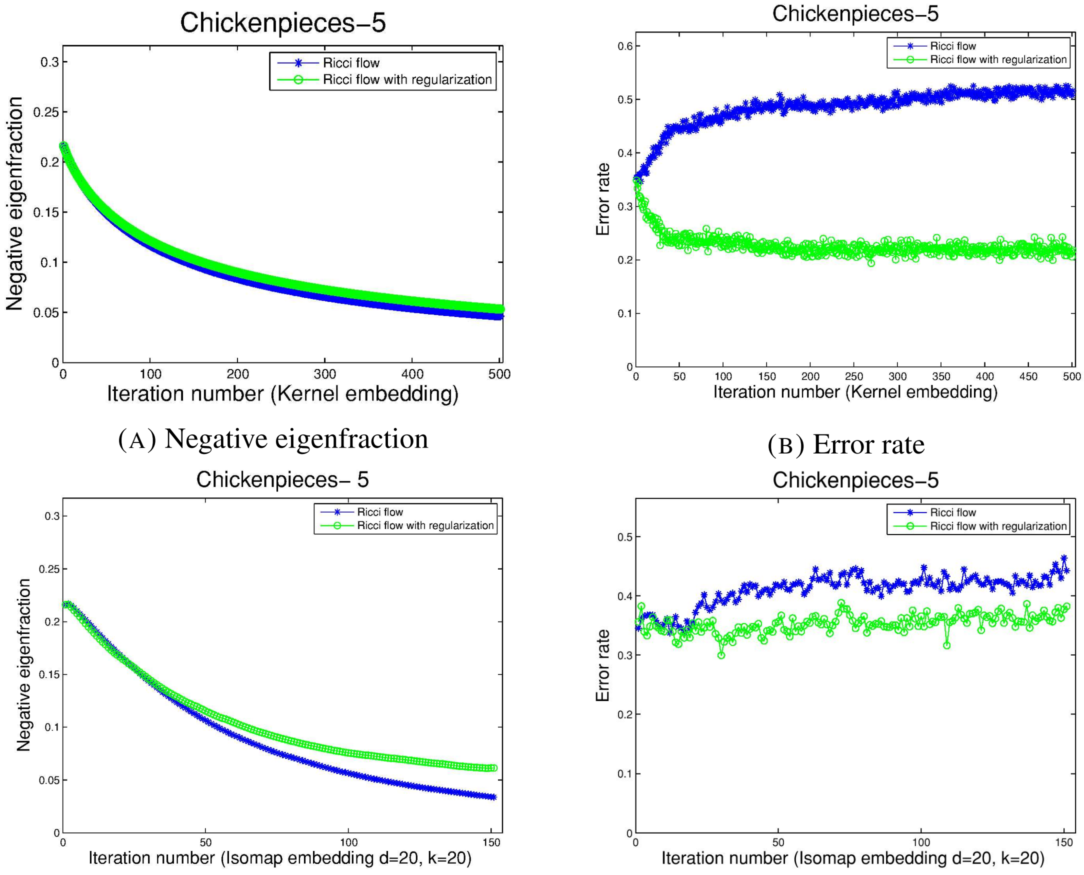
(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.20: (A)(B) show the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with kernel embedding for the CoilYork dataset. (C)(D) show the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with Isomap embedding for the CoilYork dataset.



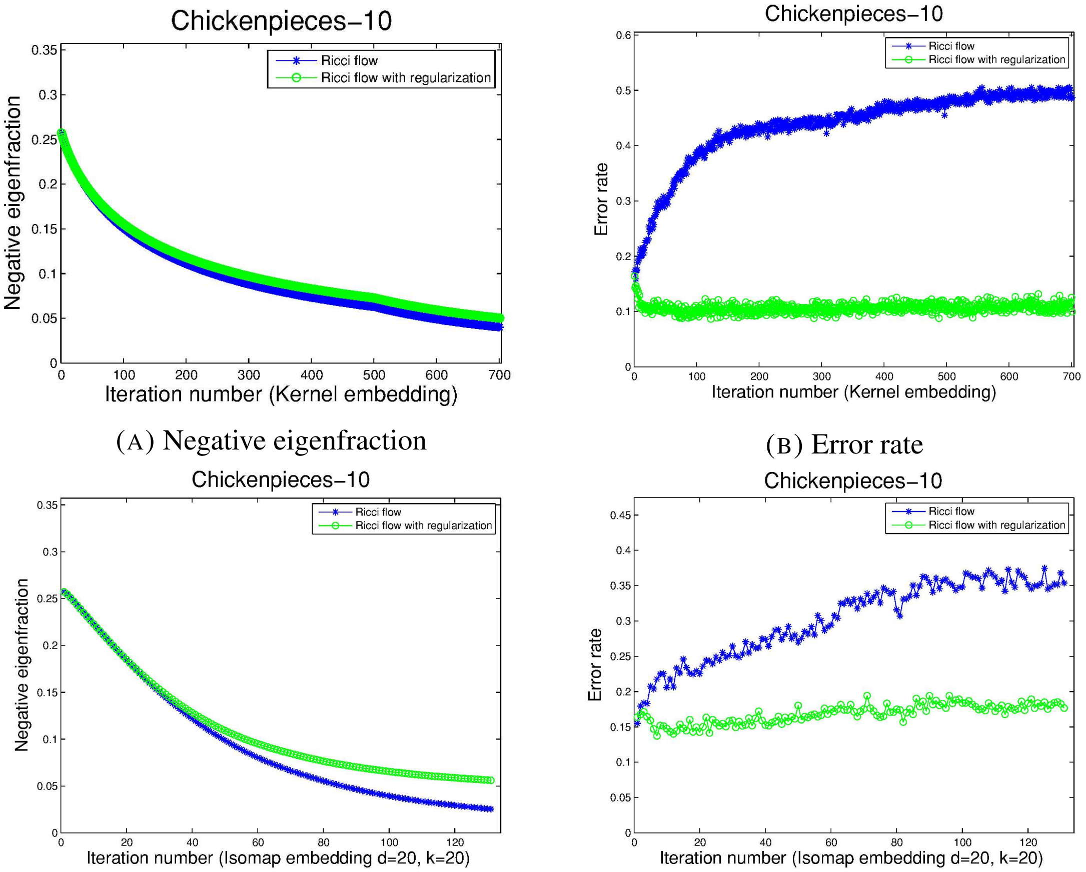
(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.21: (A)(B) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with kernel embedding for the Chickenpieces-5 dataset. (C)(D) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with Isomap embedding for Chickenpieces-5.



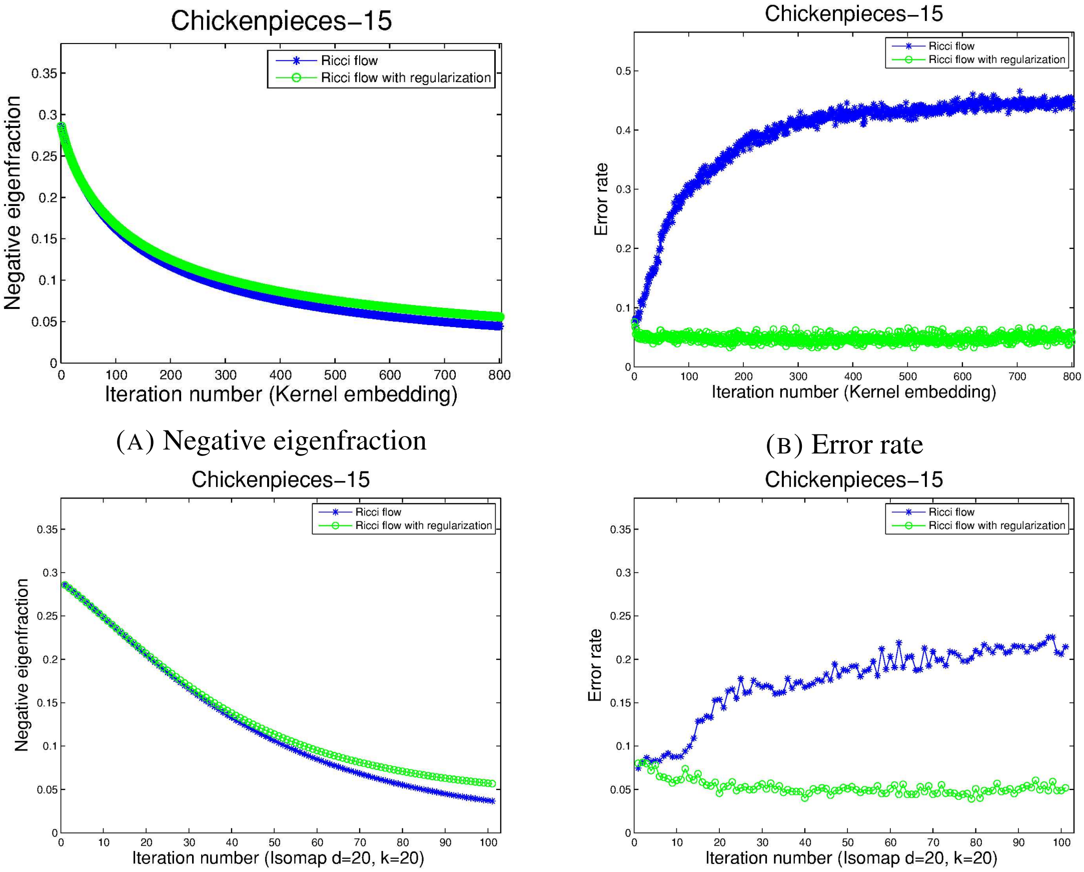
(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.22: (A)(B) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with kernel embedding for the Chickenpieces-10 dataset. (C)(D) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with Isomap embedding for the Chickenpieces-10 dataset.



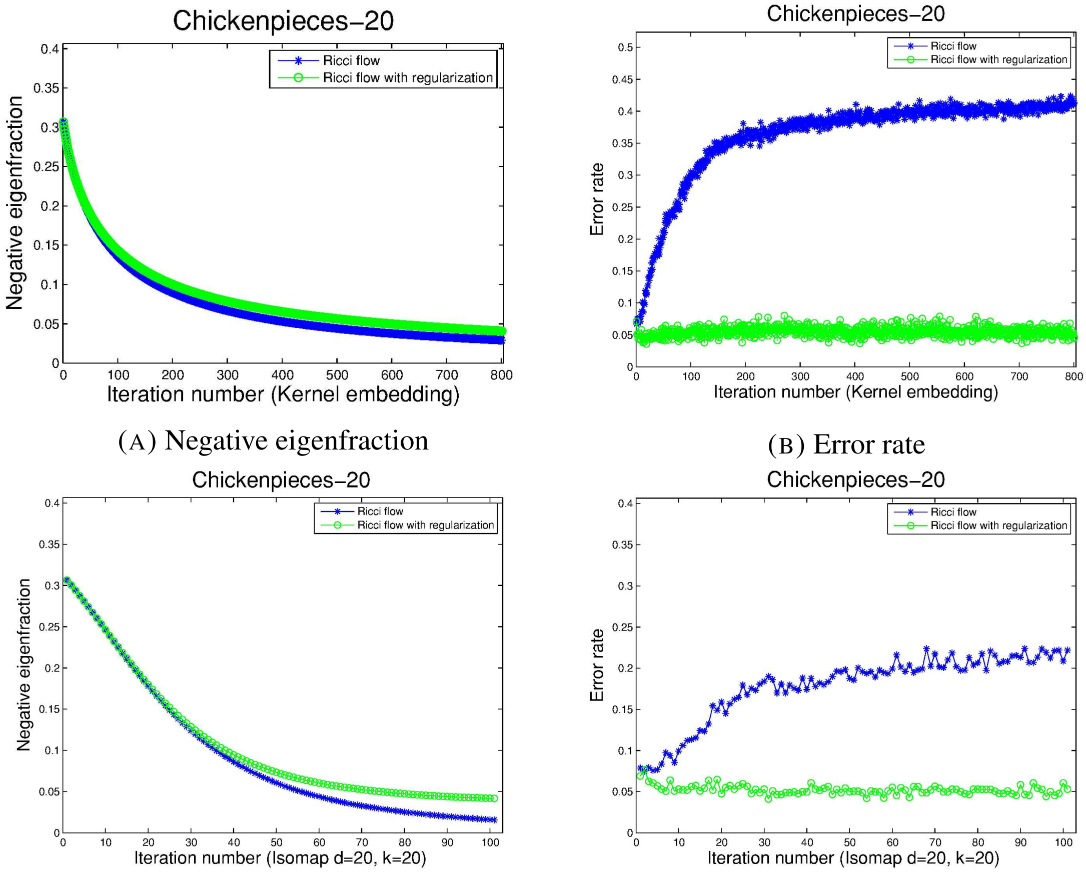
(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.23: (A)(B) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with kernel embedding for the Chickenpieces-15 dataset. (C)(D) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with Isomap embedding for the Chickenpieces-15 dataset.



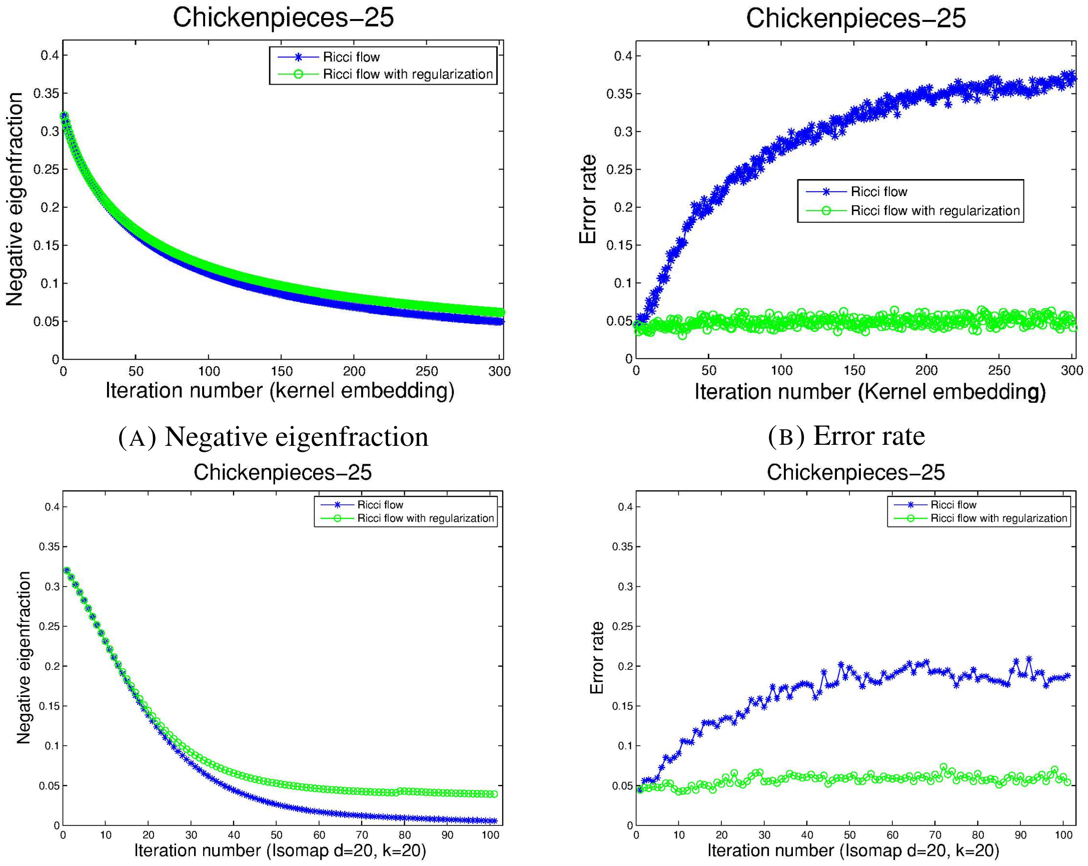
(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.24: (A)(B) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with kernel embedding for the Chickenpieces-20 dataset. (C)(D) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with Isomap embedding for the Chickenpieces-20 dataset.



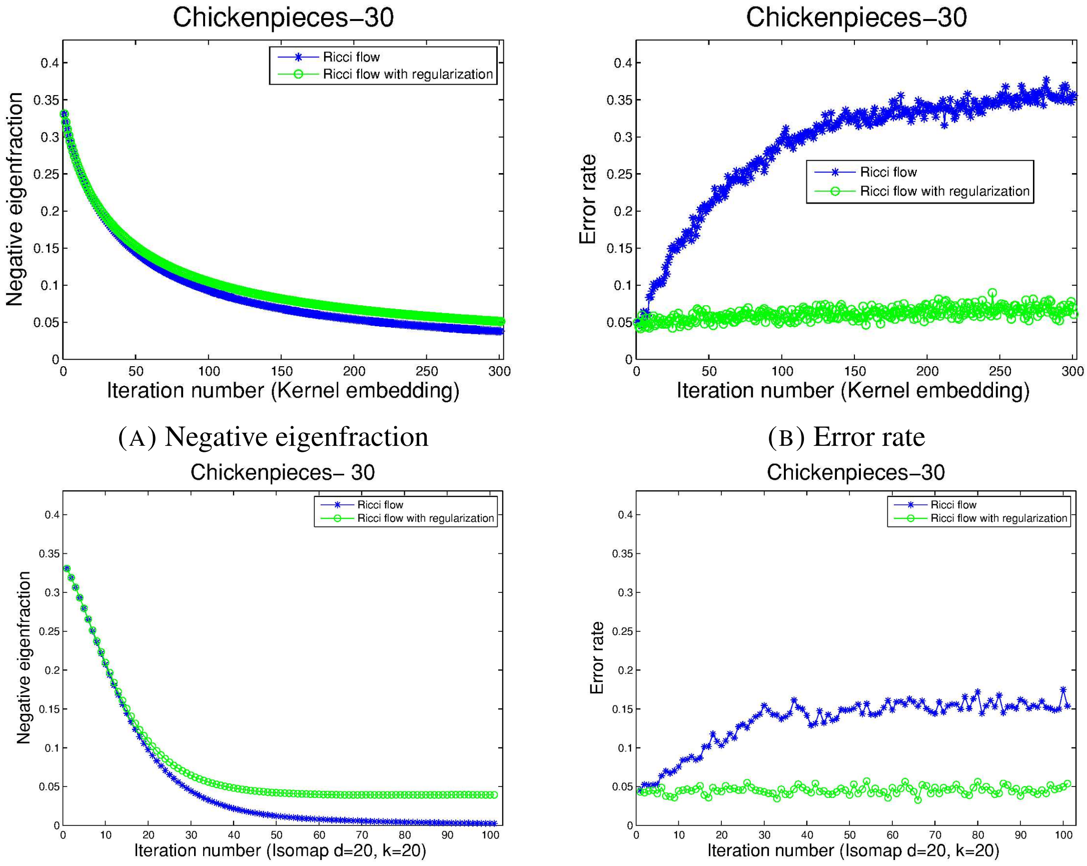
(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.25: (A)(B) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with kernel embedding for the Chickenpieces-25 dataset. (C)(D) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with Isomap embedding for the Chickenpieces-25 dataset.



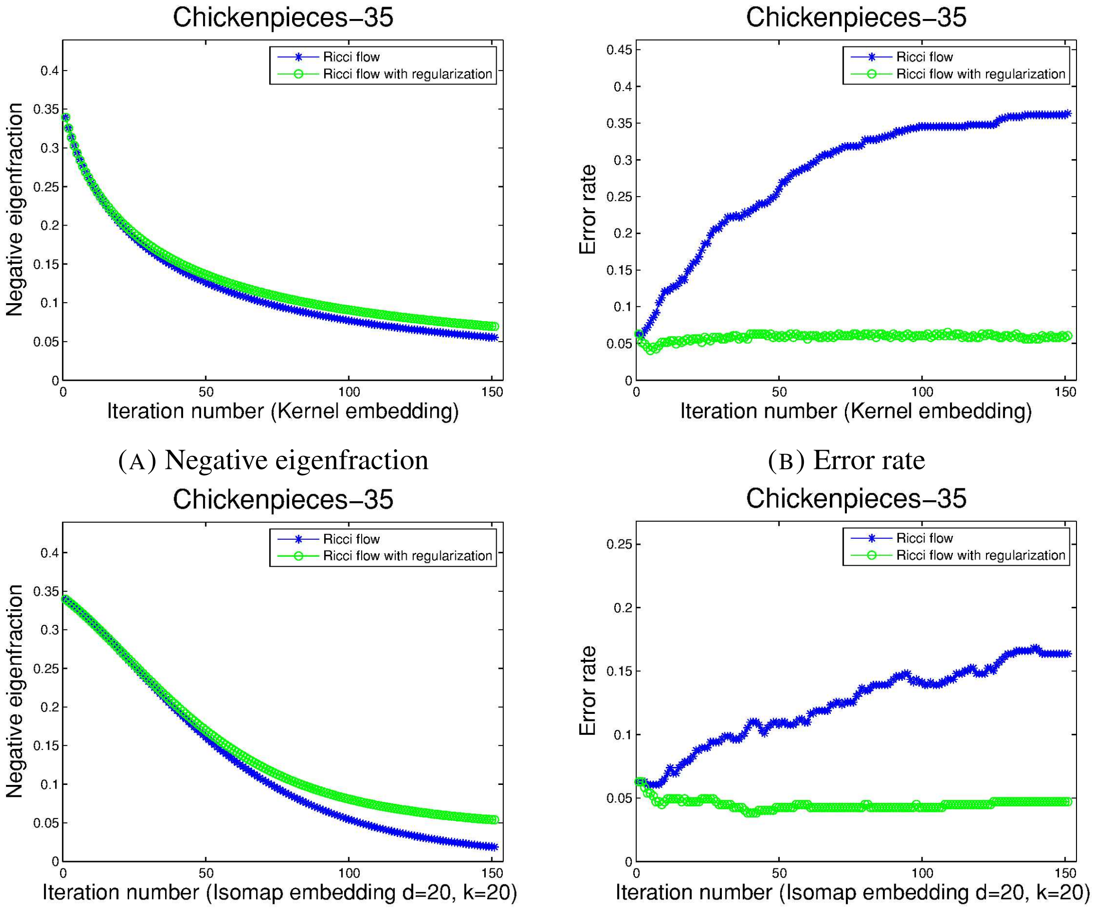
(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.26: (A)(B) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with kernel embedding for the Chickenpieces-30 dataset. (C)(D) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with Isomap embedding for the Chickenpieces-30 dataset.



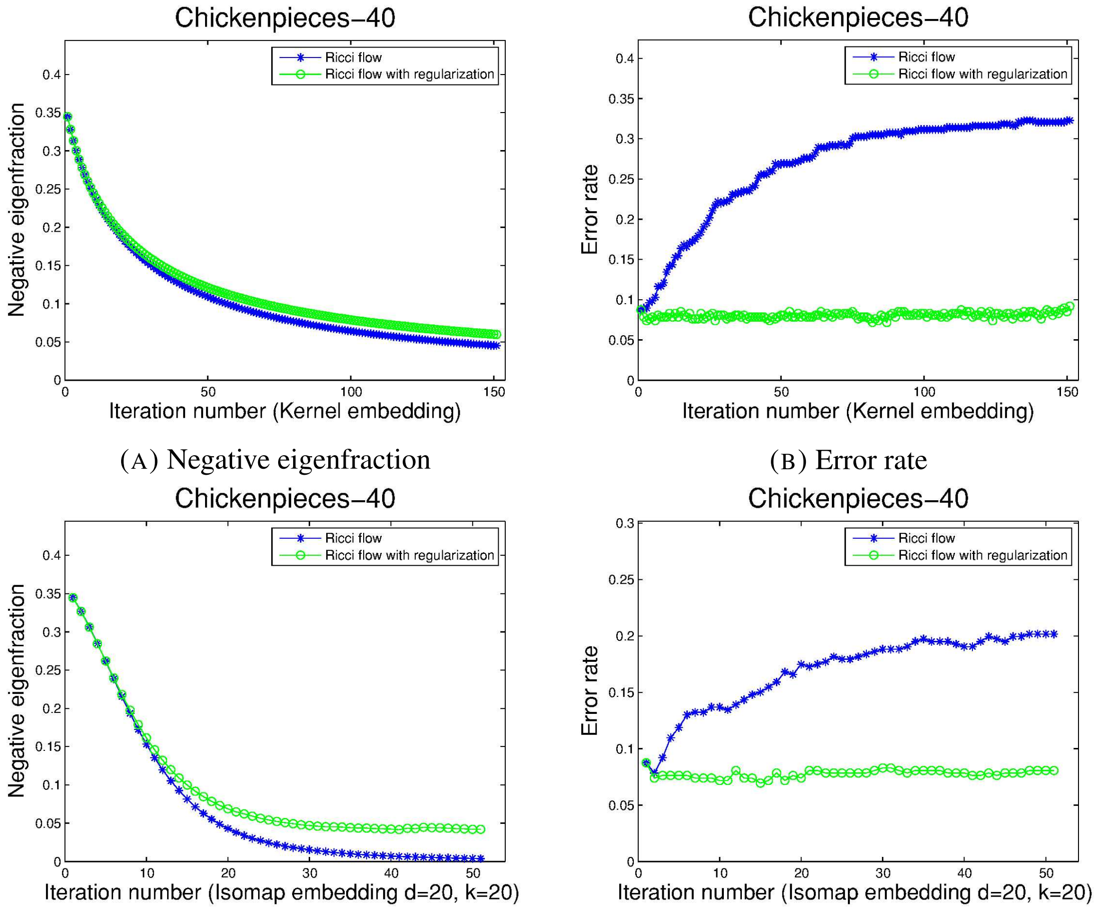
(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.27: (A)(B) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with kernel embedding for the Chickenpieces-35 dataset. (C)(D) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with Isomap embedding for the Chickenpieces-35 dataset.



(A) Negative eigenfraction

(B) Error rate

(C) Negative eigenfraction

(D) Error rate

FIGURE 4.28: (A)(B) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with kernel embedding for the Chickenpieces-40 dataset. (C)(D) plots the negative eigenfraction and 1NN error rate during the iteration of the regularised Ricci flow embedding with Isomap embedding for the Chickenpieces-40 dataset.

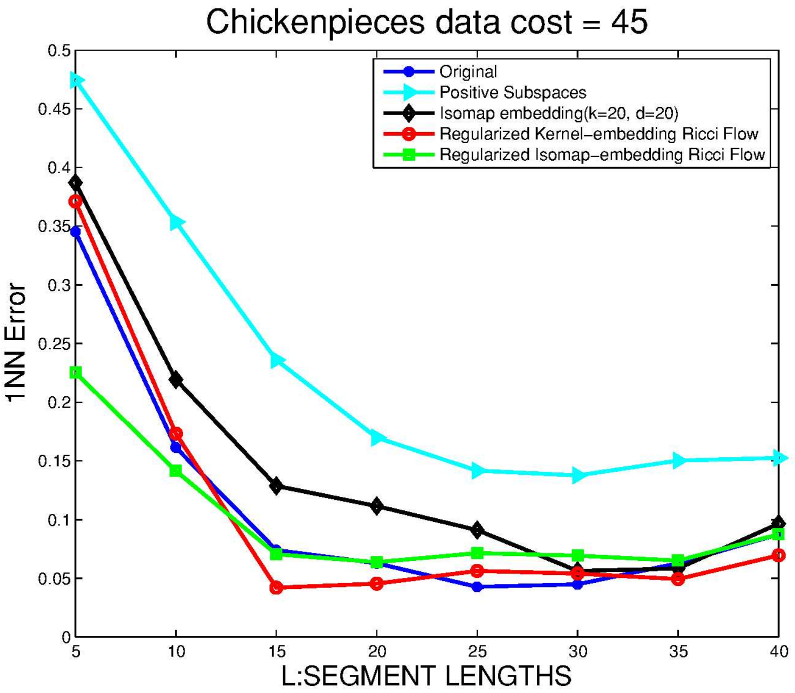


FIGURE 4.29: Error rate from 1NN for chicken pieces dataset cost =45

4.7 Summary

In this chapter, we presented two methods to rectify a set of non-Euclidean dissimilarities to make it more Euclidean. Our approach is to embed the objects represented by dissimilarities into curved manifold and correcting the non-Euclidean artifacts by evolving the curved manifold to Euclidean space with zero sectional curvature. We model the manifold consisting of individual edges with constant sectional curvature. We use the Ricci flow process and develop a way to evolve the manifold by reducing the sectional curvature.

In the piecewise Ricci flow embedding, we model each edge as a patch with constant curvature and the sectional curvature is estimated in terms of the geodesic distances and Euclidean distances. The Euclidean distance is ccomputed from the kernel embedding and the Isomap embedding. The geodesic distance is updated by evolving the sectional curvature and fixing the Euclidean distances. This allows the geodesic distance move towards the Euclidean distance.

In the regularized Ricci flow embedding, we use heat kernel process as a regularization process to smooth out the extreme curvatures caused by the kernel embedding and the Isomap embedding. We build a dual graph which allows the regularization process smooth sectional curvatures on edges.

We experimentally applied our piecewise Ricci flow method to the Chickenpieces data and the CoilYork data, we demonstrate that the distance measures can be transformed into a Euclidean space, but with some loss of discriminating power. Using the Isomap embedding schemes gives good performance, but the performance is worse when the kernel embedding is used. The loss of information is caused by the effect of the Ricci evolution process as it is applied independently on each edge and ignores the local structure of the manifold.

Hence, one way to improve this is to maintain local structure during the Ricci flow smoothing process. Thus we have developed the regularized Ricci flow embedding. When heat kernel regularization is used, the local ranking of distance measures (or the grouping of the data) is preserved, and better performance is achieved from 1NN classifier. The regularised Ricci flow has two advantages. First, it delivers data in a form so that various geometric classification methods can be applied directly to the data. Secondly, it preserves informative content hidden in the negative eigenvalues of non-Euclidean dissimilarity for classification, which is usually ignored by other Euclidean embeddings. These advantages are demonstrated by obtaining similar or even lower error rates from corrected data than those of the original data with 1NN classifier. The Ricci flow evolution minimise the curvatures, when the curvatures reach zero, then the geodesic and Euclidean distances are equal and the negative eigenfraction is zero.

Overall, our experiments show that both the piecewise Ricci flow embedding and the regularized Ricci flow embedding are able to rectify non-Euclidean dissimilarities into a set of Euclidean dissimilarities. Moreover, the regularised Ricci flow with both embedding schemes preserve the local structure of data for grouping, while the resulting dissimilarities from the piecewise Ricci flow embedding loss some discriminative power for classification.

Chapter 5

Extended Work on Ricci Flow Embedding

5.1 Introduction

In previous chapters, we have proposed two methods based on Ricci flow to rectify the non-Euclidean distances: the piecewise Ricci flow embedding and the regularised Ricci flow embedding. In the piecewise Ricci flow embedding, we experimentally find that the structure of data is distorted, due to the piecewise nature of the manifold. Although this problem can, to some extent, be remedied by regularizing the edge curvature, the issue of preserving structure persists. Moreover, we find out that the embedding methods affect the magnitude of curvature. To overcome these problems, we turn to a tangent space representation of data using the exponential map and log map. Firstly, we can reduce the reliance on the piecewise embedding and its effect on individual edges by using the tangent space representation. Secondly, the tangent space representation provides a means of preserving the distances between the points on the manifold and the origin of the map. This allows us to flatten the manifold to some extent while preserving the local structure of the data. We developed a tangent space based embedding method which use the spherical embedding to compute the initial curvatures of local patches. We demonstrate its applicability for correcting non-Euclidean dissimilarity on a set of distances from the Chickenpieces dataset.

When we investigate the local fluctuations in edge curvature at the piecewise manifold embedding, it is found that the kernel embedding and Isomap embedding do not restore the exact original Euclidean distances between close points on the embedded space. This leads to incorrect curvature estimation for the manifold, where the objects reside on. The computed curvature is affected by the embedding methods we used for computing the Euclidean distance (such as kernel embedding and Isomap embedding we used in previous chapter) in the ambient space for each pair of objects lying on the manifold. Curvature estimation is a key component in our Ricci flow framework. In order to remove the effects of the Euclidean embedding and complement our Ricci flow embedding framework, we explored a new way to compute the curvature of manifold, where the objects reside, from only pairwise distances, without a priori specifying the structure or the dimensionality of the manifold. We demonstrate its effectiveness for estimating curvatures on six sets of dissimilarity matrices, whose elements are geodesic distances between points which lie on a two dimensional sphere.

5.2 Tangent Space Reprojection

In the case of piecewise manifold, we update the sectional curvature independently for individual edge and ignore the connected relation among edges. To capture the local structure, we use heat kernel to regularize the curvature in the previous Chapter. In this section we aim to reduce the reliance on the piecewise embedding and its effect on individual edges. Similar to previous approaches, we commence by representing the dissimilarity data using a weighted graph and embed the graph onto a manifold which is embedded in some Euclidean space. The embedding manifold is assumed to contain a set

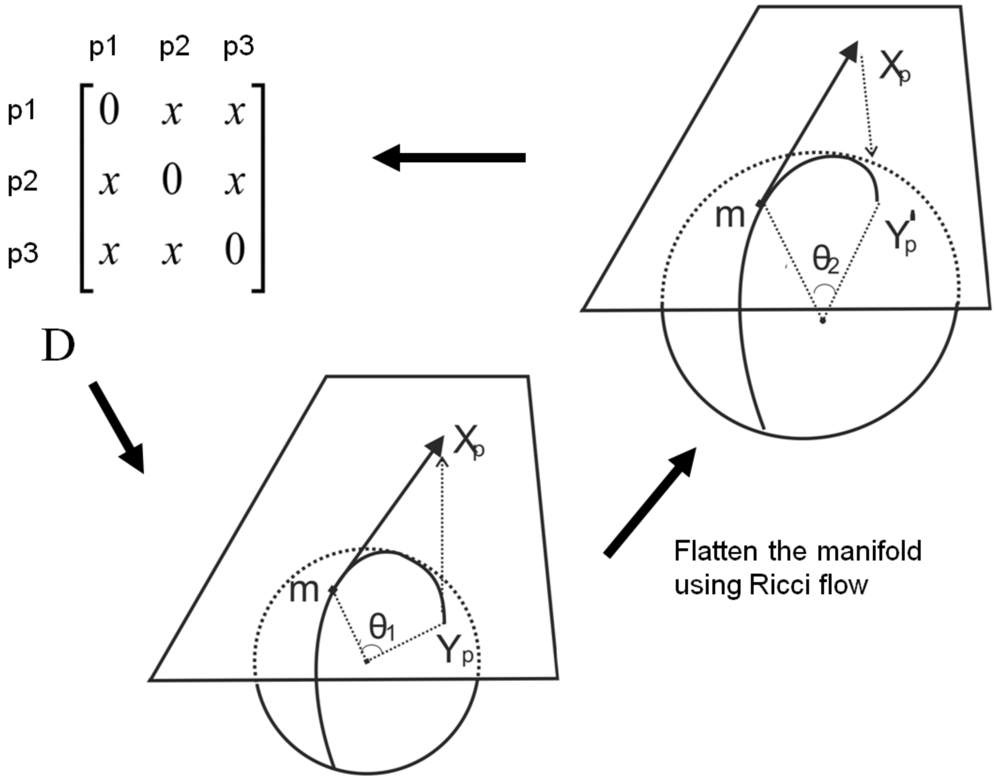


FIGURE 5.1: Illustration of the tangent space reprojection.

of local patches with constant sectional curvatures. We share the same idea of correcting non-Euclidean dissimilarities by flattening the manifold where the data points are lying as the two techniques in previous chapter. Rather than assuming an individual edge as a local patch, here we consider a 2-degree nearest neighbour graph as a local patch. We use the spherical embedding to obtain the sectional curvature of the local patch. Similarly, we evolve the hyperspherical patches under Ricci flow. With the tangent space representation, we first map the local patch centered from a data point (the reference point) to the tangent space passing through the reference point, and then map the local patch back to a slightly flattened (or inflated) hypersphere. Finally we stitch local patches to get a smoothed out global embedding manifold. This is the idea of our third Ricci flow embedding technique, the tangent space reprojection, which is illustrated in Figure 5.1

5.2.1 Spherical Embedding

Spherical embedding [122] provides a means by which to embed objects represented in terms of dissimilarity data onto constant curvature manifolds. This approach maps the non-Euclidean dissimilarity on to a hypersphere whose radius of curvature is determined by the dissimilarity data. It embeds the dissimilarity between objects onto a metric space. This approach finds out the radius of the curvature by an optimization problem on the smallest eigenvalues of a similarity matrix. On a hypersphere, the geodesic distance between two points is the length of the arc of the great circle on the hypersphere that connects the two points. Suppose that the tangent vector to the manifold undergoes a change in direction of θ_{ij} as we move along a connecting arc between the two points, then the distance between the two points is defined as

$$d_{ij} = r\theta_{ij}\text{where } r \text{ is the radius} \tag{5.1}$$

We consider the coordinates of object on a hypersphere with radius r in the ambient Euclidean space as $\mathbf{x}_i$, $i = 1, 2, \cdots, n$ and the coordinate origin at the center of the hypersphere. Thus a point can be represented by a position vector $\mathbf{x}_i$ with length r, the angle between $\mathbf{x}_i$ and $\mathbf{x}_j$ is θ_{ij}. Then the inner product is defined as

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle = r^2 \cos\theta_{ij} = r^2 \cos\frac{d_{ij}}{r} \tag{5.2}$$

The above defined the elements of inner product matrix $Z = XX^T$ where $X = (\mathbf{x}_1|\mathbf{x}_2|\cdots|\mathbf{x}_n)$. Because the embedding (ambient) space has dimension $n-1$, X is n points with dimension $n-1$ and Z is an inner product of points (vectors) in the Euclidean space, Z should be positive semi-definite with rank $n-1$. It means Z should have a single zero eigenvalue λ_0 which is the smallest eigenvalue. In this way, the radius of the curvature can be found by minimising the magnitude fo the smallest eigenvalue λ_0 of the inner product matrix $Z(r)$. The following equation presents the optimisation function over r, because the smallest eigenvalue is determined by r and Z. This is how the curvature of the embedding hypersphere is related to the eigenvalues of the inner product matrix.

$$r^* = \arg\min_r |\lambda_0[Z(r)]| \tag{5.3}$$

After locating the optimal radius, the embedding coordinates are obtained through the eigendecomposition of the inner product matrix,

$$X = \Lambda^{\frac{1}{2}}\Phi \tag{5.4}$$

where Λ is the diagonal matrix with the ordered eigenvalues of the inner product matrix Z and Φ is the matrix with the ordered eigenvectors as columns.

We choose the spherical embedding for three reasons. Firstly, spherical embedding estimates the sectional curvature of the local patch since the radius is found out during the embedding process. Secondly, due to the geometric properties of the hypersphere, there are explicit functions fit the exponential map and log map, which is very important for flattening local patches. Thirdly, the spherical embedding maps objects in terms of dissimilarity data in a metric space and preserves the local structure of the data [122].

Please note the spherical embedding is appropriate for embedded manifold with positive curvatures. For manifold with negative eigenvalues, there is counterpart hyperbolic embedding. Because hyperbolic space has only one negative dimension and the embedding space has dimension $n-1$, the inner product of points (vectors) should have one negative eigenvalue and a zero eigenvalue (the second smallest eigenvalue), with the rest positive. Similar to spherical embedding, the curvature of hyperbolic embedding is searched by minimising the magnitude of the second smallest eigenvalues of the inner product.

5.2.2 Tangent Space, The Exponential and Log Map

Flattening a local patch on a hypersphere can be achieved by inflating the hypersphere with the means of Ricci flow. In addition to this, we need to locate the points on the inflated sphere and preserve the local distances for correcting a set of non-Euclidean dissimilarities to a set of Euclidean dissimilarities. To achieve this, we require an important tool of Riemannian geometry, which is the exponential map and the log map [122]. On a Riemannian manifold, the exponential map, denoted by $\mathrm{Exp_p}[.]$, maps points on the tangent plane T_pM, which passes through the reference point on the manifold, to points on the manifold along the geodesic to the reference point. The map has an origin (or reference point) which defines the point at which we can construct the tangent plane and the tangent vector of the manifold. Figure 5.2 shows an example of exponential map on a sphere. Let T_pM be the tangent space passing through m on the manifold, X_p be the position vector with origin at m on the tangent plane, the Euclidean distance from point X_p to m on T_pM is given by $d_E(X_p, m) = ||X_p|| = \langle X_p, X_p \rangle^{\frac{1}{2}}$. Here the point m is the origin of the map and is mapped onto the origin of the tangent space. The exponential of X_p along the geodesic in the direction of X_p is the point Y_p lying on the manifold so that geodesic distance between Y_p and m equals to $||X_p||$. The inverse of the exponential map is the Log map, denoted by $\mathrm{Log_p}[.]$, which maps points from the manifold to points in the tangent space passing the reference point, i.e. mapping Y_p to X_p. An important property of these map is their ability to preserve distances from the reference point. That is, the distance between the origin of the map and a point on the manifold is the same as the Euclidean distance between the mapped point on tangent space [122]. Thus we have

$$Y_p = \mathrm{Exp}_m(X_p) \tag{5.5}$$

$$X_p = \mathrm{Log}_m(Y_p) \tag{5.6}$$

$$d_G(Y_p, m) = d_E(X_p, m) \tag{5.7}$$

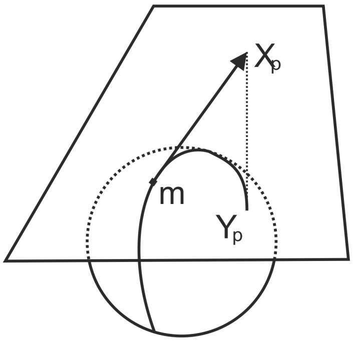


FIGURE 5.2: The exponential map and log map

The distance d_G is the geodesic distance on the manifold and d_E is the Euclidean distance on the tangent space. Note that $\mathrm{Exp_m}[.]$ in 5.5 and $\mathrm{Log}_m[.]$ in 5.6 are a formal notation and do not represent the normal $\log$ and $\exp$ functions.

On the spherical manifold, we have

$$d_G(\mathbf{y}_p, m) = r\theta \tag{5.8}$$

$$d_E(\mathbf{x}_p, m) = |\mathbf{x}_p| \tag{5.9}$$

$$\langle \mathbf{x}_p, \mathbf{y}_m \rangle = 0 \tag{5.10}$$

where $\mathbf{y}_p$ is the coordinate vector for point p on the manifold, $\mathbf{x}_p$ is the coordinate vector for point p in the tangent space, m is the reference point or origin of the map, r is the radius of the sphere, $\mathbf{y}_m$ and $\mathbf{y}_p$ be the position vector on the spherical manifold whose length equals to the radius of the sphere (the origin is the center of the hypersphere). i.e., $|\mathbf{y}_m| = |\mathbf{y}_p| = r$. And $\theta = \cos^{-1}\frac{\langle \mathbf{y}_m, \mathbf{y}_p \rangle}{r^2}$ is the angle between these vectors. Hence by solving the above equations, the coordinate $\mathbf{x}_p$ on the tangent plane mapped from a point $\mathbf{y}_p$ on the spherical manifold using the exponential map is: [122]

$$\mathbf{x}_p = \frac{\theta}{\sin\theta}(\mathbf{y}_p - \mathbf{y}_m \cos\theta) \tag{5.11}$$

the coordinate $\mathbf{y}_p$ on the spherical manifold mapped from a point $\mathbf{x}_p$ on the tangent plane using the log map is:

$$\mathbf{y}_p = \mathbf{y}_m \cos\theta + \frac{\sin\theta}{\theta}\mathbf{x}_p. \tag{5.12}$$

This set of transformations is illustrated in Figure 5.2. In the following section, we make use of the above relations to update new coordinates on the inflated sphere and on compute the new distances on the evolved local patch.

5.2.3 Updating distances

Our aim is to flatten the global manifold by gradually smoothing out the local patches. This is achieved by representing sub-graphs of objects on the local hyperspheres, mapping the points to the tangent space through the log-map function, reducing the curvatures (i.e., increasing the radii) of the individual hyperspherical patches, and then mapping the data back onto the inflated hyperspheres through the exponential-map function. The decrease in the sectional curvatures of the hyperspheres is determined by the Ricci flow [14] and follows the equation 4.21 in section 4.3 in previous chapter.

Once, the inflation and reprojection onto the hypersphere are complete, we compute the new coordinate vector of a point on the inflated hypersphere based on the old coordinate vector on the original hypersphere by using Equation 5.12:

$$\mathbf{y}_{p_{i+1}} = (1 + 2K_i t)^{\frac{1}{2}}(\mathbf{y}_{m_i}\cos\theta_{i+1} + \frac{\sin\theta_{i+1}}{\sin\theta_i}(\mathbf{y}_{p_i} - \mathbf{y}_{m_i}\cos\theta_i)). \tag{5.13}$$

where the angle on the original sphere is $\theta_n = K_n \cos^{-1}\langle \mathbf{y}_m, \mathbf{y}_p \rangle$.

As the geodesic distances to the origin are preserved, we can update the angles on the inflated sphere

$$\theta_{i+1} = \frac{K_{i+1}^{\frac{1}{2}}}{K_i^{\frac{1}{2}}}\theta_i \tag{5.14}$$

where K_i is the original curvatures, K_{i+1} is the curvature of the inflated spheres, θ_i is the initial angle on the original sphere.

In the next step, we compute new geodesic distances for the points on the inflated hypersphere. Reprojection under the log map preserves the geodesic distances to the origin of the tangent space. However, the geodesic distances between points are modified by the inflation and reprojection. The updated geodesic distances on the inflated hypersphere can be computed using the new co-ordinates on the inflated hypersphere. The update equation for the geodesic distance between point m and p on the inflated hypersphere is

$$d_{Gmp} = r_{i+1}\theta_{i+1} = \frac{\cos^{-1}(\langle \mathbf{y}_{m_{i+1}}, \mathbf{y}_{p_{i+1}} \rangle K_{i+1})}{K_{i+1}^{\frac{1}{2}}}. \tag{5.15}$$

This is how the geodesic distance between pairs of objects internal to the patch are updated using Ricci flow and tangent plane to evolve the patch locally on a spherical manifold. The geodesic distance between pairs of objects external to the patch and objects internal to the patch are approximated by adding the geodesic distances over the edge-connected path between the objects.

5.2.4 The Algorithm

Given a set $Y = \{\mathbf{y}_1, \cdots, \mathbf{y}_N\}$ of N objects and a dissimilarity measure d, a dissimilarity representation is an $N \times N$ matrix D_G with the elements $d_G(u, v)$ representing the pairwise geodesic distance between objects y_u and y_v. Figure 5.1 demonstrates the algorithmic steps. The following algorithmic steps can be used to perform Euclidean rectification of the distance matrix, and suppress its non-Euclidean artefacts:

1. Construct a local patch for a second order KNN graph, consisting of the first and second neighbours of the reference object.

2. Perform hyperspherical embedding to obtain the initial curvature and the coordinates of the objects in the local patch.

3. Update the curvature with a small time step derived from Equation 4.21 in Section 4.3.

4. Obtain the new coordinates on the inflated hypersphere using Equation 5.13.

5. Obtain the new geodesic distance matrix $d_{G_{i+1}}$ for the local patch using Equation 5.15. The distances between pairs of objects external to the patch are approximated using the old dissimilarity matrix. The geodesic distance between pairs of objects external to the patch and objects internal to the patch are approximated by adding the geodesic distances over the of edge-connected path between the objects.

6. Obtain the updated global distance matrix $D_G^{(1)}$ containing rectified geodesic distances between objects, and repeat from step 1 until D_G stabilises, i.e. there are no further decreases in the negative eigenfraction. Ideally, the centralized Gram matrix should have no negative eigenvalues.

We have shown how to evolve a patchwise hyperspherical manifold so as to rectify such artefacts in a dataset. The method uses a tangent-space reprojection method to inflate the local hyperspherical patches, while aiming to maintain the consistency of the pattern of geodesic distances.

5.3 Curvature Estimation from Distances with Least Square Fitting

Previously, we showed three ways for correcting non-Euclidean dissimilarities using the Ricci flow. During the Ricci flow process, the numeric estimation of sectional curvatures is a key component. In the piecewise Ricci flow algorithm, we estimate the curvatures in terms of the Euclidean distances and the geodesic distances. The Euclidean distance is estimated from the distance preserving embedding: the kernel embedding and the Isomap embedding. Experimental results in Chapter 4.6 show that the curvature estimation is affected by the embedding and the curvature could be extreme. This curvature estimation is improved to some extent by the heat kernel process during the regularised Ricci flow algorithm. In the tangent space reprojection, the curvature is estimated from the spherical embedding. To remove the effects of the embedding methods on sectional curvature, in this section we shows a way to compute the local sectional curvature from distances only. We model the local manifold with constant curvatures as a portion of hypersphere and find the sectional curvature in terms of the volume and the dimensionality of the space in a simplified form. We first estimate the dimensionality of the space by using the least squares method and then use the kernel density estimation to calculate the accurate volume for computing the sectional curvatures.

We consider objects of interest are on a curved manifold and the elements from the input non-Euclidean dissimilarities are the geodesic distances. The curved manifold thought to consist of local patches with constant sectional curvature. Our aim in this section is to estimate the curvature in the local neighbourhood from the distances. We model the local patch as part of a hypersphere and simplify the relationship of the volume and the local patch by using a local Taylor expansion. We find out that the volume of the local patch is determined by the dimensionality, the curvature and the enclosing radius of the local patch. As the given inputs are only pairwise distances between objects, we do not know the dimensionality of the manifold (patch), on which the objects lie. In order to estimate the dimensionality, we make two assumptions. Firstly we assume that the points in the neighbourhood are close enough, and the enclosing radius of the local patch is small; thus we are able to simplify the computation of the volume by using its first order expansion (of the volume). Secondly, we assume that the points in the neighbourhood are equally spaced, so we can infer the dimensionality with the number of points in the neighbourhood. After we find out the dimensionality, we can calculate the volume by using the kernel density estimation and compute the curvature by least-squares in terms of the volume and the dimensionality.

5.3.1 Kernel density estimation

To estimate the volume of the local patch, we need to estimate the probability density of the data. Although we have the distances, the data distribution is not given. Kernel density estimation is a non-paramtric density method and is frequently employed because it does not need to assume the form of the data distribution [21]. Let $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$ be sample point in the d-dimensional space from an unknown density, then its kernel estimate on the density is

$$f(\mathbf{x};\sigma) = \frac{1}{n\sigma^d}\sum_{i=1}^{n} K(\frac{\mathbf{x}-\mathbf{x}_i}{\sigma}) \tag{5.16}$$

where $K()$ is the kernel function, σ is the smooth parameter, also called the bandwith. There are a various choices among kernel functions such as Gaussian, triangular and rectangular kernel. A kernel function computes the contribution of an individual sample to the overall density. Here we choose the smoothing Gaussian function as the kernel

$$K(\frac{\mathbf{x}-\mathbf{x}_i}{\sigma}) = \frac{1}{(\sigma\sqrt{2\pi})^d}\exp\left[-\frac{|\mathbf{x}-\mathbf{x}_i|^2}{2\sigma^2}\right] \tag{5.17}$$

Since the distances are given, then the density for point can be computed as: $\mathbf{x}_m$ is

$$p_m = \frac{\sum_{i=1}^{n}\exp(-D^2(m,i)/2\sigma^2)}{n(\sqrt{2\pi})^d} \tag{5.18}$$

There exists extensive studies on how to optimally choose the kernel width, such as parametric methods and heuristic estimates [105]. For this work, we simply fix σ to be the squared nearest-neighbour distance by multiplying the maximum squared distances with a very small constant. We randomly choose a small constant 0.07 here, so $\sigma_m = 0.07 * \max D(m,:)$. We prefer the Gaussian kernel for two reasons. Firstly, the Gaussian kernel only requires one coefficient, the scale variable σ, to determine the kernel function. Second, using distances is straightforward, since our data is represented by pairwise distances (not in vector form).

5.3.2 Enclosed Volume as a Function of Curvature

Locally, the manifold is of constant curvature K. We start with the simple two dimensional surface of a sphere, the volume element is

$$d_{S^2}V = r^2 \sin\theta d\theta_1 d\theta_2 = K^{-1} \sin\theta d\theta_1 d\theta_2 \tag{5.19}$$

Without considering the ambient space, the objects of interest are considered to lie on a small portion of a sphere abstracted by the local neighbourhood graph on the sphere surface. The shape of the local sphere in three-dimensional Euclidean space (the sphere with a polar angle θ_2 fixed) is like an icecream cone. Let the surface radius (the top of a cone) $R = r\theta_2 = \frac{\theta_1}{\sqrt{K}}$, then the polar angle will be $\theta_1 = R\sqrt{K}$. Thus the volume of the local patch with the surface radius R can be computed as:

$$\begin{aligned} V_{S^2} &= K^{-1} \int_{\theta_1=0}^{R\sqrt{K}} \int_{\theta_2=0}^{2\pi} d_{S^2}V & (5.20) \\ &= 2\pi K^{-1} \int_{\theta_1=0}^{R\sqrt{K}} \sin x dx & (5.21) \end{aligned}$$

Now if we extend this to high dimensionality, the volume element of a d-sphere is

$$d_{S^d}V = K^{-d/2} \sin^{d-1}\theta_1 \sin^{d-2}\theta_2\theta_2 \cdots \sin\theta_{d-1} d\theta_1 d\theta_2 \cdots d\theta_d \tag{5.22}$$

and therefore the volume of a local patch of radius R is

$$
\begin{aligned}
V_d &= K^{-d/2}\int_{\theta_1=0}^{R\sqrt{K}}\int_{\theta_2=0}^{\pi}\cdots\int_{\theta_d=0}^{2\pi} d_{S^2}V && (5.23)\\
&= 2\pi K^{-d/2}\left\{\prod_{k=1}^{d-2}\int_0^{\pi}\sin^k xdx\right\}\int_{\theta_d=0}^{R\sqrt{K}}\sin xdx && (5.24)
\end{aligned}
$$

By integration,

$$
\int_0^{\pi}\sin^k xdx = \begin{cases} \frac{(k-1)(k-3)\cdots 1}{k(k-2)\cdots 2)}\pi & k \text{ is even} \\ \frac{(k-1)(k-3)\cdots 2}{k(k-2)\cdots 3)}\pi & k \text{ is odd} \end{cases} \quad (5.25)
$$

so replace the product of the above equation with the usual Gamma function, we can find that

$$
V_d = \frac{2\pi^{d/2}K^{-d/2}}{\Gamma(d/2)}\int_{\theta_d=0}^{R\sqrt{K}}\sin xdx \quad (5.26)
$$

The approximation to the volume in the local neighbourhood of a point can be simplified by using a local Taylor expansion of $\sin x$,

$$
\begin{aligned}
V_d &= \frac{2\pi^{d/2}K-d/2}{\Gamma(d/2)}\int_{\theta_d=0}^{R\sqrt{K}}\sin^{d-1} xdx && (5.27)\\
&= \frac{2\pi^{d/2}K-d/2}{\Gamma(d/2)}\left\{\frac{K^{d/2}R^d}{d} - \frac{(d-1)K^{d/2+1}}{6(d+2)}R^{d+2}\right\} && (5.28)\\
&= \frac{2\pi^{d/2}}{d\Gamma(d/2)}R^d\left\{1-\frac{d(d-1)K}{6(d+2)}R^2\right\} && (5.29)
\end{aligned}
$$

The equation shows that the volume of the local neighbourhood encolsed by a particular radius is determined by the dimensionality of the space d and the local sectional curvature K. To find out the curvature, we must know the volume and the dimensionality of the manifold. In order to accurately calculate the volume, we must account for the non-uniform spacing, unless the points are equality distributed. Although the distances between points are given, we can not infer the volume surrouding a point unless the dimensionlarity of the space is given. Therefore the volume can not be used directly to

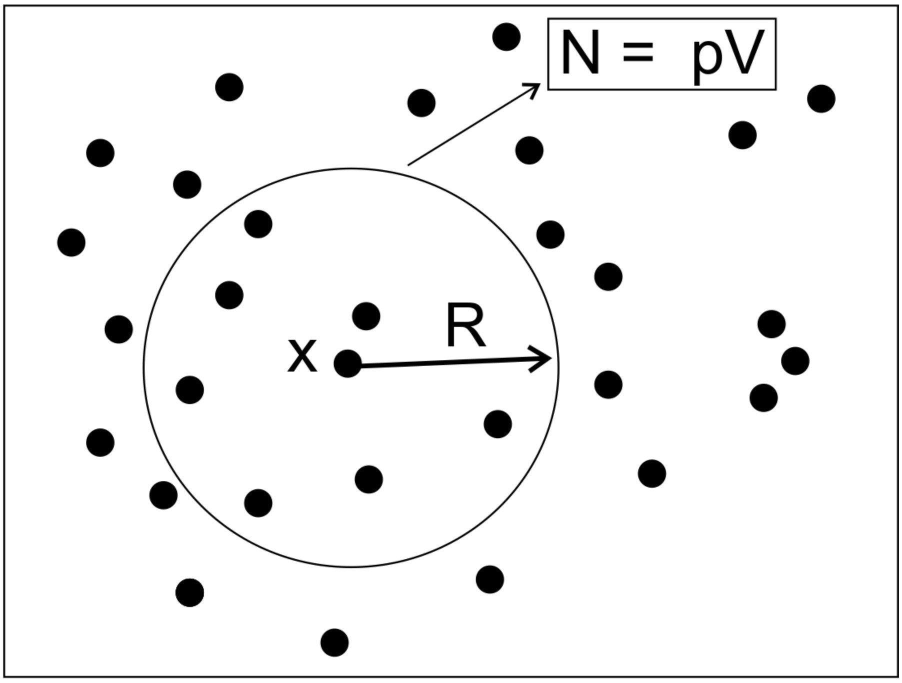


FIGURE 5.3: Illustraion of the volume estimation.

calculate the curvature without the dimensionlity.

In order to estimate the dimensionality, we need a second assumption, that the points are uniformly distributed with density p close to the point of interest illustrated in Figure 5.3. If the points are equality distributed, the number of points we expect is equal to the density times the volume. If the density is high (the points in the neighbourhood are close enough to the point of interest, the enclosing radius is small), we can use the first order expansion of the volume only

$$N = pV = \rho \frac{2\pi^{d/2}}{d\Gamma(d/2)} R^d \tag{5.30}$$

and taking logs:

$$\log N = \log \left[\frac{2\rho\pi^{d/2}}{d\Gamma(d/2)} \right] + d \log R \tag{5.31}$$

so the dimensionality of the space d is the gradient of the $\log R / \log N$ graph, but we must use only the closest points to the point of interest so that the two assumptions hold. Using least-squres, we can fit this function to find coefficients d. Please note Equation 5.30 is independent of curvature and is used to estimate the dimensionality of manifold. With this first estimation of space dimensionality, the volume of manifold can be estimated.

Once the dimensionality is calculated, we can calculate the local density of points ρ_m using a kernel density estimation. The goal of density estimation is to approximate the probability density function of a random variable. From this we can compute the volume $V = \sum_m \frac{1}{\rho_m}$. Finally taking logs of V on both size of equation Equation 5.29, we find

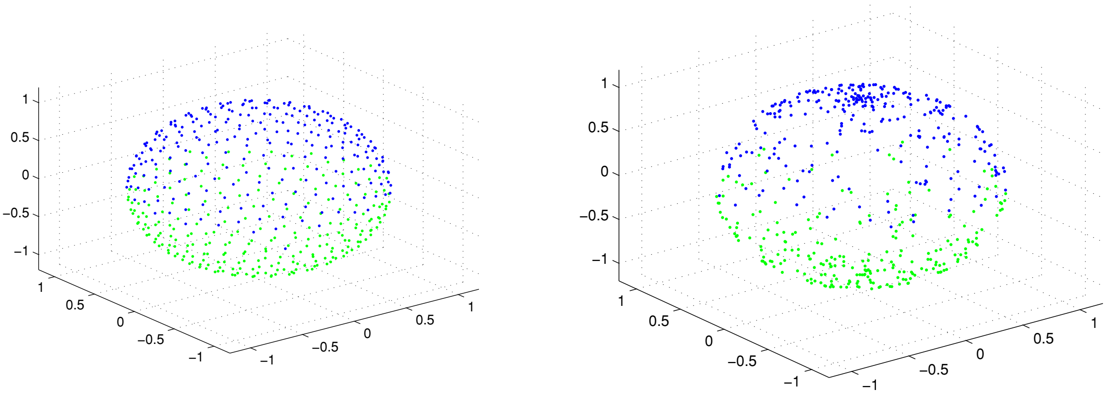
$$\log V = \log \left[\frac{2\rho\pi^{d/2}}{d\Gamma(d/2)} \right] + d \log R - \frac{d(d-1)K}{6(d+2)} R^2 \tag{5.32}$$

Using the least squares curve fitting, we can determine the first coefficient $\alpha = d$ which is the gradient of $\log R / \log V$ that best fits the data. This is the second estimation of space dimensionality. Compare to the first estimation based on Equation 5.31, the second estimation is likely to be better than the first because of the greater amount of data. The first estimation is used for calculating the volume from kernel density which is more accurate than using the number of equally-spaced points locally in Equation 5.30. From the second estimatation of the space dimensionality, we can determine the curvature K from the second coefficient $\beta = -\frac{d(d-1)K}{6(d+2)}$.

5.4 Experiments

In previous section, we present implementation details for tangent space reprojection, which is extended Ricci flow on constant curvature of elliptic background geometry for correcting non-Euclidean dissimilarity. We have applied the tangent space reprojection on the Chickenpieces dataset to demonstrate its applicability.

We also explore the geometric properties of the non-Euclidean dissimilarity data by



(A) Uniformly distributed

(B) Randomly distributed

FIGURE 5.4: Sphere data.

estimating the sectional curvature of the manifold, on which the objects in terms of dissimilarity reside. As the distances of objects on sphere are metric but non-Euclidean, and the sphere is one of few geometries of which the sectional curvature is known and easy to simulate, we use geodesic distances of objects sampled on 2D sphere to test our curvature estimation technique. We thoroughly tested our algorithms on the dissimilarity data, whose elements are geodesic distances among points lying on a 2D sphere surface with two different distribution, namely random distribution and uniform distribution, for estimating the curvature of the manifold from distances only.

Sphere data The coordinate of a point on a sphere with radius R is $\mathbf{x} = (r \sin \phi \sin \theta, r \cos \phi \sin \theta, r \cos \theta)^T$. We generate the spherical coordinates for a set of n point which are uniformly distributed over 2D sphere by generating n uniformly distributed values respectively for $p \in (0, 1)$ and $q \in (0, 1)$, let $\phi = 2\pi p, \theta = \cos^{-1}(2q - 1)$ [121]. For generating the spherical coordinates for a set of n point which are randomly (non-uniformly) distributed over 2D sphere, we generate random values for θ and ϕ from the uniform distribution on the intervals $\phi \in [0, 2\pi], \theta \in [0, \pi]$. Then the geodesic distance of pairwise points $d(u, v) = r \cos^{-1}\langle \mathbf{x}_u, \mathbf{x}_v \rangle$ is regarded as the weight of edge between u

and v. In the experiments, We use six sets of dissimilarity matrices, whose elements are the pairwise geodesic distances among 500 points sampled on a unit sphere which centers at the origin illustrated at Figure 5.4.

5.4.1 Experiments on Tangent Space Reprojection

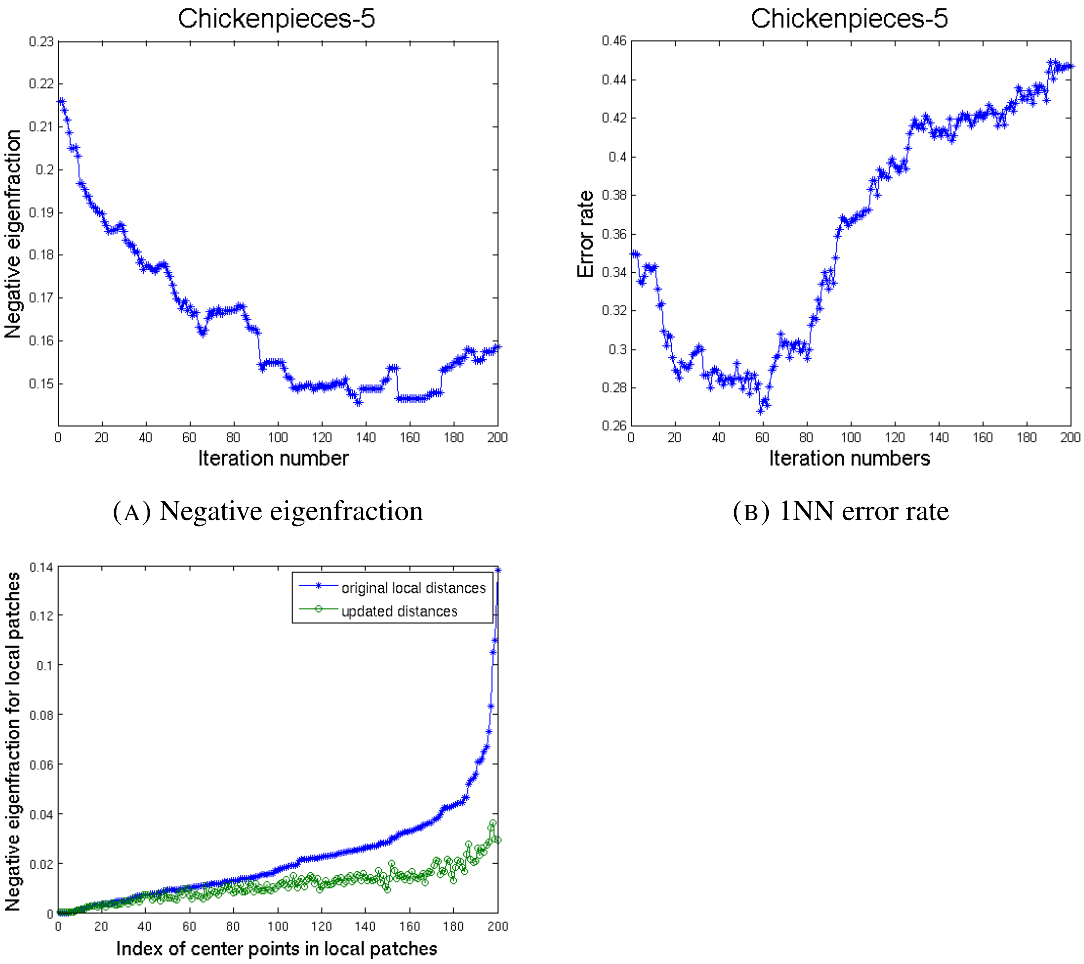
In tangent reprojection, we model the manifold consisting of local patches on hypersphere and aim to smooth out the manifold by flattening local patches. We have one set of distances on local patches, representing distances between objects internal to the local patches and the original distance matrix, which represent geodesic distance between objects on the manifold. We apply the tangent reprojection on a set of distances from the Chickenpieces data ($L = 5, \text{cost} = 45$). Figure 5.5 displayes the experimental results. It shows the negative eigenfraction of all distances, the 1NN error rate and the negative eigenfraction of distances from the local patches as the the distance matrix is evolved under the tangent reprojection. Figure 5.5 (C) shows that the negative eigenfraction for distances between points on each local patch is decreased in each iteration, indicating the evolution has flattened the local patches. The negative eigenfraction for the whole distance matrix drops from 22% to 15% and then increases again illustrated in Figure 5.5 (A)(B), indicating that the evolution has succeed in flattening the manifold, but then deteriorates. The deterioration is caused by the path-based approximation of geodesic distances between internal and external objects on the local patches. Each local patch is nearly Euclidean after the tangent space Ricci projection, we use the geodesic distances for the shortest path to connect the local patches, which makes the global patch more curved. These distances have been inflated more rapidly than the local distances on the hyperspherical surface. This exaggerates the overall curvature. With more points, the intrinsic geodesic distances between patches can be better approximated by the graph distances and becoming more accurate. The accuracy depends on the geometric structure of the manifold and the density of points. We expect a global manifold can be flattened by locally flattening curved patches and connecting flattened patches given enough points.

5.4.2 Investigation on The Curvature Computation from Kernel Embedding and Isomap Embedding

Since the previous experiments in Chapter 4 show that the embedding methods affect the magnitude of curvatures, we first investigate the effects of the kernel embedding and the Isomap embedding for estimating curvature in the piecewise Ricci flow embedding. In the next step, we present the experimental results of the least square fitting method introduced in this Chapter.

We apply the curvature estimation techniques from the kernel embedding and Isomap embedding, which are used in the piecewise Ricci flow embedding on a sphere dataset. The sphere datasets is a 500×500 dissimilarity matrix, whose elements are the pairwise distances among 500 points which are uniformly distributed on a half sphere. Figure 5.6 plots all pairs of distances between 500 points. For the kernel embedding, the distance matrix has negative eigenvalues. The kernel embedding removes the eigenvectors corresponding to the negative eigenvalues, thus the embedded Euclidean distances is always larger than the geodesic distances, and the sphere geodesic distances are larger than the original Euclidean distances, this is why the embedded Euclidean distances are larger than the original Euclidean distances which is illustrated in Figure 5.6 (A). This makes the curvatures used in the piecewise Ricci flow embedding based on equation 4.24 in previous Chapter are always negative, however the curvatures on the unit sphere should be positive.

Another fact is that the kernel embedding preserves the distances of the distant points, thus the curvatures of distant points have smaller magnitude than that of local points, that is, the distant edges are less curved than the local edges. During the Ricci flow process, the stronger the curvature, the faster the distance is extending (for negative curvature and contracting for positive curvature). Thus, after some iterations of the Ricci flow, the corrected distance of the local points exceed that of the distant points, which destroy the

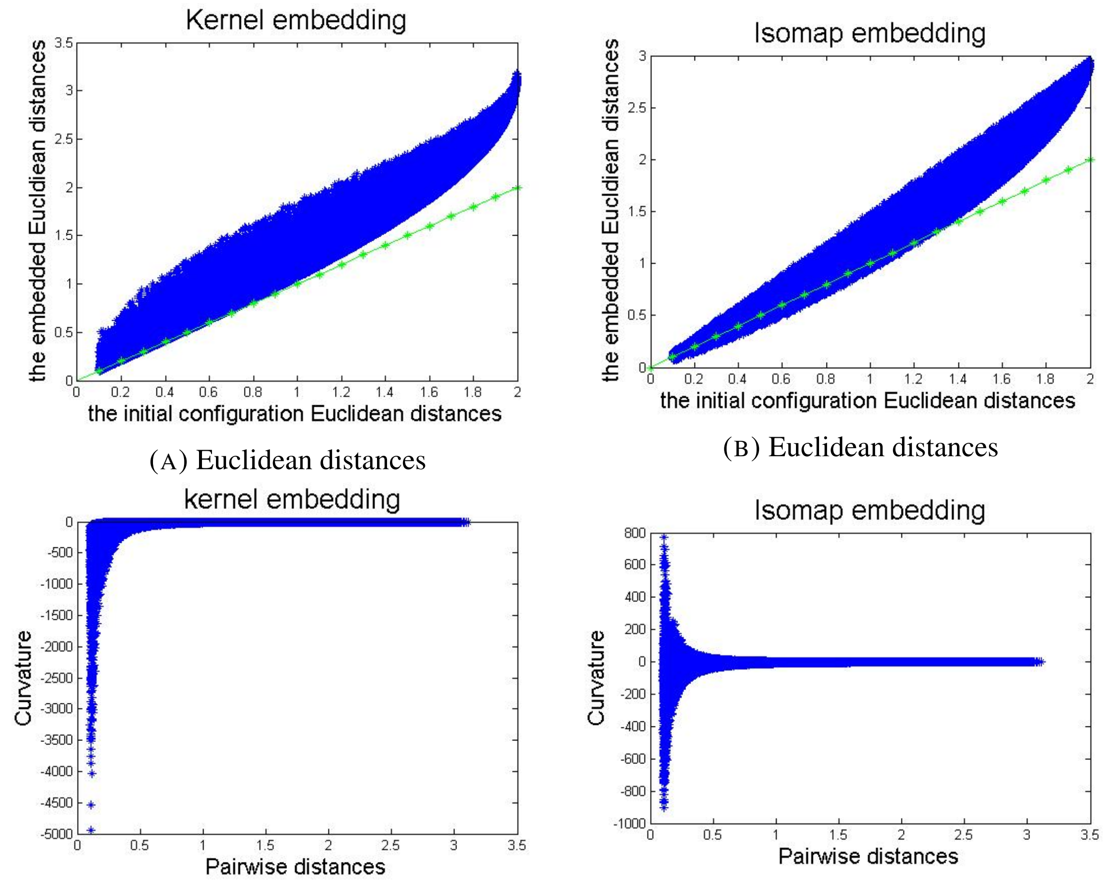


(A) Negative eigenfraction

(B) 1NN error rate

(C) Local negative eigenfraction

FIGURE 5.5: (A)(B)(C) show the negative eigenfraction of all distances, the 1NN error rate and the negative eigenfraction of local patches during the iteration of the tangent space reprojection for the Chickenpieces-5.



(A) Euclidean distances

(B) Euclidean distances

(C) Curvature

(D) Curvature

FIGURE 5.6: (A) (B) show the embedded Euclidean distances with the kernel embedding and the Isomap embedding as a function of the Euclidean distances based on the initial configuration; (C) (D) show the curvatures using the kernel embedding and the Isomap embedding respectively as a function of ordered pairwise distances between pairs of points which are uniformly distributed on half of a unit sphere.

local structure of the original distances.

For the Isomap embedding, the hopped distance obtained from the neighborhood graph of the dissimilarity data preserve the geodesic distances of local points. But after performing MDS (i.e. kernel embedding) on the hopped distance, the distances between pairs of close points will be stretched (increased), since the kernel embedding preserves the distances of the distant points. In the end, most of the embedded Euclidean distances would be still larger than the geodesic distances. This makes the shape of the curvatures distribution computed by the Isomap embedding similar to that obtained by the kernel embedding illustrated in Figure 5.6 (C)(D). Hence part of the curvature estimated from the Isomap embedding are negative as well. During the Ricci flow evolution process, the same reason here destroys the local structure as the kernel embedding. Figure 5.6 indicates that the curvature estimation does not give the accurate curvature of the manifold. This is because the Euclidean distances between local points given by the two embedding schemes are not exactly the embedded Euclidean distances (in 2-sphere manifold, the embedded Euclidean distances are the distances in a 3D Euclidean space) illustrated in 5.6 (A)(B).

5.4.3 Experiments on Curvature Estimation with Least Square Fitting

We have applied our curvature estimation technique based on the least square fitting method to six sets of dissimilarities. These dissimilarities are constructed such that their elements are pairwise geodesic distances between points, which are lying on a unit 2D sphere, on half of a unit sphere and on a quarter of a unit sphere respectively, with two different distributions, namely uniform distribution and random distribution on the sphere. The actual curvature on the sphere is 1. Figure 5.7 shows the estimated radius using the least square curve fitting of volume method from these sphere data. This method gives radius as 0.9865 for distances between objects uniformly distributed on a unit sphere, 1.007

for distances between objects uniformly distributed on half of a unit sphere, and 188.281 for distances between objects uniformly distributed on a quarter of a unit sphere. This shows good approximation for points uniformly distributed on the whole sphere and half sphere, but for points on the sphere portion less than half sphere, the estimated radius is very inaccurate. It might be because the local patch is small and nearly flat, so a much bigger radius like 188.281 fits it. For the distances from the randomly distributed points on the sphere, the approximation of radius is far from the true radius, which is not accurate. This is because the density in small neighbourhoods is not uniformly distributed, which makes an accurate numeric estimation of the space dimensionality impossible. In our assumption, we do not require globally uniformly distributed but require locally uniformly distributed for estimating the dimensionality of manifold. The inaccurate estimation of dimensionality leads to inaccurate density estimation. The dissimilarities are distances of datapoints randomly sampled from a sphere, thus the second requirement for locally uniformly distributed datapoints is not satisfied. If we add Gaussian noise, the estimated dimensionality is expected to be further distorted by the addition of noise. The calculation for the manifold volume depends on the density. Thus the curvature estimated by fitting the volume and the diameter using the least square fitting is not working properly for randomly distributed points on the sphere. If the underlying dimensionality of the manifold is known, the volume would be estimated more accurate and then the least square fitting method would work. But it is still a challenging problem to find true dimensionality of manifold from only pairwise dissimilarities. There exists methods like Isomap which attempts to predict the dimensionality of manifold. It would be interesting if we combine some techniques for estimating the space dimensionality with our least square fitting method for estimating curvatures of manifolds in future work.

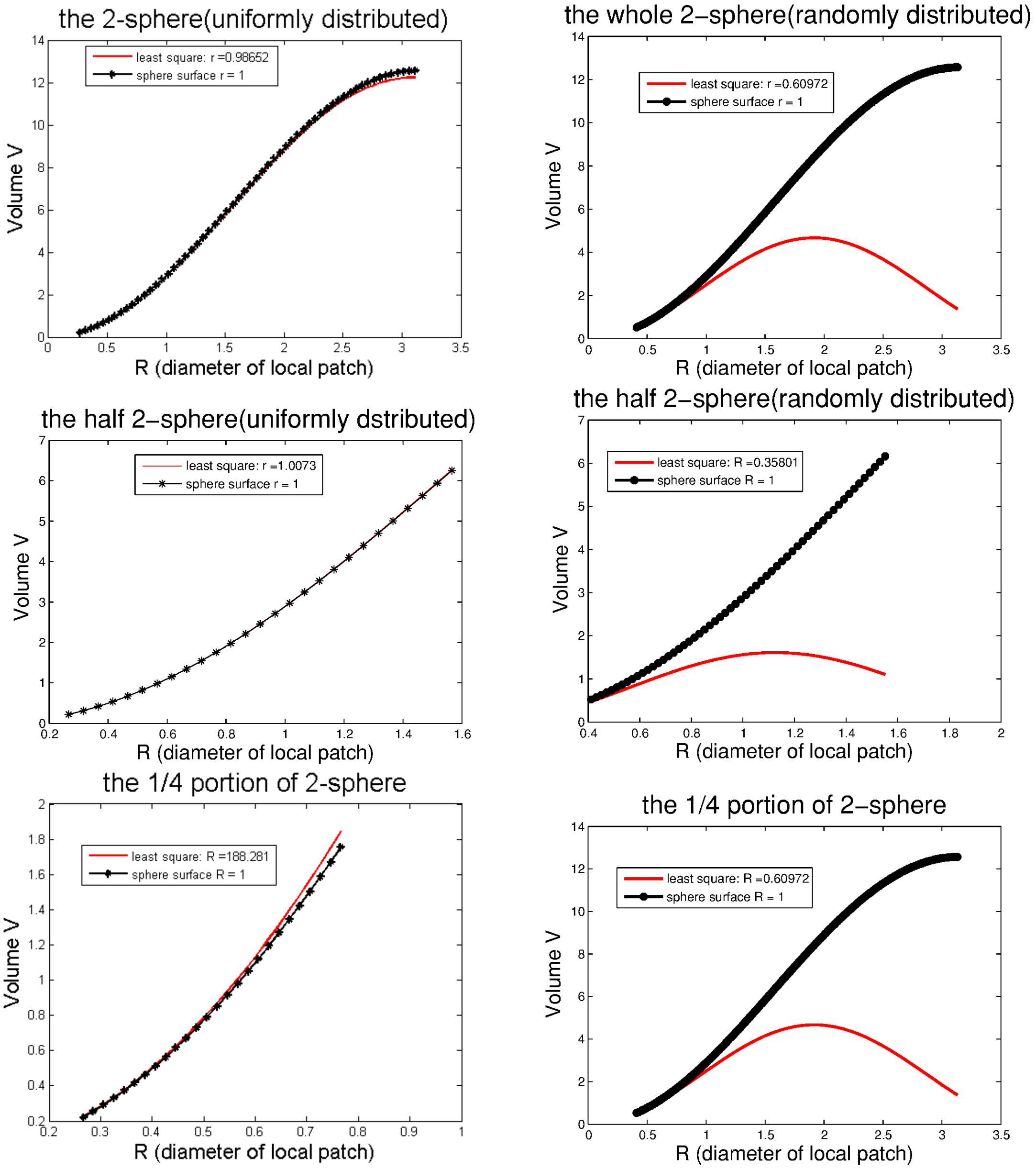


FIGURE 5.7: Left column shows the fitting curves of sphere patch from uniform sampling, right column shows the fitting curves of sphere patch from random sampling.

5.5 Summary

In this chapter, we extend the Ricci flow embedding framework and develop a new method, the tangent space reprojection, to rectify a set of non-Euclidean dissimilarities to make it more Euclidean. We showed in previous chapter that the non-Euclidean artefacts can be rectified by evolving the curved manifold, on which the objects represented by dissimilarities reside. We assume that the manifold is made of individual patches with constant sectional curvature. Following the same idea, the tangent space projection further develop the Ricci flow embedding framework by extending the construction of a local patch from an edge in previous techniques to a 2-degree neighbourhood graph. The experimental results show that the distances can be corrected to some extent, indicating that the tangent reprojection is not able to correct non-Euclidean dissimilarity into a set of Euclidean dissimilarity like the piecewise Ricci flow embedding and the regularised Ricci flow embedding. This is caused by the path-based approximation of geodesic distances between internal and external objects on the local patches.

To remove the effects of the embedding methods on sectional curvature, in this section we show a way to compute the local sectional section using the least square fitting from dissimilarties only. We apply our curvature estimation technique to the sphere data. The experiments shows good approximation for the curvature of the manifold when dissimilarities are distances between pairs of points, that are uniformly distributed on the whole sphere surface and half of sphere surface, but is not accurate for the manifold where points covers on a quarter of sphere surface. It might be because the local patch is small and nearly flat, hence it is hard to find a good fit for the volume and the curvatures. It is impossible to estimate the curvature of a manifold where points are randomly distributed, because the data points are not necessarily locally uniformly distributed which is against the second assumptions in our least square fitting method for computing curvature from distances only..

Chapter 6

Conclusions

Dissimilarity is a fundamental way to represent relations between objects. Sometimes it is the only available representation in pattern recognition when meaningful features are difficult to extract. This thesis studies several aspects of non-metric or non-Euclidean dissimilarity data, which violates the metric or Euclidean requirements. Our interest focuses on the origins of non-metric dissimilarities and subsequent data correction for metric but non-Euclidean dissimilarity data.

The dissimilarity data in pattern recognition applications is rarely metric or Euclidean. Only a few special classifiers are available to analyse non-Euclidean data, while lots of powerful traditional learning classifiers are available for vectorial data. How should we deal with non-metric or non-Euclidean data? Are the metric violations informative? Should we correct (embed) them into Euclidean distances so that they are suitable for the traditional geometric classifiers? Can we find corrections without distortion?

This thesis may answer the first question. To deal with the non-metric dissimilarity data, we analyse the sources of the metric violations and then perform corresponding data analysis. For non-metric data which arises from Gaussian noise, there is no direct solution to denoise the data. If its negative eigenvalues are informative for classification, Ricci flow embedding can be applied on it and the resulting Euclidean dissimilarity data might preserve relevant information. For non-metric data which arises from a few outlier objects, we can remove these noisy objects to restore the Euclideanness of the dissimilarity matrix. For metric but non-Euclidean data which resides on a curved manifold, we can use Ricci flow embedding to rectify them into Euclidean space. Our interest mainly focuses on the Ricci flow embedding. The overall hypothesis presented in this thesis was that for the non-Euclidean dissimilarity embedded on a curved manifold with constant curvature, Ricci flow embedding would correct them into Euclidean distances by flattenning the manifold. To test this hypothesis, a series of experiments were executed on two sets of dissimilarity data from two public datasets in pattern recognition applications. These experiments were designed primarily to correct the non-Euclidean distances, but also to explore and improve our techniques so that the rectified distances preserve the group structure of non-Euclidean data. The results of these experiments are summarized in the following section, and indicate that the piecewise Ricci flow embedding and the regularised Ricci flow embedding transform non-Euclidean distance into Euclidean distance successfully. The tangent space projection corrects non-Euclidean distances to some extent because of the edge-based estimation for distances between different patches. The work also demonstrates that piecewise Ricci flow embedding distorts local structure of data because the Euclidean embedding effects on individual edges. This problem is addressed to some extent by the regularised Ricci flow embedding. The experiments shows the regularised Ricci flow embedding preserves local structure of data but is still affected by the curvature estimation from the Euclidean embedding. We explore the tangent space representation and curvature estimation to improve the Ricci flow embedding so that the relevant information hidden in the metric violations are not lost during the correction process.

The second and fourth questions has been answered in the literature review. The non-Euclidean violations are informative if the classification results deteriorates by removing the non-Euclidean artefacts. For such data, the correcting procedures which ignore the non-Euclidean characteristics works counter-performance. The Ricci flow embedding tries to preserve the useful information in the Euclidean violations by evolving the geodesic distances in small step for each iteration. Although it is impossible to embed non-Euclidean dissimilarity data into vector space without distortion, it is possible to find a set of vectors such that the local structure of data is preserved by the regularised Ricci flow embedding. Although individual edge is still affected by the Euclidean embedding methods (the kernel embedding and the Isomap embedding), the regularised Ricci flow embedding can locate a set of Euclidean distances, which gives similar or better classification performance than the original non-Euclidean distances for the two datasets in our experiments.

The third question is still an open discussion. Our guideline is that if the correcting procedure removes useful information for classification, it is better to use the original non-Euclidean data and design special classifier for the non-Euclidean data instead of correcting them. If the correcting procedure preserves such information, it is optimal to correct the data so that many traditional learning tools are available for analysing them.

6.1 Contributions

This thesis includes the novel ideas on the sources detection and the Ricci flow embedding framework and their related application in real word problems. It has made the following contributions to the dissimilarity based recognition:

- For analysing the underlying causes of the metric violations, a simple empirical test is introduced by characterising the shape of the spectrum of the negative eigenvalues of the corresponding Gram matrix and the additive constant required to render it metric. We use this method to investigate the possible sources of non-metric behaviours on various non-metric or non-Euclidean dissimilarity data. Besides this, a new measure was introduced that assesses the contribution of each object to the mass of negative eigenvalues. We proved experimentally that this measure is effective in identifying outlying objects (Chapter 3).

- For rectifying the non-Euclidean dissimilarity matrix, a framework based on the Ricci flow embedding is introduced. We map the dissimilarity matrix to the curved manifold whose metric is the pairwise distances. Non-Euclidean dissimilarity can be rectified by flattening the curved manifold. This is achieved by evolving the metric based on the Ricci flow. We start with piecewise manifold and explore two Euclidean embedding methods for approximating the curvature of the manifold. We proved experimentally that this piecewise embedding can correct non-Euclidean dissimilarity into Euclidean distances effectively (Chapter 4)
- For preserving the local structure of data, a regularised Ricci flow embedding method was introduced. The experiments show it can stabilise the curvature in local neighbourhood and find a set of Euclidean distances with good classification results, similar to the classification results of the original non-Euclidean data. This indicates regularised Ricci flow not only rectify non-Euclidean distances into Euclidean space effectively, but also retains the local group structure lost in other alternative Euclidean correction techniques (Chapter 4).
- We combine the Ricci flow with the tangent representation to introduce a new correcting procedures. The curvature is estimated from spherical embedding. The experimental results show the technique can remove the non-Euclidean effects to some extent and preserve the local structure of data (Chapter 5).
- Curvature estimation is a key component in our Ricci flow framework. We observe that the computed curvature is affected by the Euclidean methods we used for computing the Euclidean distance on each edge and these two Euclidean methods do not give accurate Euclidean distances based on the experimental results on artificial dataset. We explored a way to compute the curvature of a manifold the objects

reside on without knowing the dimensionality, given only pairwise distances. The method using the volume as a function of radius was promising. But this works only well with uniformly distributed and dense points (Chapter 5).

6.1.1 Detecting the Sources of Negative Eigenvalues

The first contribution of this thesis is to model the three sources of negative eigenvalues and extract effective measures to distinguish the three sources on modeled dissimilarity data. All the non-Euclidean dissimilarities are transformed into Euclidean space through some embedding techniques before analyzing them using traditional geometric learning tools. The positive subspace embedding assumes that metric violation are an artifact of noise and the distance in the negative subspace do not carry any signification discriminative information. Some researchers [85, 86, 72, 71] have showed that negative eigenspace can contain valuable information for classification. Euclidean correction can lead to poor classification results by ignoring the negative space. Thus before using the correction approaches, it is necessary to analyse the underlying causes of negative dissimilarity eigenvalues and choose the suitable correction methods.

We think of three case where non-Euclidean dissimilarity arises which allow us to model. Firstly, if the data points reside on a curved manifold, the distance between them are intrinsically non-Euclidean. We model such data as points on the surface of a sphere, a simple surface where distances are easy to compute. Secondly, if the objects are not point-like but rather are extended in space, then the distance between them are measured between the closest points on their surface, thus non-Euclidean and possibly non-metric. Finally, if the Gaussian noise is added to the original Euclidean dissimilarities, the noisy Euclidean dissimilarity is non-Euclidean and non-metric. The Gram matrix of non-Euclidean dissimilarity data is indefinite, while the Gram matrix of Euclidean dissimilarity data has non-negative eigenvalues. It means there is negative eigenvalues. It is the existence of negative eigenvalues that make it impossible to do distortion-free embedding in Euclidean space. It is natural to analyse the negative spectrum of the Gram matrix. We commence by examining the shape of the negative spectrum of the Gram matrix of non-Euclidean dissimilarity from the three sources. And we find out that the negative spectrum under the three models follows an exponential decay. The spectrum of the non-Euclidean dissimilarities from the sphere and balls datasets exhibits a strongly falling negative tails, while the non-Euclidean dissimilarities from Gaussian noise is characterised by a more slowly decreasing negative spectrum. The slope and intercept extracted from an exponential fit should be able to discriminate at least the Gaussian noise model from the remaining two models. Moreover, the dissimilarity measure over the sphere are metric. Thus the dissimilarities under the three models can be characterized by the non-metricity, the slope and the intercept of an exponential fit. These three measures representing the non-metrcity, the slop and the intercept can be used as features for detecting the sources of the non-Euclidean behaviors. We use the three measures to investigate the origins of the negative eigenvalues of various real dissimilarity datasets. The experiments show that the simple test give suggestive origins of the non-Euclidean artefacts.

Finally in Chapter 3, we analyse the sources of non-Euclidean artefacts from the perspective of individual objects. We consider the case that the non-Euclidean artifacts are created soly by the set of distances to a few outlying objects. Then it is possible to restore the data to a Euclidean state by editing these objects from the dataset. Before restoring the data into a Euclidean state, we need a way to identify the outlying objects. Motivated by this, the notion of measuring the contribution of each object to the negative eigenfraction of a dissimilarity matrix is introduced. That is, the fraction given by the ratio of the sum of the negative distances from an individual object to each of the remaining objects, divided by the total negative and positive distances from this reference object. We demonstrated experimentally the new measure can identify the outlying objects on both artificial and real dataset.

The work described in Chapter 3 has two advantages. We modeled the three sources of non-Euclidean artifacts and extracted three measures for discriminating the possible sources, which is useful for choosing suitable correction techniques. Secondly, the novelty measure introduced to assess each object's contribution to the negative eigenvalues can provide further analysis of non-Euclidean artefacts, especially for those dissimilarities caused by a few outlying objects.

6.1.2 Ricci flow embedding

In Chapter 4 we followed the detection results of the possible sources of non-Euclidean artefacts, and developed a framework to deal with the non-Euclidean dissimilarity data when the non-Euclidean artefacts arise from manifold. This is the major contribution of the thesis. The framework of the Ricci flow embedding aims to solve two questions. The first question is how to rectify the non-Euclidean data so as to minimize the non-Euclidean artefacts and to prepare dissimilarity data ready for traditional geometry based learning classifiers. The second question is how to make the new Euclidean distances are not less discriminating than the non-Euclidean dissimilarity, that is, to preserve the group structure of data. The main idea of the Ricci flow embedding is to consider the data to be embedded as points on a curved manifold and evolve the manifold using the concept of Ricci flow so that the manifold is gradually flattened. Ricci flow provide a way to update(or evolve) the curvature on the manifold.

As mentioned earlier, a key issue in evolving distances across points in different local patches(neighbourhood manifold) is to estimate the curvature of the manifold. Based on exactly how we construct manifold and how we use these different ways to compute curvature gives rise our three algorithms.

Piecewise Ricci flow embedding

We consider each pair of objects reside on an individual manifold and the given dissimilarities to be the geodesic distances on the manifold between these points. We use a

Euclidean embedding of the points and use the difference between the geodesic distance on the manifold and the Euclidean distance in the embedded space to compute the curvature. We use two Euclidean embedding methods to find the Euclidean distance in the embedded space. The Isomap method is the first method to find a low-distortion embedding to map the objects to a low dimensional Euclidean vector space. The second method is the kernel embedding, considering only the positive part of the Gram matrix of the dissimilarity matrix. Once the curvature on the edge is known, the curvature is updated iteratively using the Ricci flow so that the curvature moves towards zero. After each iteration, the manifold is less curved than before. With the new curvature and the Euclidean distance obtained from Euclidean embedding methods, the new geodesic distance on the updated manifold is found in terms of the Euclidean distance and the curvature. Once the local patches (manifolds) are flattened, a set of final Euclidean distances is found. We use the resulted Euclidean distances to compare the classification results with those obtained using some alterative Euclidean correction techniques by performing 1NN classification using leave one out crossvalidation . Experiments on the CoilYork and the chickenpieces dataset show that curvatures are decreasing in each iteration and the piecewise Ricci flow embedding successfully correct the non-Euclidean distances to a set of Euclidean distances with some loss of discriminating power. The loss of information maybe caused by the effect of the piecewise Ricci evolution process as it is applied independently on each edge. It is also experimentally demonstrated that the better performance was obtained by using Ricci flow embedding with the Isomap embedding, where the degradation compared the original non-Euclidean dissimilarity is smaller that that with the kernel embedding.

Regularized Ricci flow embedding

The piecewise embedding manifold solve the question of how to correct the non-Euclidean dissimilarity to minimis the non-Euclidean artefacts. Though the discriminating power of the Ricci flow embedding with the Isomap embedding is better than some alternative Euclidean correction procedures, few of the informative non-Euclidean artefacts are transformed to the Euclidean form, which is indicated by worse classification results than those from the original non-Euclidean dissimilarity data. One way to improve the piecewise Ricci flow embedding is to maintain local structure during the Ricci flow smoothing process. This can be achieved by regularizing the curvatures on the local neighbourhood. We use the heat kernel graph regularization to smooth the curvatures on the edges. The heat kernel graph regularization process is similar to the flow of heat and creates a smooth distribution of heat after the heat flows from high to low concentration. Similarly this process can be used for smoothing curvatures before updating curvature with the Ricci flow. The heat kernel is to smooth function defined on each point of a graph. We need to transform the curvatures defined on edges to the curvatures defined on vertices which are corresponding to the edges. This is achieved by building a dual graph which has vertices corresponding to edges of object-pairs and edges describing a neighbourhood structure of these pairs. Once the regularized curvatures are known, we evolve the curvatures using Ricci flow iteratively so that the manifolds become flat. Experiments show the heat kernel regularization process removes some large curvature values and creates smoother distribution of the curvatures. Compared with the piecewise Ricci flow embedding, the regularized Ricci flow embedding gives better classification results for both the Isomap and the kernel embedding and has little effect on the rate of curvatures decreasing. Compared with alternative Euclidiean correction procedures, the regularized Ricci flow embedding with kernel embedding is the best, given close or even better classification results than the original non-Euclidean data.

Tangent space projection

In the piecewise manifold Ricci flow embedding and the regularized Ricci flow embedding, the Euclidean embedding methods of the Isomap and the kernel embedding during the Ricci flow smooth process disrupts the group structure of data. Regularisation process is one way to maintain the local structure. Another way is to remove the dependence on the Euclidean embedding methods and use the concept of tangent space representation to capture local structures and the concept of spherical embedding to estimate curvatures. We proposed a tangent space based embedding method. We consider a second order of k-NN graph as a manifold (local patch). Then spherical embedding is performed on individual local patch for obtaining the curvature of the local patch and the object (point) position vector in the local patch. Ricci flow is used to reduce the curvature of local patch. The points in the original patch are mapped to its tangent plane by using the log map and then the points on the tangent plane are mapped back to the new local patch which is slightly inflated with reduced curvature by using the exponential map. Our aim is to flatten the global manifold by gradually smoothing out the local patches. The mapping process with tangent plane preserves the distances from reference point to the other points on the manifold. The tangent plane mapping during the Ricci flow process updates the geodesic distances among points on the same local patch. The question of how to estimate the distances for pairs of objects, one is on the local patch and the other object is external to the local patch arises. The two objects are in different local patches with different sectional curvatures, that is, locally they are in different coordinate systems, the global coordinate system is not known except the geodesic distances on different patches are given. Thus we use the shortest edge-connected path to approximate the geodesic distances between them. the local patches. For the first method, After a local patch is smoothed out during each iteration, the whole set of distance matrix is updated in this way so that the new geodesic distances for local patch is obtained by the Ricci flow evolution, the distances between pairs of objects external to the patch are approximated using the old dissimilarity matrix before evolving the local patch, the geodesic distances between pairs of objects external to the patch and the objects internal to the patch are approximated by adding the geodesic distances over the edge-connected path between the objects. This is the tangent space projection. method, the geodesic distances between pairs of objects external to the patch and the objects internal to the patch are not considered until all the local patches have been updated, we update the local patch one by one. This is called on-line tangent space projection.

The experiments on chicken pieces dataset shows the tangent space projection is able to minimize the non-Euclidean artefacts to some extent, but then deteriorates. This is caused by the path-based approximation of geodesic distance between internal and external objects on the local patches. The distances have been inflated more rapidly than the local distances on the patch. shows the the online tangent projection is also able to correct the non-Euclidean artefacts to some extent, but then deteriorates. This is caused by the incorrect estimation of curvatures. After a few iterations, the overlapping edges deviates from the rest of local patch. In this case, it can not apply the spherical embedding to estimate the curvatures correctly.

Curvature estimation

In the Ricci flow embedding framework, estimating the intrinsic curvature of the manifold the set of points reside on is a very important component. In the above work, we use either the Euclidean embedding like the Isomap and kernel embedding or the manifold embedding like spherical embedding to estimate the curvature. More or less, the embedding procedures during the Ricci flow either destroys the local structure or not applicable after a few iterations, one direction of improving our current methods is to estimate the curvature correctly without these embedding procedures. A way to estimate the curvature of manifold is to use the enclosed volume of the manifold as a function of radius by modeling the embedded manifold consisting of local patches on the hyperspheres. We deduced the volume of hypersphere as a function of radius. We can find the curvature by fitting the volume. The experiments shows good approximation of curvature of manifold where points uniformly distributed on the whole sphere and half sphere, but is not accurate for the manifold where points covers less than half sphere. It might be because the local patch is small and nearly flat, hence hard to find a good fit. It is impossible to estimate the curvature of the manifold where points are randomly distributed, because the densities are hard to estimate which results in incorrect estimation of dimension and curvature. Also real dataset is hardly uniformly distributed. It is a difficult question to estimate the intrinsic dimension and curvature of manifold from a set of geodesic distances, without knowing the structure and distribution of data.

6.2 Limitations and criticisms

There are a number of shortcoming in this thesis which can be amended with further work and further explored. In Chapter 3, we have modeled the three cases with various sample size which are $100, 500, 1000$. However, some extensions of the current work still exists. The non-Euclidean violations tends to be more strong with big dataset. It would be interesting to explore the relation of the three measure in terms of the sample size. the objects are randomly generated in the three cases, it assumes normal distribution. The real datasets may not follow normal distribution. It would be interesting to start with the Euclidean distance from real dataset and modify them to make it non-Euclidean. Can we detect the sources based on the distribution of the non-Euclidean artefacts?

In Chapter 4, there are clearly a number of ways in which the work reported can be extended. For the piecewise Ricci flow embedding, we choose the Isomap and the kernel embedding to compute the Euclidean distances, as these are two classical distance-preserving method. We can explore the use of other distance-preserving methods like maximum variance unfolding (MVU) and Hessian locally linear embedding (HLLE). In the kernel embedding piecewise manifold, it is because the edges with short distances expand faster than those with big distances so that the group structure is disturbed. A possible way to overcome this problem is to propagate distances from local to global. We would explore this direction by building a minimum-spanning tree to locate local distances and update the local distances using Ricci flow and using the connected path in the minimum-spanning tree to locate the global distances. For the regularized Ricci flow embedding, we used heat kernel graph regularization to alleviate the problem of edge-curvature fluctuation, it would be interesting to explore other graph regularization methods like the p-Laplacian operator regularization.

The tangent space projection is not able to reduce the eigenfraction to zero , that is, it is able to make non-Euclidean distances less non-Euclidean but not Euclidean. The experiments demonstrates the deterioration in the middle is caused by the edge-based approximation of geodesic distances between internal and external objects on the local patches. Without the actual structure nor the dimension of the local patches (manifold), to compute the distance between points on different patches is a big challenge and is worth to explore in the future work. How to estimate curvature of manifolds without relying on piecewise Euclidean embedding and spherical embedding? We try to resolve this problem in the last section. Here we use nearest neighbour graph to build local manifolds, the neighbour graph might not match exactly manifolds, it would be interesting to use some manifold detection approaches like clustering methods for locating manifolds. Another possible future work is to develop a non-linear global alignment for locally curved patches during the Ricci flow process, various work has mapped local patches to vector space and using the linear alignment of embedded point vectors to create a global coordinates like LLE.

Finally in the last section of Chapter 4, we explore a way to compute the curvature of manifolds the data point resides on. The experiments demonstrate the method give close estimation for uniformly-distributed dense points on the sphere. But a real dissimilarity data rarely corresponds to dense points uniformly distributed on sphere. For a real dissimilarity data, we do not know the structure and density of the points on manifold, we need to infer the structure and dimension of manifold from the pairwise geodesic distances, is it possible? Given only geodesic distances between pairs of points randomly distributed on the manifold, without knowing the dimension and density, it is a challenge to estimate the curvature with high accuracy.

For the experimental evaluations in all chapters, the various datasets from the SIMBAD project in real pattern recognition applications are used. For detecting the sources of non-Euclidean artifacts, we can not evaluate our results on real datasets because no ground truth is available for real datasets. Using 2-sphere to model manifold is limited and ignores high dimension and complicated cases. Our Ricci flow embedding framework aims for objects on manifold and it is expected better classification performance through Ricci flow embedding for such data. The Ricci flow is meaningful for non-Euclidean dissimilarity data where the negative eigenvalues are informative for classification. Only CoilYork and Chickepieces datasets are such data. The source detection suggests the negative dissimilarity eigenvalues are from noise but can not be removed as they are informative for classification. This seems contradictory, though the results from Ricci flow embedding is better than ignoring the negative spectrum. The results for real datasets are inclusive. This might be caused by the limitation of simple model used for the manifold case of detecting sources and the real dataset resides on a manifold with more complicated structure.

For the Ricci flow embedding framework, the final rectified distances are constructed by using both training and testing set and no dimension (or feature) reduction is applied. In order to compare with the original non-Eulcidiean distances, all experiments are performed on the original data and only 1NN(Nearest Neighbour) classifier is used. The results are presented as leave one out crossvalidation 1NN classification errors. It would be interesting to transform the corrected Euclidean distances into vector form, and apply feature selection and advanced feature-based machine learning classifiers like SVM(support vector machine) and NM(nearest mean) to evaluate our methods. Although there is no distortion-free embedding for non-Euclidean dissimilarity and our aim is to preserve the group structure of data implied in the non-Euclidean artifacts, it would be good to measure the distortion in terms of distance-preserving. However, our embedding methods are iterative and hard to analyze the distortion in mathematical terms.

All things considered, we give answers for some questions of non-Euclidean dissimilarity data and also find interesting new questions for future work along the way. This thesis address the two major questions:

1. What is the sourers of non-Euclidean dissimilarity data?

2. How to correct the non-Euclidean dissimilarity data if the data resides on manifolds?

3. How to estimate the intrinsic curvature of simple manifolds (sphere) given the data is dense and uniformly distributed on sphere?

The hard questions for future work is as follows:

1. How to estimate the intrinsic curvature and dimension of manifolds with high accuracy without knowing the structure of the manifold, from the dissimilarity data?

2. How to estimate the geodesic distances between two points on different manifolds?

3. How to project samples outside the training set into the resulted Euclidean space?

List of Notations

D	A dissimilarity Matrix
S	A similarity Matrix
I	The identity matrix, $i_{ij} = 1$ if $i = j$, 0 else
G	The Gram Matrix
K	Sectional curvature
Φ	The Column Matrix consisting of eigenvectors
Λ	The diagonal Matrix consisting of eigenvalues
Y	The matrix containing vectors $\mathbf{y}_i$
N	Usually the number of samples in a dataset
ϕ	eigenvector of Gram matrix G
λ_i	eigenvalues of Gram matrix G
t	time
$H(t)$	Heat kernel
n	Usually the dimensionality of space
d_{ij}	The dissimilarity between object i and object j
s_{ij}	The similarity between object i and object j
d_G	Geodesic distance
d_E	Euclidean distance
$\mathbf{x}_i$	A vector indexed by i
$\mathbf{y}_i$	A vector indexed by i
$\mathbf{1}$	The vector $(1, 1, \cdots, 1)^t$
$\mathbf{0}$	The vector $(0, 0, \cdots, 0)^t$
$\langle \, \rangle$	An inner product
$\| \, \|_2$	An Euclidean norm
psd	positive semidefinite
SVM	Support Vector Machine
KNN	k-Nearest Neighbour
$1NN$	1-Nearest Neighbour

Bibliography

[1] S. Agarwal, G. Lanckriet, J. Wills, S.P. Imageworks, L. Cayton, and S. Belongie. Generalized non-metric multidimensional scaling. *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.

[2] M. Barreno. Spectral methods for image clustering. Technical report, Tech-Report CS 218B, U.C. Berkeley, 2004.

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems, 14*, 1:585–591, 2002.

[4] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.

[5] C. M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, USA, 1995.

[6] I. Borg and P. J. F. Groenen. *Modern multidimensional scaling: Theory and applications*. Springer, 2005.

[7] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8):689–694, August 1997.

[8] H. Bunke and A.Sanfeliu. *Syntactic and Structural Pattern Recognition: Theory and Applications*. World Scientific, 1990.

[9] H. Bunke, S. Günter, and X. Jiang. Towards bridging the gap between statistical and structural pattern recognition: Two new concepts in graph matching. *Advances in Pattern Recognition, ICAPR 2001*, pages 1–11, 2001.

[10] C .J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[11] L. Chen and A. Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104(485):209–219, 2009.

[12] Y. Chen, E.K. Garcia, M.R. Gupta, A. Rahimi, and L. Cazzanti. Similarity-based classification: Concepts and algorithms. *The Journal of Machine Learning Research*, 10:747–776, 2009.

[13] R. Chitta, R. Jin, T.C. Havens, and A.K. Jain. Approximate kernel k-means: solution to large scale kernel clustering. *Proc. ACM SIGKDD*, pages 551–556, 2011.

[14] B. Chow and F. Luo. Combinatorial Ricci flows on surfaces. *J. Differential Geom*, 63(1):97–129, 2003.

[15] F.R.K. Chung. *Spectral graph theory*. American Mathematical Society, 1996.

[16] J.A. Costa and A.O. Hero. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Transactions on Signal Processing*, 52(8):2210–2221, 2004.

[17] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[18] T.F. Cox and M.A.A. Cox. Multidimensional scaling on a sphere. *Communications in Statistics-Theory and Methods*, 20(9):2943–2953, 1991.

[19] T.F. Cox and M.A.A. Cox. *Multidimensional scaling*. Chapman & Hall, 2000.

[20] M. Cuturi, K. Fukumizu, and J.P. Vert. Semigroup kernels on measures. *Journal of Machine Learning Research*, 6:1169–1198, 2005.

[21] L. Devroye and L Gyorfi. *Nonparametric Density Estimation: The L1 View*. Wiley-Blackwell, New York, 1985.

[22] I.S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. pages 269–274, 2001.

[23] I.S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: spectral clustering and normalized cuts. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556, 2004.

[24] D.L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003.

[25] M.P. Dubuisson and A.K. Jain. A modified Hausdorff distance for object matching. In *IAPR International Conference on Pattern Recognition*, volume 1, 1994.

[26] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern classification*. Wiley New York, 2001.

[27] R.P.W. Duin, D. de Ridder, and D.M.J. Tax. Experiments with a featureless approach to pattern recognition. *Pattern Recognition Letters*, 18(11-13):1159–1166, 1997.

[28] R.P.W. Duin and E. Pekalska. Datasets and tools for dissimilarity analysis in pattern recognition. Technical report, Delft University of Technology, Netherlands, 2009.

[29] R.P.W. Duin and E. Pekalska. Non-Euclidean dissimilarities : causes and informativeness. *SSPR/SPR 2010*, pages 324–333, 2009.

[30] R.P.W. Duin, E. Pekalska, A. Harol, W-J. Lee, and H. Bunke. On Euclidean corrections for Non-Euclidean dissimilarities. In *SSPR/SPR*, pages 551–561, 2008.

[31] S. Edelman. Representation is representation of similarities. *Behavioral and Brain Sciences*, 21(4):449–467, 1998.

[32] H. ElGhawalby and E.R. Hancock. Graph characteristic from the Gauss-bonnet theorem. In *SSPR/SPR*, pages 207–216, 2008.

[33] H. ElGhawalby and E.R. Hancock. Measuring graph similarity using spectral geometry. In *ICIAR*, pages 517–526, 2008.

[34] H. ElGhawalby and E.R. Hancock. Characterizing graphs using spherical triangles. In *IbPRIA*, pages 465–472, 2009.

[35] B.S. Everitt and S. Rabe-Hesketh. The analysis of proximity data. *Recherche*, 67:02, 1997.

[36] C. Faloutsos and K.I. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *ACM SIGMOD international conference on Management of data*, pages 163–174. ACM New York, USA, 1995.

[37] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta. A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1):176–190, 2008.

[38] B. Fischer and J.M. Buhmann. Bagging for path-based clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1411–1415, 2003.

[39] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:377–388, 1996.

[40] L. Goldfarb. A new approach to pattern recognition. *Progress in pattern recognition*, 2:241–402, 1985.

[41] L. Goldfarb, D. Gay, O. Golubitsky, and D. Korkin. What is a structural representation? a proposal for a representational formalism. *Techn. Report*, 2006.

[42] J.C. Gower. Properties of euclidean and non-euclidean distance matrices. *Linear Algebra and its Applications*, 67:81–97, 1985.

[43] J.C. Gower and P. Legendre. Metric and Euclidean properties of dissimilarity coefficients. *Journal of Classification*, 3(1):5–48, 1986.

[44] T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer. Classification on pairwise proximity data. *Advances in neural information processing systems*, pages 438–444, 1999.

[45] X. Gu, Y. He, M. Jin, F. Luo, H. Qin, and S.T. Yau. Manifold splines with a single extraordinary point. *Computer-Aided Design*, 40(6):676–690, 2008.

[46] B. Haasdonk. Feature space interpretation of svms with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):482–492, 2005.

[47] B. Haasdonk and E. Pekalska. Indefinite kernel fisher discriminant. In *Proc. of ICPR 2008, International Conference on Pattern Recognition*, 2008.

[48] B. Haasdonk and E. Pekalska. Classification with kernel mahalanobis distance classifiers. *Advances in Data Analysis, Data Handling and Business Intelligence*, pages 351–361, 2010.

[49] J. Ham, D.D. Lee, S. Mika, and B. Schölkopf. A kernel view of the dimensionality reduction of manifolds. *Proceedings of the twenty-first international conference on Machine learning*, page 47, 2004.

[50] R.S. Hamilton. Three-manifolds with positive Ricci curvature. *J. Differential Geom*, 17(2):255–306, 1982.

[51] R.S. Hamilton. The ricci flow on surfaces. *Contemp. Math*, 71:237–262, 1988.

[52] L. Herault and R. Horaud. Figure-ground discrimination: A combinatorial optimization approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):899–914, 1993.

[53] D. Hidovic and M. Pelillo. Metrics for attributed graphs based on the maximal similarity common subgraph. *International Journal of Pattern Recognition & Artificial Intelligence*, 18(3), 2004.

[54] T. Hofmann and J.M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):1–14, 1997.

[55] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[56] L. Hubert, P. Arabie, and J. Meulman. Linear and circular unidimensional scaling for symmetric proximity matrices. *British journal of mathematical and statistical psychology*, 50(2):253–284, 1997.

[57] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

[58] R. Inokuchi and S. Miyamoto. LVQ clustering and SOM using a kernel function. *2004 IEEE International Conference on Fuzzy Systems, 2004*, 3:1497–1500, 2004.

[59] T.S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, pages 487–493, 1999.

[60] D.W. Jacobs, D. Weinshall, and Y. Gdalyahu. Classification with nonmetric distances: Image retrieval and class representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 583–600, 2000.

[61] T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *The Journal of Machine Learning Research*, 5:819–844, 2004.

[62] O. Jesorsky, K. Kirchberg, and R. Frischholz. Robust face detection using the Hausdorff distance. *Audio-and video-based biometric person authentication*, pages 90–95, 2001.

[63] X. Jiang and X. Gu. Multiscale, curvature-based shape representation for surfaces. *International Conference on Computer Vision (ICCV)*, pages 1887–1894, 2011.

[64] M. Jin, J. Kim, and X. Gu. Discrete surface ricci flow: Theory and applications. *Mathematics of Surfaces XII*, pages 209–232, 2007.

[65] M. Jin, F. Luo, and X. Gu. Computing surface hyperbolic structure and real projective structure. *Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pages 105–116, 2006.

[66] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[67] Open University Milton Keynes. *Understanding space and time*. Open University Press, 1979.

[68] G.A. Klanderman, D.P. Huttenlocher, and W.J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. on Pattern Anal. Mach. Intell*, pages 850–863, 1993.

[69] R.I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the ICML*, pages 315–322, 2002.

[70] S. Lafon and A.B. Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1393–1403, 2006.

[71] J. Laub. *Non-metric pairwise proximity data*. PhD thesis, Berlin Institute of Technology, 10, Dec 2004.

[72] J. Laub, V. Roth, J.M. Buhmann, and K.R. Müller. On the information and representation of Non-Euclidean pairwise data. *Pattern Recognition*, 39(10):1815–1826, 2006.

[73] J.M. Lee. *Riemannian Manifolds: An Introduction to Curvature: v. 176*. Springer, 1997.

[74] H.T. Lin and C.J. Lin. A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods. Technical report, National Taiwan University, 2003.

[75] Harold Lindman and Terry Caelli. Constant curvature Riemannian scaling. *Journal of Mathematical Psychology*, 17:89–109, 1978.

[76] Fiedler M. Laplacian of graphs and algebraic connectivity. *Combinatorics and Graph Theory*, 25:57–70, 1989.

[77] D.J.C. MacKay. Gaussian processes-a replacement for supervised Neural Networks? *NIPS tutorial*, 1997.

[78] S. Mika, B. Schölkopf, A.J. Smola, K.R. Müller, M. Scholz, and G. Rätsch. Kernel pca and de-noising in feature spaces. *Advances in neural information processing systems*, 11(1):536–542, 1999.

[79] T. Mitchell. *Machine learning*. McGraw-Hill, 1997.

[80] V. Mottl, S. Dvoenko, O. Seredin, C. Kulikowski, and I. Muchnik. Featureless pattern recognition in an imaginary hilbert space and its application to protein fold classification. *Machine Learning and Data Mining in Pattern Recognition*, pages 322–336, 2001.

[81] L. Ornstein. Computer learning and the scientific method: A proposed solution to the information theoretical problem of meaning. *Journal of the Mount Sinai Hospitial*, 32(4):437–494, 1965.

[82] SIMBAD partners. The technical annex of beyond features: similarity-based pattern reanalysis and recognition. Technical report, University of York, UK, 2008.

[83] E. Pekalska and R.P.W. Duin. *The Dissimilarity Representation for Pattern Recognition. Foundations and Applications*. World Scientific Publishing Co Pte Ltd, 2005.

[84] E. Pekalska and R.P.W. Duin. Beyond traditional kernels: classification in two dissimilarity-based representation spaces. *IEEE Transactions on Systems, Man and Cybernetics–Part C*, 38(6), November 2008.

[85] E. Pekalska, R.P.W. Duin, S. Gunter, and H. Bunke. On not making dissimilarities Euclidean. *Lecture notes in computer science*, pages 1145–1154, 2004.

[86] E. Pekalska, A. Harol, R. Duin, B. Spillmann, and H. Bunke. Non-Euclidean or non-metric measures can be informative. *Structural, Syntactic, and Statistical Pattern Recognition*, pages 871–880, 2006.

[87] E. Pekalska, P. Paclik, and R.P.W. Duin. A generalized kernel approach to dissimilarity-based classification. *J. Mach. Learn. Res.*, 2:175–211, 2001.

[88] G. Perelman, G. Perelman, and G. Perelman. Finite extinction time for the solutions to the ricci flow on certain three-manifolds. *PLoS Biology*, 4(5):e8, 2006.

[89] J.C. Platt. Using analytic QP and sparseness to speed training of Support Vector Machines. *Advances in neural information processing systems*, pages 557–563, 1999.

[90] D. Poole. *Linear algebra: a modern introduction*. Cengage Learning, 2005.

[91] A.K. Qin and P.N. Suganthan. Kernel neural gas algorithms with application to cluster analysis. *Proceedings of the 17th International Conference on Pattern Recognition (ICPR), 2004*, 4:617–620, 2004.

[92] A. Robles-Kelly and E.R. Hancock. A Riemannian approach to graph embedding. *Pattern Recognition*, 40(3):1042–1056, 2007.

[93] V. Roth, J. Laub, J.M. Buhmann, and K.R. Müller. Going metric: Denoising pairwise data. *Advances in Neural Information Processing Systems*, 15:817–824, 2002.

[94] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[95] W.J. Rucklidge. Locating objects using the Hausdorff distance. *International Conference on Computer Vision*, pages 457–464, 1995.

[96] A. Sanfeliu and K. S. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(3):353–362, 1983.

[97] R. Sarkar, X. Yin, J. Gao, F. Luo, and X.D. Gu. Greedy routing with guaranteed delivery using ricci flows. *International Conference on Information Processing in Sensor Networks*, pages 121–132, 2009.

[98] E. Saucan, E. Appleboim, E.G. Wolansky, and Y.Y. Zeevi. Combinatorial ricci curvature and Laplacians for image processing. *CISR*, 2009.

[99] L.K. Saul, K.Q. Weinberger, J.H. Ham, F. Sha, and D.D. Lee. Spectral methods for dimensionality reduction. *Semisupervised learning*, pages 293–308, 2006.

[100] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[101] B. Schölkopf, A. Smola, and K.R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[102] L. G. Shapiro and R. M. Haralick. A metric for comparing relational descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(1):90–94, 1985.

[103] Y. Shavitt and T. Tankel. Hyperbolic embedding of internet graph for distance estimation and overlay construction. *Networking, IEEE/ACM Transactions on*, 16(1):25–36, 2008.

[104] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

[105] B.W. Silverman. Choosing the window width when estimating a density. *Biometrika*, 65:1–11, 1978.

[106] D.G. Sim, O.K. Kwon, and R.H. Park. Object matching algorithms using robust Hausdorff distance measures. *IEEE Transactions on Image Processing*, 8(3):425–429, 1999.

[107] M. Steyvers. Multidimensional scaling. *Encyclopedia of cognitive science*, 2002.

[108] A. Strehl and J. Ghosh. Relationship-based clustering and visualization for high-dimensional data mining. *INFORMS Journal on Computing*, 15(2):208–230, 2003.

[109] T. Tao. Ricci flow. Technical report, Department of Mathematics, UCLA, 2008.

[110] J.B. Tenenbaum, V. Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[111] W.S. Torgerson. Multidimensional scaling: theory and method. *Psychometrika*, 17(4):401–419, 1952.

[112] W.S. Torgerson. Theory and methods of scaling. 1958.

[113] A. Torsello, D. Hidovic, and M. Pelillo. A polynomial-time metric for attributed trees. *ECCV*, pages 414–427, 2004.

[114] A. Tversky. Features of similarity. *Psychological Review*, 84(4):327, 1977.

[115] LJP Van der Maaten, EO Postma, and HJ Van Den Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 2009.

[116] Deepak Verma and Marina Meila. A Comparison of Spectral Clustering Algorithms. Technical report, University of Washington, 2003.

[117] U. von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[118] U. von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, pages 555–586, 2008.

[119] W. Wallis. Graph distances using graph union. *Pattern Recognition Letters*, 22(6-7):701–704, May 2001.

[120] K.Q. Weinberger and L.K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70(1):77–90, 2006.

[121] E. W. Weisstein. From mathworld—a wolfram web resource.

[122] R. C. Wilson and E. R. Hancock. Spherical embedding and classification. *Structural, Syntactic, and Statistical Pattern Recognition*, pages 589–599, 2010.

[123] R. C. Wilson, E. R. Hancock, E. Pekalska, and R.P.W. Duin. Spherical embeddings for Non-Euclidean dissimilarities. *Computer Vision and Pattern Recognition*, pages 1903–1910, 2010.

[124] World Scientific Publishing Company. *Riemannian manifolds of positive curvature*, 2011.

[125] G. Wu, E.Y. Chang, and Z. Zhang. An analysis of transformation on non-positive semidefinite similarity matrix for kernel machines. Technical report, University of California, Santa Barbara, 2005.

[126] B. Xiao and E. R. Hancock. Geometric characterisation of graphs. *ICIAP*, LNCS 3617:471–478, 2005.

[127] B. Xiao, E.R. Hancock, and R.C. Wilson. Geometric characterization and clustering of graphs using heat kernel embeddings. *Image and Vision Computing*, 28(6):1003–1021, 2010.

[128] Wilson R. C. & Hancock E. R. Xiao, B. Characterising graphs using the heat kernel. *BMVC*, 2005.

[129] R. Xu, D. Wunsch, et al. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

[130] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273, 2003.

[131] S.X. Yu and J. Shi. Multiclass spectral clustering. *Ninth IEEE International Conference on Computer Vision*, pages 313–319, 2003.

[132] W. Zeng, D. Samaras, and D. Gu. Ricci flow for 3d shape analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):662–677, 2010.

[133] W. Zeng, X. Yin, Y. Zeng, Y. Lai, X. Gu, and D. Samaras. 3d face matching and registration based on hyperbolic ricci flow. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–8, 2008.

[134] H. Zhang, A.C. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:2126–2136, 2006.

[135] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *Journal of Shanghai University (English Edition)*, 8(4):406–424, 2004.