The University of Newcastle, Australia

Doctoral Thesis

Topological Analysis, Non-linear Dimensionality Reduction and Optimisation Applied to Manifolds Represented by Point Clouds

---

Author:
Rahul Paul
The University of Newcastle

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy in Computer Science

in the

Interdisciplinary Machine Learning Research Group

School of Electrical Engineering and Computing

The University of Newcastle, Australia

This research was supported by an Australian Government Research Training Program (RTP) Scholarship

Submitted June 2018

Declaration of Authorship

Statement of Originality

I, Rahul Paul, hereby certify that the work embodied in the thesis titled, 'Topological Analysis, Non-linear Dimensionality Reduction and Optimisation Applied to Manifolds Represented by Point Clouds' is my own work, conducted under normal supervision.

The thesis contains no material which has been accepted, or is being examined, for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I give consent to the final version of my thesis being made available worldwide when deposited in the University's Digital Repository, subject to the provisions of the Copyright Act 1968 and any approved embargo.

Acknowledgements

First of all, I would like to express my deep gratitude to my supervisor, A/Prof. Stephan K Chalup, for being supportive and motivating throughout this research. His serious attitude towards scientific research greatly influences me in research and life. Most of the work in this thesis is attributed to lively meetings and discussions with Stephan.

I am greatly indebted to A/Prof. Yuqing Lin for his valuable suggestions and kind support towards my research. I would also like to thank my previous supervisor, G.N.S Prasanna, who helped me build up the basic framework of this research during my Masters.

I am grateful to A/Prof. Thomas Fiedler for providing us with the high resolution image data set.

I am also grateful to Aaron Scott for support with the UON HPC system.

I would like to acknowledge the financial support provided by the University of Newcastle with the provision of an UNRSC50:50 PhD scholarship.

Last, but definitely not least, I offer my special thanks to my wife, Satarupa Das, for all her support, encouragement and understanding.

Contents

List of Figures

List of Tables

THE UNIVERSITY OF NEWCASTLE, AUSTRALIA
Faculty of Engineering and Built Environment
School of Electrical Engineering and Computing

Abstract of PhD Thesis

Topological Analysis, Non-linear Dimensionality Reduction and Optimisation Applied to Manifolds Represented by Point Clouds

by Rahul Paul

In recent years, there has been a growing demand for computational techniques that respect the non-linear structure of high-dimensional data, in both real-world applications and research. Various forms of manifolds can describe non-linear objects. However, manifolds are abstract mathematical concepts and in applications these are often represented by high-dimensional finite sets of sample points. This thesis investigates techniques from machine learning, optimisation and computational topology that can be applied to such point clouds. The first part of this thesis presents a topological approach for validating non-linear dimensionality reduction. During the process of non-linear dimensionality reduction, manifolds represented by point clouds are at risk of changing their topology. The impact of manifold learning is evaluated by comparing Betti numbers based on persistent homology of test manifolds before and after dimensionality reduction. The second part of the thesis addresses the processing of large point cloud data as it can occur in real applications. The topological analysis of this data using traditional methods for persistent homology can be a computationally costly task. If the data is represented by large point clouds, many current computing systems find processing difficult or fail to process the data. This thesis proposes an alternative approach that employs deep learning to estimate Betti numbers of manifolds represented by point clouds. The third part of the thesis investigates simulated examples of optimisation on general differentiable manifolds without the requirement of a Riemannian structure. A barrier method with exact line search for the optimisation problem over manifolds is proposed. The last part of this thesis reports on collaborative field work with Xerox India using a real-world data set. A heuristic algorithm is employed to solve a practical task allocation problem.

Chapter 1

Introduction

In computer science and machine learning research, the investigation of the qualitative properties of a data set are a crucial part of data analysis (Kaski and Peltonen, 2011). As the dimension increases, the ability to analyse the topological properties of a given data set becomes even more important. This problem can become computationally expensive for large random point cloud data sets sampled from a probability distribution. Very often data is generated as an unordered sequence of points in Euclidean space (Ludu, 2016). This data can be in the form of time series generated from sensors, point cloud data from 3D scans and motion capture data. The unordered sequence of data can reside in a very high dimensional space. The global topology of the essential information or structure behind the data can give important information about the underlying event which generated the data. This type of point cloud data can be sampled from any 2-, 3-, or higher-dimensional object.

Parts of this research focused on extracting global topological information about the structure of objects represented by data sets of discrete sample points. Point cloud data can appear in both low and high dimensions, and in both cases, can represent the same underlying object. The topology type can be calculated for both representations, and should then be identical. Maybe it is not crucial in dimensionality reduction to preserve the topology but more importantly to preserve the information so that classifiers or other

machine learning techniques do very well. However, this thesis focuses on aspects of non-linear dimensionality reduction that either preserve or change the topology of the manifold underlying the data. One difficulty that can occur in this context lies in distinguishing between the topological features of the actual point cloud data and the topological features of the underlying manifold. Some of this difficulty is related to the random nature of point clouds.

One way to control or analyse high-dimensional data is to apply dimensionality reduction. In this thesis, we focused on manifold learning, that is, non-linear dimensionality reduction (Lee and Verleysen, 2007). In the potential applications of manifold learning, the topology and dimensionality of data can vary significantly.

For some applications, such as clustering, optimisation and embeddings of low-dimensional manifolds, the calculation time to reduce dimensionality and to determine the topology can be significant, and depends on the number of sample points. This provides motivation for the exploration of accurate and more rapid methods for validating manifold learning techniques with the help of various topology tools. We review techniques for quality assessment of manifold learning and propose the use of persistent homology to evaluate the topological impact of manifold learning by comparing the Betti numbers of test manifolds before and after dimensionality reduction. A non-linear projection of data into a lower-dimensional space is regarded to be of good quality if the connections (e.g., edges and faces) between neighbouring data points after the projection reflect the same topological relationships as before the projection (Gashler et al., 2011).

Persistent homology (PH) is an algebraic method for measuring topological features of various shapes and functions. It has many applications, for example, in the fields of computer vision and image analysis. PH is a response to the challenge that one encounters when trying to assign topological invariants to manifolds that are represented by point clouds. Data in high-dimensional spaces is ubiquitous across a variety of domains. A major problem in data analysis is how to create effective schemes to reduce the dimensionality while maintaining or improving the ability to make geometric and statistical inferences (Carlsson et al., 2009).

Motivated by the importance of topology in machine learning and data analysis, an important question is how can we validate the preservation of topological features of data during manifold learning and optimisation. In this work, topological methods are developed to address validation of non-linear dimensionality reduction. These studies include improvements over current algorithms that are demonstrated by simulation experiments. It is essential for data analysis to investigate the number of connected components as well as any holes present in the data. In 3-dimensional data, as well as in data of higher dimensions, there are more topological properties, such as tunnels, pretzels and voids and their higher dimensional analogous, that need to be considered. PH uses hole counts in various dimensions and gives a classification of holes in terms of Betti numbers. Betti numbers can therefore be used to determine the similarity of manifold data with respect to the topology.

Computational topology algorithms and manifold learning techniques are all very costly algorithms. For most of these methods, data must be loaded into the physical memory to compute simplices which determine the topology of the data. This requires efficient updates and computation of huge simplicial complexes that have a persistent topology type, even when the distribution of the data and parameters might change slightly based on sample selection variations. It can become computationally infeasible to calculate the topology of point cloud data if the number of sample points increases in the data. However, due to the complex structure of some manifolds and real-world data, substantial numbers of data points are necessary to understand the topology underlying the manifold. These considerations motivated the deep learning-based solutions to topology estimation investigated in Chapter 4.

Another aspect that can be investigated in the context of point cloud data are the local minima and maxima of functions on a dataset. In this context the thesis explores solutions to optimisation on manifolds. The thesis reports on example experiments of a proposed solution for a problem that the standard solver in Matlab was not able to process successfully. We also compared our results in terms of time complexity where my experiments indicate that it was linear if the number of sample points of the manifold was increased. Optimisation on manifolds is another form of modern non-linear techniques associated with various data. Riemannian optimisation generalises tools from continuous,

unconstrained optimisation, such as gradient descent, Newton methods and trust-region methods. In transitioning from the classical Euclidean case to Riemannian search spaces, some features are lost such as certain convergence guarantees for these processes. Under substantially the same conditions, global and local convergence results can be established as shown in the theory laid out by Absil et al. (2010).

It is always useful for the community to apply the outcomes of research to real-world industrial data. Often industry data is simple and traditional linear techniques are appropriate but sometimes it can be complicated, non-linear and very high dimensional. To deal with this data appropriately can also be more computationally costly than expected. This observation motivated the inclusion of an industrial research component in Chapter 5 that used several meta-heuristic techniques on a given real-world task.

Research Objectives

1. When manifold learning is applied to point cloud data, difficulties can arise because the point clouds are generated using uniform random numbers [1]. The underlying manifolds can be very complex and non-linear dimensionality reduction can cause geometrical distortions that are so substantial that the original neighbourhood structure of the points on the manifolds is modified. This is referred to as topological distortion (Aupetit, 2007). Therefore, the first objective of this research was to validate the process of non-linear dimensionality reduction technique when applied to point cloud data that represents manifolds of different topological complexity.

2. Traditional techniques to calculate PH can become too computationally costly for large point cloud data. The computational complexity of these techniques can become exponential in space and time (Otter et al., 2017). Therefore, the second research objective was to identify faster and computationally less expensive alternatives to the traditional algorithms for PH. Our hypothesis was that deep neural networks could recognise an object's topology from sample data.

[1] The random number could be generated according to Uniform, Gaussian or other distribution. A detailed analysis and discussion of the impact of the different sampling mechanisms were beyond the scope of this thesis.

3. Optimisation on Riemannian manifolds is a well-established research area with many applications. However, it has been established that these methods are excessively slow in high dimensions (Absil et al., 2010). Our third objective was to explore the possibility of a faster approach by applying unconstrained optimisation to general differentiable manifolds.

1.1 Background

1.1.1 Topological Space

Topology is a branch of mathematics that is related to analysis, geometry and algebra. A metric space is a set together with a distance function defined on pairs of points, satisfying certain conditions.

In school geometry, one begins with the study of maps that preserve congruence, followed by those preserving similarity. This involves the preservation of nearness of points to sets (Naimpally and Peters, 2013). The most common way to measure distances is a metric in a metric space.

Definition 1.1. If $\mathcal{X}$ is a set, **metric** on $\mathcal{X}$ is a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ satisfying

(i) $d(x, y) \geq 0$ for all $x, y \in \mathcal{X}$ with equality if and only if $x = y$.

(ii) $d(x, y) = d(y, x)$ for all $x, y \in \mathcal{X}$

(iii) $d(x, z) \leq d(x, y) + d(y, z)$ for all $x, y, z \in \mathcal{X}$

A metric space is a set $\mathcal{X}$ together with a specific choice of metric d on $\mathcal{X}$. Sometimes we will say "Let $(\mathcal{X}, d)$ be a metric space". If $\mathcal{X}$ is a metric space with metric d, the elements of $\mathcal{X}$ are usually called its points, because we think of $\mathcal{X}$ as a "space" having a certain "shape", rather than just as a set. The number $d(x, y)$ is called the distance from x to y.

More general, a topological space is a set together with a structure leading to the study of continuous functions from one topological space to another.

Definition 1.2. Let M be a set and $\mathcal{P}(M)$ be the power set of M, i.e., the set of all subsets of M (Lee and Verleysen, 2007).
A set $\mathcal{O} \subseteq \mathcal{P}(M)$ is called a **topology**, if it satisfies the following:

(i) $\emptyset \in \mathcal{O}$, $M \in \mathcal{O}$

(ii) $U \in \mathcal{O}, \quad V \in \mathcal{O} \implies U \cap V \in \mathcal{O}$

(iii) $U_\alpha \in \mathcal{O}, \quad \alpha \in \mathcal{A}$ ($\mathcal{A}$ is an index set $\implies \left(\bigcup_{\alpha\in\mathcal{A}} U_\alpha\right) \in \mathcal{O}$

Terminology:

1. the tuple $(M, \mathcal{O})$ is a **topological space**.
2. $\mathcal{U} \in M$ is an **open set** if $\mathcal{U} \in \mathcal{O}$.
3. $\mathcal{U} \in M$ is a **closed set** if $M \setminus \mathcal{U} \in \mathcal{O}$.

1.1.2 Topological Manifolds

Manifolds are topological spaces that look locally like Euclidean space. A little more precisely it is a space together with a way of identifying it locally with a Euclidean space that is compatible on overlap.

Some topological spaces can be charted analogously to how the earth is charted in an atlas. This particular type of topological space is called a topological manifold. The basic definition of a topological manifold was developed more than 100 years ago (C.F. Gauss 1777-1855) in mathematics and is now central to geometry and topology. It generalises the concepts of curves and surfaces and addresses these objects without referring to any neighbouring ambient space. There are several different ways to introduce the fundamentals of manifold theory. Many of them are designed as a basis for further general and formal mathematical studies, for example, in geometric topology or differential geometry. The following statements aim to give a concise and geometrically intuitive overview of the basic definitions and terminology of manifold theory, as they are essential for practical applications in computer science, computer graphics and machine learning.

A topological space $(M, \mathcal{O})$ is called a **d-dimensional topological manifold** if for all points $p \in M$, there is a open set $\mathcal{U}$ around the point p in respect to the topology $\mathcal{O}$, such that there is a homeomorphism x onto $\mathbb{R}^d$. This can be expressed as below:

$$\forall p \in M : \exists \mathcal{U} \in \mathcal{O} : p \in \mathcal{U}, \exists x : \mathcal{U} \to x(\mathcal{U}) \subseteq \mathbb{R}^d \tag{1.1}$$

satisfying the following:

(i) x **is invertible**: $x^{-1} : x(\mathcal{U}) \to \mathcal{U}$

(ii) x **is continuous** w.r.t. $(M, \mathcal{O})$ and $(\mathbb{R}^d, \mathcal{O}_{std})$

(iii) x^{-1} **is continuous**

Further, the standard demand is that M is a Hausdorff space and its topology has a countable basis. Not all topologies are Hausdorff. Non-Hausdorff spaces can display unusual and counterintuitive behaviour. From the perspective of mathematical iterative algorithms, the most troubling event is that a converging sequence on a non-Hausdorff topological space may have many discrete limit points (Absil et al., 2009). Our definition of manifolds rules out non-Hausdorff topologies (Lee, 2010):

Definition: a **d-dimensional topological manifold** is a second countable Hausdorff space that is locally Euclidean of dimension d.

Definition: A topological space $\mathcal{X}$ is Hausdorff if for any $x, y \in \mathcal{X}$ with $x \neq y$ there exist open sets $\mathcal{U}$ containing x and $\mathcal{V}$ containing y such that $\mathcal{U} \cap \mathcal{V} = \emptyset$ (Lee, 2010).

Since the only manifolds considered in this thesis are topological manifolds, they are simply referred to as **d-dimensional manifolds**, or even just **manifolds**. The most obvious example of an n-manifold is $\mathbb{R}^n$ itself.

Fig.1.1 shows a torus topological manifold.

Sometimes it is desirable to judge the suitable conditions not on the object itself but on a chart representation of that real world object. Sometimes this is the only way to define properties like continuity of real world objects or a curve in the real world. In our experiment, we will be using the Swiss roll and heated Swiss roll (Lee and Verleysen, 2007) manifolds for all our comparisons and topology investigations.

1.1.3 Manifold Models

According to Whitney (1936), any d-manifold can be embedded in R^{2d+1} , meaning that $2d+1$ dimensions at most are necessary to embed a d-manifold. Dimensionality reduction of manifolds involves the learning of the manifold structure. The charts and the atlas,

which can make the manifolds, as well as the map, define the topological space for a given manifold. For example, a sphere is a (compact) 2-manifold that can be embedded in R^3 but not in R^2.

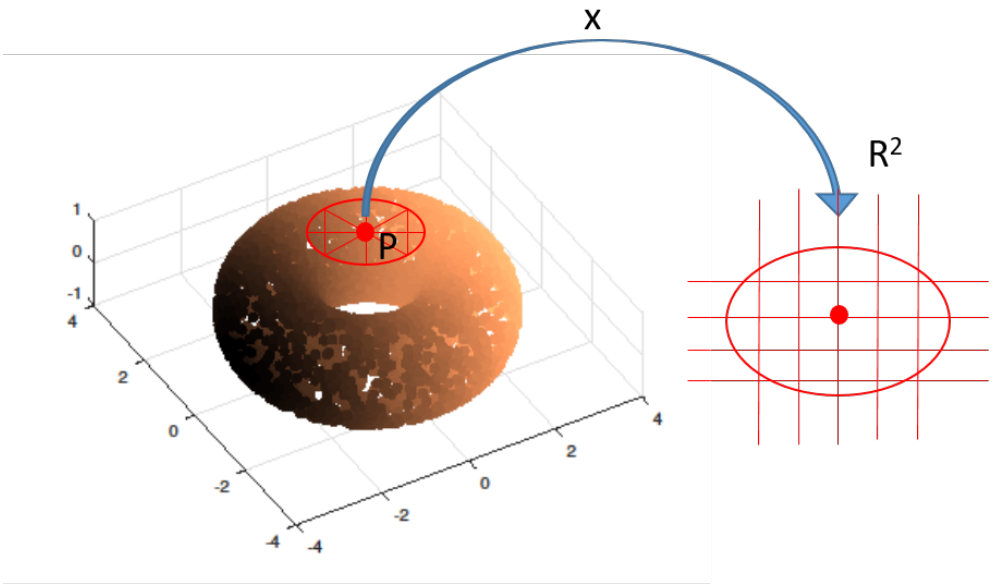


FIGURE 1.1: Topological manifold

1.1.4 Differential Manifold

A differentiable structure for a manifold M is an atlas $(\mathcal{U}_i, x_i)_{i\in J}$ of M which is maximal and possess the property that, for all regions of intersection $\mathcal{U}_i \cap \mathcal{U}_j \neq \emptyset$, $i, j \in J$ all chart transitions

$$x_j x_i^{-1}, x_i x_j^{-1} : \mathbb{R}^d \to \mathbb{R}^d \tag{1.2}$$

are differentiable. **A differentiable manifold** of dimension $d \geq 1$ is a d-dimensional topological manifold M with a differentiable structure. For each point p of a d-dimensional differentiable manifold M, there is a tangent space T_pM which has the same dimension as the manifold's model space $\mathbb{R}^d$. Intuitively, T_pM can be interpreted as a linear approximation of M in p. There are many different ways to formally define the tangent space. Since we may assume that M is embedded in some $\mathbb{R}_m$, the tangent space at point p can be defined as the subspace spanned by the tangent vectors of all smooth curves in M through point p.

For example, The euclidean spaces $\mathbb{R}^n$ and the n-spheres $S^n = \{x \in \mathbb{R}^n; \|x\| = 1\}$ are n-dimensional manifolds.

1.1.5 Riemannian Manifolds

Topological and differentiable manifolds are objects of research in geometric and differential topology. To study the properties of manifolds with differentiable deformations (i.e., homeomorphisms and diffeomorphisms, respectively), metric structures on differentiable manifolds must be taken into account. By defining a dot product intrinsic to each manifold, the lengths of tangent vectors and the angles between them can be measured. This allows measurement of the lengths of curves on the manifolds and allows study of geometric properties such as curvature.

A Riemannian metric for a differentiable manifold M is a dot product for each tangent space T_pM which depends smoothly on the base point $p \in M$. **A Riemannian manifold** is a differentiable manifold with a Riemannian metric. Every differentiable manifold can be equipped with a Riemannian metric. The dot product of a Riemannian manifold is represented by a positive definite, symmetric matrix where the coefficients depend smoothly on the base point.

1.1.6 Persistent Homology

PH can be employed to analyse data sets of points that are sampled from such manifolds. PH (Kaczynski et al., 2004) is an algebraic tool for measuring the topological properties of a given data set. In particular, in the current experiment we intended to use persistent Betti numbers. We recall some definitions.

PH provides us with tools for counting the holes and other topological features of manifolds that are represented by point cloud data. It helps us to determine how many sample points are required to describe a manifold and its topological features.

Simplicial complexes connect the data points using an n-dimensional triangular structure from the original data. An abstract simplicial complex can be characterised by:

1. A set Z of vertices or 0-simplices.

2. For each $k \geq 1$, a set of k-simplices $\sigma = [z_0, z_1, ..., z_k]$ where $z_i \in Z$.

3. Each k-simplex has $k + 1$ faces obtained by deleting one of the vertices.

The following membership property must be satisfied: if σ is in the simplicial complex, then all faces of σ must be in the simplicial complex. We think of 0-simplices as vertices, 1-simplices as edges, 2-simplices as triangular faces, and 3-simplices as tetrahedrons (Tausz et al., 2014a).

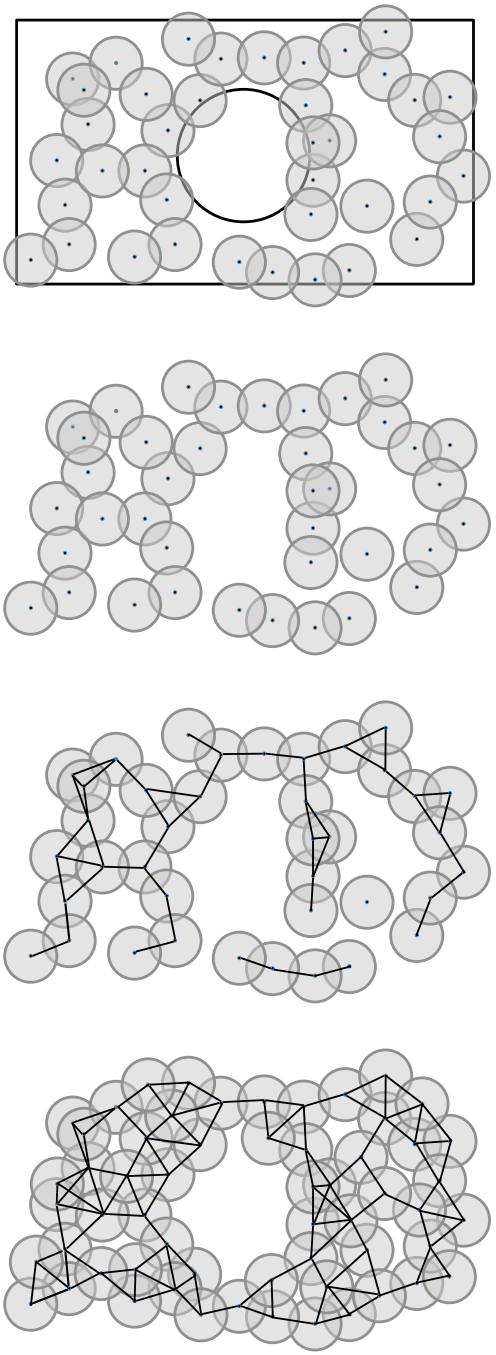


FIGURE 1.2: (a) Points sampled from the Japanese flag manifold; (b) each point is surrounded by a ball of radius r; (c) points and associated simplicial complex; (d) when increasing the number of sample points, the number of connected components and holes can change and approximates the values characteristic of the underlying manifold.

Betti numbers describe the number of n-dimensional holes in the data space. B_k is the k-th Betti number and equates to the number of k-dimensional holes in the data set (Tausz

et al., 2014a). Betti numbers can also tell us broadly about the topology of an examined space or object. Suppose we sample random points from a given object. The corresponding Betti numbers are a vector of random variables B_k.

Algebraic topology provides tools to describe the topological structure of objects such as n-dimensional manifolds (Dold, 2012). PH is a concept in computational topology that was designed for real applications where topological objects such as manifolds are represented or approximated by sets of sample points (Carlsson et al., 2009; Edelsbrunner et al., 2000; Ghrist, 2008; Tausz et al., 2014b).

The i-th Betti number of a topological space $M \subset \mathbb{R}^n$ is defined as $B_i(M) = \text{rank } (H_i(M))$, where $H_i(M)$ is the i-th singular homology group of M (Dold, 2012). Similarly, as the Euler characteristic χ can be used to count the holes in manifolds, the Betti numbers can be used for counting connected components, cavities, tunnels and related higher-dimensional topological features of manifolds. The Betti numbers give a more detailed description of a manifold's topological structure than the Euler characteristic alone. The relation between the Betti numbers and the Euler characteristic is

$$\chi = B_0 - B_1 + B_2 - B_3 + ... + B_n \tag{1.3}$$

where B_i is the i-th Betti number and equates to the number of i-dimensional holes in the manifold. An i-dimensional hole can be interpreted as an area or volume bounded by an $(i-1)$-dimensional sphere. For manifolds in $\mathbb{R}^3$, only the first three Betti numbers are relevant because no higher-dimensional holes can occur, i.e.

B_0 = number of path-connected components

B_1 = number of circular holes

B_2 = number of 3-dimensional cavities or bubbles

For example, for the surface of a torus T^2, the 2-dimensional sphere S^2, the Japanese flag manifold J^2, and the Swiss roll SR_i^2 with i = 0, 1, 3, and 7 holes, the first three Betti numbers and the Euler characteristic χ are as follows:

Manifold	B_0	B_1	B_2	χ
T^2	1	2	1	0
S^2	1	0	1	2
SR_0^2	1	0	0	1
J^2, SR_1^2	1	1	0	0
SR_3^2	1	3	0	-2
SR_7^2	1	7	0	-6

In order to associate a collection of points in $\mathbb{R}^n$ with a global object (e.g. a manifold M which is the source of the data), each point x of the point cloud is used as a vertex of a combinatorial graph whose edges are determined by neighbourhood balls $B_r(x) = \{p \in \mathbb{R}^n; d(x,p) \leq r\}$ of radius r (Ghrist, 2008).

An abstract simplicial complex S can be characterised by: 1) a set V of vertices or 0-simplices; 2) for each $k \geq 1$, a set of k-simplices $\sigma = [z_0, z_1, ..., z_k]$ where $z_i \in V$; and 3) each k-simplex has $k+1$ faces obtained by deleting one of the vertices. The following membership property must be satisfied: if σ is in S, then all faces of σ must be in S.

For 3-dimensional objects, 0-simplices are vertices, 1-simplices are edges, 2-simplices are triangular faces, and 3-simplices are tetrahedrons. A simplicial complex can be defined as a finite collection of simplices K such that, 1) if $\sigma \in K$ and τ is a face of σ then $\tau \in K$; and 2) if $\sigma, \sigma' \in K$ then $\sigma \cap \sigma'$ is either empty or a face of both (Edelsbrunner and Harer, 2010).

There are various ways to generate simplicial complexes from point clouds. A computationally efficient method is to generate a Vietoris-Rips complex, which is the simplicial complex whose k-simplices are determined by unordered $(k+1)$-tuples of points that are pairwise within a given distance $r > 0$ (Ghrist, 2008; Tausz et al., 2014b).

An important parameter when converting a set of points into a simplicial complex is the radius r of the neighbourhood balls $B_r(x)$. If r is sufficiently small, the complex becomes a discrete set; if r is too big the complex fuses into a sole high-dimensional simplex and information about essential topological features such as characteristic holes of the manifold, is lost. Fig. 1.2 shows an intuitive example of the Japanese flag manifold

that demonstrate how additional sample points and the suitable choice of r can remove small holes to distinguish them from the larger hole that is characteristic of the manifold's topology. It can be seen that under certain niceness conditions, a discrete approximation of a manifold can be achieved (Niyogi et al., 2008). However, in general, this approach depends not only on the structure of the manifold, but also on the sample distribution. Therefore, the process of 'fattening' points into balls can uncover essential topological features of the manifold at one end, while simultaneously covering up essential features at the other end of the manifold. Consequently, there may not be a suitable ball size for a given point cloud. The idea of PH is not to use one complex with a fixed ball size, but to use a filtration of complexes associated with a sequence of increasing ball sizes, and to observe when topological features of the manifold occur and when they disappear in the process (Ghrist, 2008). The features that persist longest in this process are then regraded as essential.

1.1.7 Dimensionality Reduction

Statistical and machine learning methods face a formidable problem when dealing with high dimensional data. Normally, the number of input variables are reduced before a data mining algorithm can be successfully implemented. Dimensionality reduction can be performed in two ways: either by only keeping the most appropriate variables from the original dataset, or by exploiting the redundancy of the input data and finding a smaller set of new variables. Both techniques result in a smaller set of data, each being a blend of input variables containing the same information as the original data.

The intrinsic dimensionality of a data vector can be defined as the minimal number of the parameters or latent variables required to describe the data vector. For more rigorous definition of intrinsic dimensionality, see Chapter 3 of the book by Lee and Verleysen (2007). Non-linear dimensionality reduction (DR) is also referred to as manifold learning. The task of manifold learning can be described as, given a data set of dimension D, determine its intrinsic dimensionality d, and extract a manifold of dimension d from the data, where $d < D$ or often $d \ll D$. The difficulty of manifold learning is to extract the intrinsic manifold without changing its topology and geometry.

In general, neither the topology, geometry or the intrinsic dimensionality of the dataset is known in advance. Fig. 1.3 shows the techniques available for DR as described by van der Maaten et al. (2009). Table 1.1 provides a list of the most used DR techniques in practice.

TABLE 1.1: Most used dimensionality reduction algorithms in the literature, listed chronologically

1901	Principal Component Analysis (PCA)
1969	Sammon Mapping (SM)
1997	Curvilinear Component Analysis (CCA)
1998	Kernel PCA (KPCA)
2000	Isomap
2000	Locally Linear Embedding (LLE)
2001	Linear Discriminant Analysis (LDA)
2001	Laplacian Eigenmaps (LE)
2004	Maximum Variance Unfolding (MVU)
2006	Diffusion Maps (DM)
2008	t-Stochastic Neighbor Embedding (t-SNE)
2010	Maximum Entropy Unfolding (MEU)

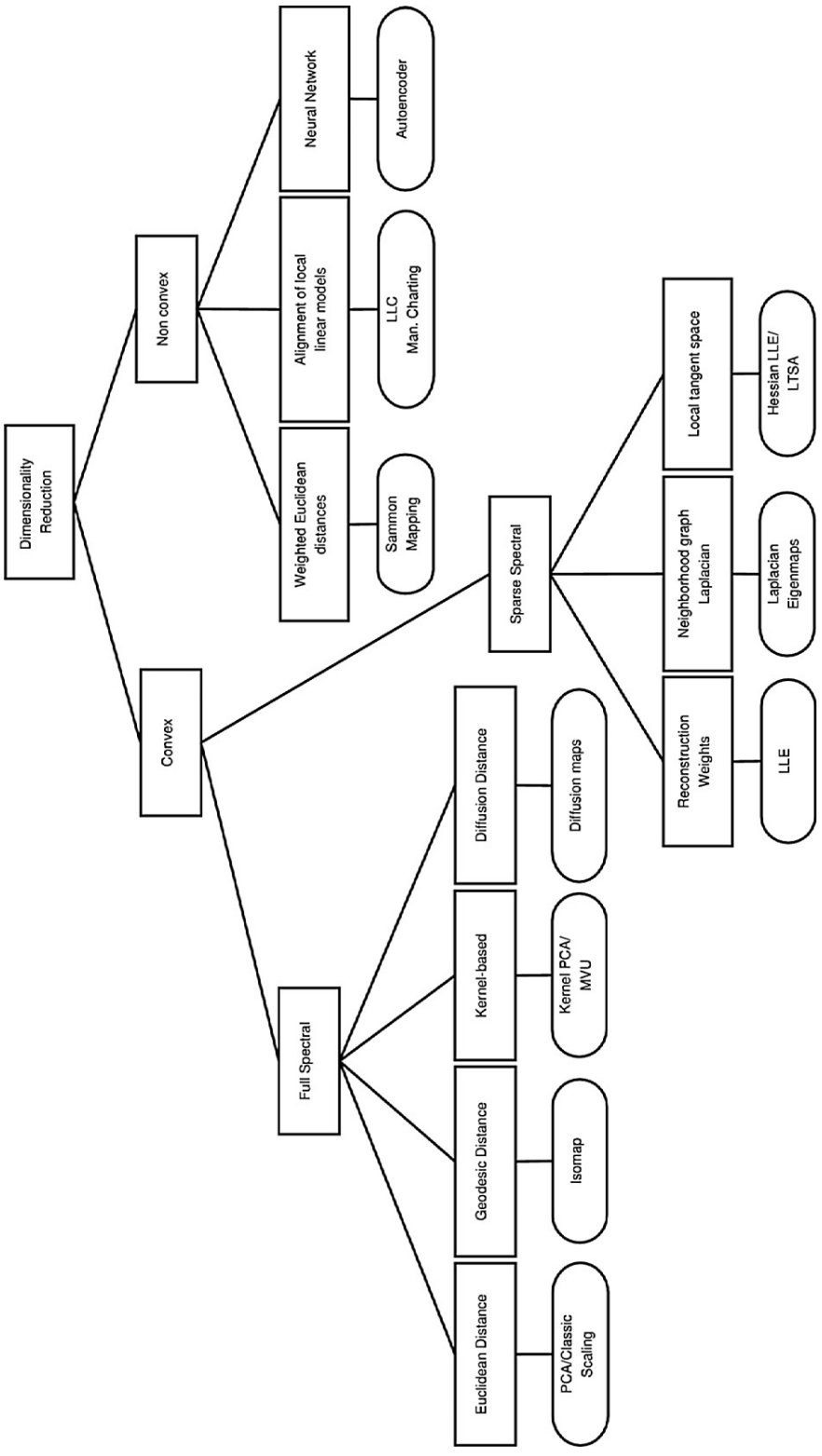


FIGURE 1.3: Taxonomy of dimensionality reduction (van der Maaten et al., 2009)

1.1.8 Manifold Learning

The essential geometrical and topological information of a set of points can lie on a non-linear, low-dimensional manifold that is embedded in a high-dimensional ambient vector space. For most data sets in real-world pattern recognition applications, the underlying manifold would initially be unknown. Through manifold learning, extraction of the underlying manifold can be attempted by mapping the data into a space of lower dimension (Lee and Verleysen, 2007; van der Maaten et al., 2009).

The minimum dimensionality of a manifold's ambient space depends on the topological properties of the manifold. In some cases, the ambient space could be of the same dimension, and in other cases, it may require a higher dimension than the manifold. For example, an n-dimensional sphere S^n can be embedded in $\mathbb{R}^{n+1}$ or any higher-dimensional Euclidean space. According to a theorem by Whitney (1936), every n-dimensional, smooth manifold can be embedded in a $2n$-dimensional ambient space. Variations of this result exist in special cases and for more general manifolds (Hirsch, 1976). In real-world tasks, for example, in computer vision when the data consists of digital images, the ambient space is often of a much larger dimension than the intrinsic dimensionality of the underlying manifold. In these cases, dimensionality reduction can help to reduce computational complexity without changing the nature of the data, and can lead to a better understanding of the data.

Traditional approaches to DR, such as Principal Component Analysis (PCA) (Jolliffe, 2002) and Multi-Dimensional Scaling (MDS) (Cox and Cox, 2000), can recover the true dimensionality of intrinsic manifolds from high-dimensional data, but only when the relationship between the high-dimensional representation and the low-dimensional latent variables is approximately linear (Mardia et al., 1979).

Manifold learning is synonymous with non-linear DR (Lee and Verleysen, 2007). Manifold learning typically tries to find a non-linear mapping from a high-dimensional space to a lower-dimensional space, that preserves the topological or local geometrical structure of the manifold underlying the data (Baraniuk and Wakin, 2009). Example methods include Isomap (Tenenbaum et al., 2000), Locally Linear Embedding (LLE) (Roweis and Saul, 2000a) and Maximum Entropy Unfolding (MEU) (Lawrence, 2012). LLE tries to

preserve the local linear neighbourhood structure of the points taken from the point cloud data. Isomap attempts to learn an isometric embedding of manifolds while trying to preserve the geodesic distances between points on the manifold. MEU uses the non-linear generalisation of PCA. It uses spectral dimensionality reduction which views these methods as Gaussian Markov random fields. It also uses the maximum entropy principle to get the maximum variance unfolding. All these methods used for dimensionality reduction. As we consider only k-nearest neighbour methods for dimensionality reduction in this thesis, other methods can be explored as one of the future work.

Geodesic distances are approximated by distances in k-nearest neighbour graphs and by avoiding any measurements that would exit the manifold and significantly short cut through the ambient space. A suitable k has to be determined upfront for each combination of manifold and DR method. A goal of these methods is to embed the manifold data authentically into a lower-dimensional space where it is easier to analyse. Each of the methods can cause distortions, folds, rips at the edges, or the emergence of additional holes and cavities during the DR process (Akkucuk and Carroll, 2006; Balasubramanian and Schwartz, 2002a; Li et al., 2005; Saul et al., 2006).

1.1.9 Principal Component Analysis (PCA)

PCA (Hotelling, 1933) is a linear technique for DR. PCA is performed by embedding the high dimensional data into a linear subspace of lower dimension. Although there are many linear DR techniques available, PCA is by far the most popular and effective linear DR technique. Therefore, in this thesis only PCA is included. PCA creates the low-dimensional representation of the data with the help of variance from the data. It finds the linear basis of reduced dimensionality for the data where the variance is maximum. Suppose we have a high-dimensional dataset in the form of a matrix. In some cases, the dimensions can be thought of as directions that the information varies along. If we want to reduce the dimensionality of the information, then what we want to do is to find an approximate 'basis' to the set of direction. That is, we want to find only the crucial dimensions that serve most of the information of the other dimensions. The idea here is that the dimensions we discard are in some sense duplicates of the dimensions we keep

because they can be reconstructed from the small set of linear basis. PCA calculates, as the input into most other multidimensional scaling techniques, a pairwise Euclidean distance matrix D whose entries d_{ij} represent the Euclidean distance between the high-dimensional data points x_i and x_j. Classical scaling finds the linear mapping M that minimises the cost function

$$\sum_{i,j}(d_{ij}^2 - \|y_i - y_j\|^2) \tag{1.4}$$

in which $\|y_i - y_j\|^2$ is the Euclidean distance between the low dimensional datapoints y_i and y_j which are represented from high-dimensional points x_i and x_j.

PCA may also be used as probabilistic PCA (Roweis, 1998) which uses a Gaussian prior over the latent space, and a linear-Gaussian noise model. The probabilistic formulation of PCA leads to an Expectation-Maximization algorithm which is computationally more efficient for very high-dimensional data. By using Gaussian processes, probabilistic PCA may also be extended to learn nonlinear mappings between the high-dimensional and the low-dimensional space (Lawrence, 2005). Another extension of PCA uses the eigenvectors corresponding to the most significant eigenvalues in the linear mapping (as principal components) relevance in embedding.

1.1.10 Isomap

PCA has been shown to be successful in many applications, but it is limited by the fact that it primarily aims to preserve pairwise Euclidean distances, and does not consider the distribution of the neighbouring data points or neighbourhood. If the high-dimensional data lies on or near a curved manifold, such as in the Swiss roll dataset which is explained in Chapter 2, PCA may consider two data points to be very near, whereas their distance over the manifold or geodesical distance is much larger than the typical Euclidean distance. Isomap (Balasubramanian and Schwartz, 2002a) is a technique that resolves this problem by attempting to preserve pairwise geodesic distances between data points. Geodesic distance is calculated by estimating the distance over a curved surface.

Specifically, geodesic distance is calculated by generating the neighbourhood graph. In a neighbourhood graph, every data point is connected with its k nearest neighbours from

the data set. The shortest path between two points from the dataset in the graph forms an estimate of the geodesic distance between these two points, as determined using curve fitting. This has to be repeated for all the points in the data set. Therefore, we obtain the pairwise geodesic distance matrix for all points. In the low-dimensional representation, preservation of the pairwise distance is attempted.

An important drawback of the Isomap algorithm is that it makes the object topologically unstable after projection (Balasubramanian and Schwartz, 2002b). Isomap may construct erroneous connections in the neighbourhood graph. These errors can make the object topologically different from the original object. Another weakness of Isomap is that it cannot perform well if holes present in the manifold. This issue can be managed by tearing manifolds with holes (Li et al., 2005) or by increasing the sample points per cycle. Isomap can easily deal with 'open' structures, but fails to recover an object in low dimensions when the data structure is 'closed', such as when there are regularly arranged points on a sphere (Akkucuk and Carroll, 2006). Fig. 1.4 shows successful embedding of a Swiss Roll (SR) Fig. 2.1a and Heated Roll (HR) Fig. 2.1e into the 2-dimensional plane.

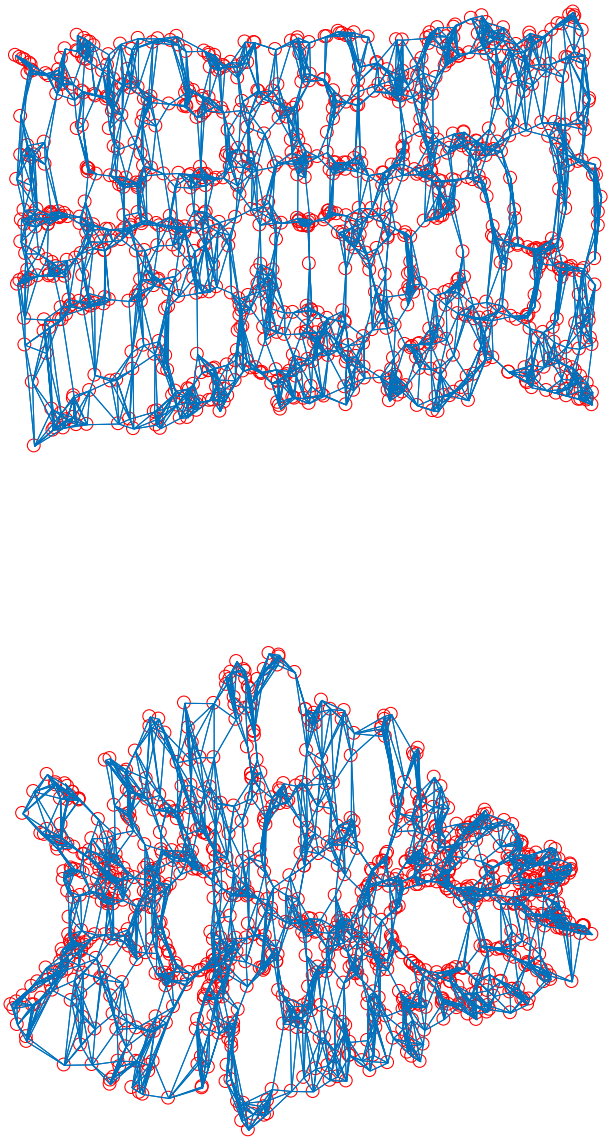


FIGURE 1.4: Low dimensional embedding of SR (upper graph) and HR (lower graph) using Isomap.

1.1.11 Locally Linear Embedding (LLE)

Local Linear Embedding (LLE) (Roweis and Saul, 2000b) is a method that is similar to Isomap, in that it builds a graphical representation of the data points. However, in contrast to Isomap, it attempts to preserve local properties of the data. As a result, LLE is less

sensitive to short-circuiting than Isomap, because only a small number of local properties are affected if short-circuiting occurs. Furthermore, the protection of local properties allows for successful embedding of non-convex manifolds. In LLE, the local properties of the data manifold are captured by expressing the high-dimensional data points as linear combinations of their nearest neighbours. In the low-dimensional representation of the data, LLE attempts to retain the reconstruction weights of these linear combinations as well as possible.

LLE describes the local properties of a manifold around a datapoint x_i with the help of a linear combination with weights w_i of its k nearest neighbours. Thus, LLE generates a hyperplane through the point x_i and its nearest neighbours, assuming the manifold is locally linear. The local linearity assumption make the weights w_i invariant to translation, rotation and rescaling during reconstruction in a lower dimension. In other words, if the high-dimensional data representation preserves the local geometry of the manifold, it can help to reconstruct the weights in a lower dimension.

1.1.12 Optimisation over Manifolds

Manifolds are of interest in optimisation theory because of their applicability to abstract geometry. Their ability to provide a geometrical understanding of higher dimensional surfaces, curves and volumes are the key to development of better smart algorithms, not only within optimisation theory but also for other problems with a complicated structure. The general terminology of optimisation is used in this study. It will always be assumed that the manifolds considered are sufficiently smooth.

A small neighbourhood of a point x in the manifold can be approximated in the Euclidean space. This phenomenon suggests that all numerical methods used for Euclidean spaces can be applied to a manifold.

An optimisation problem can be described as follows :

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{subject to} \quad & c_i(x) \leq 0,\ i = 1, \ldots, m. \end{aligned}$$

An optimisation problem is defined by minimisation (or maximisation) of a function over a set of constraints. The constraints are referred to as the equality and inequality constraints which represent a matrix of real numbers, typically $\mathbb{R}^n$ or $\mathbb{R}^{n\times p}$. The objective function or the constraint region leads to various optimisation techniques, including linear, non-linear and quadratic techniques. Another subclass of optimisation problems discussed in this paper is the optimisation of a function over a manifold. The general optimisation formulation can be used to formulate this kind of optimisation problem. We can directly use the manifold as a representation of equality and inequality constraints, such as with a Stiefel manifold (Kvernelv, 2013). However, in some cases, the constraint region may not be defined explicitly. Then, the optimisation problem becomes:

$$\min_{x\in\mathcal{M}} \quad f(x)$$

Where $\mathcal{M}$ is the search space manifold for the problem. It can be formulated as an unconstrained optimisation problem, but as the search space is a manifold, we need several geometrical concepts that can be used in optimisation theory, and we need gradients, directional derivatives and Hessians in the more general context of a manifold.

Recently, optimisation over manifolds has drawn attention as it can be extended to a vast variety of robotics and machine learning applications. Optimisation over manifolds has turned out to be more practical, as it can lessen the dimension of the problem in contrast with the ambient space. Its applications appear in medicine (Adler et al., 2002), signal processing (Manton, 2002), machine learning (Nishimori and Akaho, 2005), computer vision (Helmke et al., 2004; Ma et al., 2001), and robotics (Helmke et al., 2002a,b). Optimisation approaches on manifolds can be categorised into Riemannian approaches (Ring and Wirth, 2012; Smith, 1994) and non-Riemannian approaches. The current research focuses on non-Riemannian approaches.

1.1.13 Meta Heuristic Search for High Dimensional Data (Industry Data)

Service delivery organizations (SDOs) provide operations support, sales services and information technology services such as transaction processing. It is not uncommon for organisations to handle a large volume of data transactions (of the order of a few hundred

thousand) every day. In addition to the volume and dimension of data, organisations often face issues associated with the inter-arrival rate of the task and the different types of transactions that they handle. Furthermore, organisations have to meet SLAs agreed upon with clients, and are always required to perform better in terms of optimal cost and operational efficiency. An example SLA could be that 98% of transactions should be completed within one hour (i.e., the *turnaround time* of transactions should be less than one hour for at least 98% of transactions). Apart from SLAs, organisations typically define operational *key performance indicators* (KPIs) to monitor an organisation's performance and its way of working. Processing time of transactions (i.e., the actual amount of time spent in processing a transaction), resource utilisation and productivity are a few examples of KPIs (Mulla et al., 2016). Client-level SLAs are often translated into several operational KPIs in order to closely track an organisation's performance. For example, turnaround time of each transaction is broken down into expected processing times (referred to as baselines) for individual activities/tasks involved in the execution of the transaction. Once baselines are defined for each task, the organisation can keep track of the number of baseline violations and the magnitude by which these baselines are violated[2], and can take corrective actions (if need be) before progression to an SLA violation (in terms of turnaround time).

Task allocation is one of the key planning exercises that plays a significant role in an organisation's quest to meet SLAs and to achieve operational excellence. Employees within an organisation assume roles based on their skills, proficiency and experience. Since the execution of tasks requires specific skills, tasks need to be assigned to employees (referred to as resources) with appropriate skills/proficiency. Two resources possessing the same skills but having different proficiencies in those skills may take different times to process a task. In spite of all of these intricacies, most SDOs still resort to manual allocation of tasks.

Such manual allocation in SDOs is generally performed by a team lead. A team lead handles the incoming volume of transactions and, depending on the type of transaction and the tasks that need to be executed to process it, manually allocates the tasks to resources that she/he manages. While doing so, the team lead needs to consider a multitude

[2]Typically, SLAs are defined such that the penalty for violations variesy based on the magnitude of violation.

of factors such as task complexity, skill requirements, baseline processing times, resource skills/proficiency, workload, utilisation, fairness, etc. All of these factors need to be manually processed, making task allocation extremely challenging for team leads and making it almost impossible to keep in check the factors mentioned above; thus, there is a higher risk of impact on the KPIs and SLAs. Manual task allocation is further characterised by the danger of inherent biases that team leads may adopt. Such biases will impact resources, whose incentive payouts are dependent on the tasks that they process and their productivity. These factors warrant the design of efficient and automated task allocation schemes for SDOs to achieve operational excellence via fair task allocation, which involves better utilisation of resources, improved productivity, reduced costs and improved operational KPIs, thereby meeting the SLAs.

1.2 Contributions

This section outlines the main contributions of this thesis, and provides references to the appropriate chapters for each contribution.

1.2.1 Validation of Non-linear Dimensionality Reduction (Chapter 2)

A popular parameter to validate non-linear DR methods like Isomap, LLE and Maximum Variance Unfolding (MVU) is residual variance (Balasubramanian and Schwartz, 2002b). In this study, we develop and explore the topological stability of the data before and after DR. We attempt to calculate the persistence of point cloud data in both high and low dimensions. We show that PH can be used to validate the non-linear DR of a point cloud dataset. This method does not perform a nearest neighbour search or calculate residual variance; thus, calculation time can be independent of the dimensionality of data.

Further, PH homology calculation can provide the optimal number of sample points required for DR. Several experiments have shown that PH could be a handy tool to determine the minimum number of sample points for DR for point cloud data. In many tests, residual variance provides good dimensionality reduction whereas PH provides inferior dimensionality reduction.

One weakness of PH calculation for point cloud data is that although it can handle any dimensional data, it has a relatively high computation cost if the data set consist of massive sample points. To improve this, supervised deep learning methods were investigated as a follow-up study. The improvement in topology detection in Chapter 3 forms the next significant part of this work.

1.2.2 Topology Detection with Deep Learning (Chapter 3)

Although we used PH techniques for validating the DR technique, we faced another issue during our research, that is, that PH tools can be computationally costly if there is a large point cloud dataset. Analysis of a large dataset as a whole with PH is a prolonged process. To reduce the complexity of PH calculation, we used deep learning methods, which are very good at object recognition. Several recent studies (Cang and Wei, 2017; Hofer et al., 2017) have used deep learning techniques with topology properties to predict bimolecular property and topological signatures. However, (Cang and Wei, 2017) only used deep learning for a 3-dimensional biological image.

This study developed an approach to PH calculation for 2-dimensional and 3-dimensional point cloud data with a large number of sample points, reducing memory usage to linear in the number of data-points (Chapter 3). We also show that the deep learning framework can predict the topological properties of 2-dimensional and 3-dimensional data sets with a large point cloud.

For this study, the data set was generated with all kinds of variations that can be topologically complicated if we want to calculate PH through a traditional tool like Javaplex (Adams et al., 2014). Full details and results are provided in this Chapter.

1.2.3 Optimization over Manifolds (Chapter 4)

This thesis presents an approximate barrier algorithm for optimisation over the manifold, with a theoretical basis for lower-bound performance in Euclidean space. The line search algorithm with barrier method has a single parameter controlling approximation, which gives a smooth trade-off between the optimal result and searching speed. Comparisons

with a Matlab solver showed that for near accuracy levels, the approximate optimal point is very competitive regarding performance. We took the convex hull over the manifold to perform the starting point calculation. Optimisation over the manifold also provides a significant advantage over traditional non-convex methods, as this method is represented directly and in linear space, using data points.

1.2.4 Improving Operational Performance using Meta Heuristic algorithm (Chapter 5)

To obtain a better understanding of high dimensional data, we explored industry financial data while performing industry work with Xerox Research Centre India. The data was SDO data with transaction and resource details. SDO provides operations support, sales services and information technology services like transaction processing, around the clock. Large volumes of transactions (of the order of a few hundred thousand to millions) are processed every day, and this is projected to increase further in the coming years. It is imperative that organisations find ways to optimise their operations to meet the SLAs catering to the increasing workload. One means to achieve this objective is to explore ways of increasing the throughput of employees handling the transactions. Task allocation plays a crucial role in the performance of employees and in the satisfaction of SLAs (e.g., quality, turnaround time, etc.).

In Chapter 5, we proposed an algorithm and a method for allocating tasks to employees in services organisations. Chapter 5 discuss the difficulties of task allocation. Currently, our meta-heuristics algorithm outperforms the ILP with regard to time complexity when a large number of tasks and resources are used. Full details and results are given in Chapter 5.

1.3 Structure of Thesis

Topological analysis of point cloud data or high dimensional data has received a significant amount of research attention to date. Due to the increasing use of high-resolution images,

3-dimensional images and high dimensional real-world data, there is an increasing drive for research and development of faster and more memory efficient data analysis methods.

The utility of topological analysis becomes immediately apparent whenever we think of DR, learning methods, visualisation and optimisation, in which data resides on very high dimensional space, and we want to identify the source information. In this case, we try to explore extreme situations of topological analysis and optimisation. From a technology perspective, this type of task introduces several complex requirements.

Chapter 2

Validating Non-Linear Dimensionality Reduction Using Persistent Homology

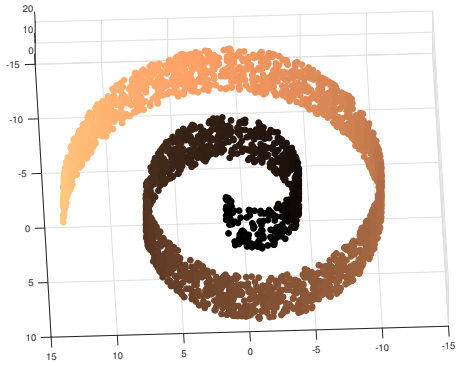
During the process of non-linear DR, manifolds represented by point clouds are at risk of altered topology. We review techniques for quality assessment of manifold learning and propose the use of PH to evaluate the topological impact of manifold learning by comparing the Betti numbers of test manifolds before and after DR. We propose a benchmark suite of test manifolds based on the Swiss roll dataset with added geometrical and topological complexity. The experiments demonstrate the effectiveness of the approach by analysing examples of test manifolds where the embedding failed. Betti numbers based on PH are also used to select suitable sampling rates for the manifold point clouds, and to determine optimal values for the nearest neighbour parameter k of selected manifold learning methods. The results indicate that the more complex the manifold is, the more sample points and larger values of k are required.

2.1 Quality Assessment of Manifold Learning

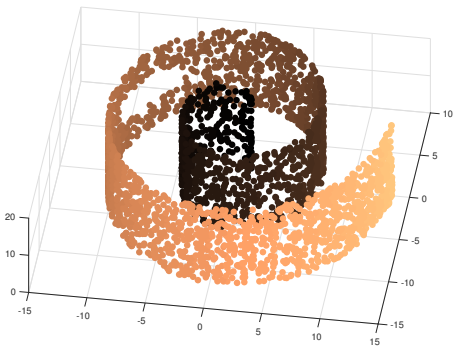
Following the emergence of a variety of methods for manifold learning that occurred after the proposals of Isomap and LLE in 2000 (Roweis and Saul, 2000a; Tenenbaum et al., 2000), it was recognised that there was a lack of criteria or explanation for how well manifold learning techniques perform for different data sets, and that additional measures would be required to assess their performance. Several more recent publications have specifically addressed the issue of quality assessment of manifold learning (Akkucuk and Carroll, 2006; France and Carroll, 2007; Gracia et al., 2014; Lee et al., 2014; Meng et al., 2011) and associated data visualisation (Venna et al., 2010). Most of these studies observed that good visualisations could be obtained from techniques that preserve the local neighbourhood around each point. It was noted that many studies had no quantitative measure to evaluate the quality of the outcome of DR, and had to rely on visual inspection alone (Venna et al., 2010). Of course, this is only possible for manifolds of one, two or three dimensions, and excludes manifolds that still have four or more dimensions after DR.

Gracia et al. (2014) provided an extensive review of quality assessment measures for DR techniques published between the years 1962–2012. Using a selection of 11 quality assessment measures, Gracia et al. (2014) provided a methodology that allows different DR methods to be compared using the concept of preservation of geometry. Their approach is aimed at helping researchers choose a suitable DR method for a given data set. Their experiments used 12 real-world data sets for testing. The results indicated that, among all algorithms tested, Isomap, MVU (Weinberger et al., 2004) and t-SNE (Van der Maaten and Hinton, 2008) were best at preserving the original characteristics of the data.

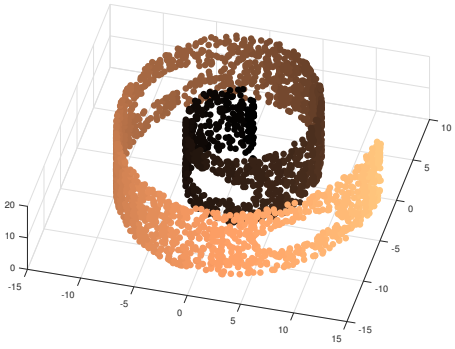
It should be acknowledged that many DR methods include random aspects and depend on parameter settings that can have an impact on an algorithm's performance (Mokbel et al., 2013). If the quality of DR methods is measured point wise in local neighbourhoods, this can help to improve the parameterisation quality measure in the evaluation process (Mokbel et al., 2013). A more precise assessment method uses a co-ranking matrix to determine the deviation of points from their original rank. However, a problem arises when trying to facilitate a more detailed evaluation, and an overall assessment and comparison of different visualisations.



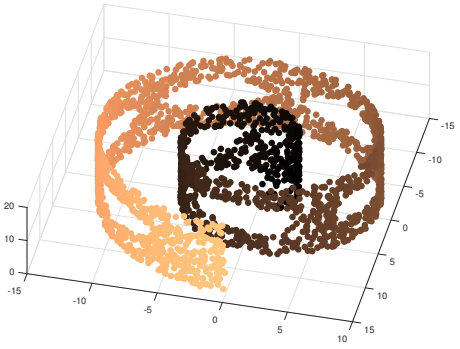
(A) Swiss Roll (SR)



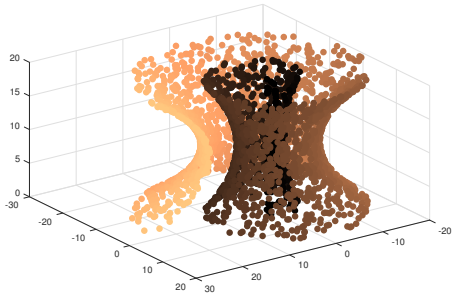
(B) SR with one hole



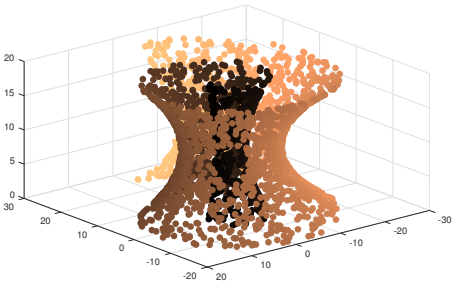
(C) SR with three holes



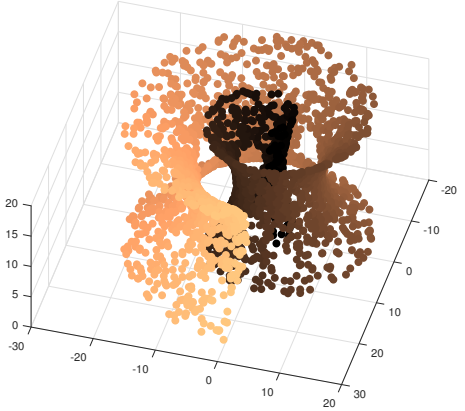
(D) SR with seven holes



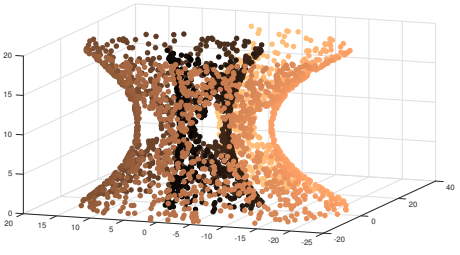
(E) Heated Swiss Roll (HR)



(F) HR with one hole



(G) HR with three holes



(H) HR with seven holes

FIGURE 2.1: A rectangular strip is curled, bent and punched to generate 8 datasets of different geometric and topological complexity: (a-d) Swiss Roll (SR) with two turns and 1, 3, and 7 holes; (e-h) Heated Swiss Roll (HR) with two turns and 1, 3, and 7 holes.

Another technique, termed Anisotropic Scaling Independent Measure (ASIM), can efficiently compare the similarity between two configurations of data points under rigid motion and anisotropic coordinate scaling (Zhang et al., 2012). Based on this technique, the embedding quality assessment method, termed NIEQA (Zhang et al., 2012), considers both the local and global topology of a data set to provide an overall assessment. This method depends on different neighbourhood sizes for local and global assessment.

The co-ranking matrix of Lee and Verleysen (2009) can be used to compare the ranks in the initial data set and in the data after DR. Rank errors and concepts such as neighbourhood intrusions and extrusions can be associated with different blocks of the co-ranking matrix, and can be used as quality assessment criteria for DR. Rank-based criteria have been extended to scale independent quality measure methods, given that most DR techniques rely on a scale parameter that distinguishes local from global data properties (Lee and Verleysen, 2010).

An early quality assessment method that can also provide an estimate of suitable target dimensionality is residual variance, which was used by Tenenbaum et al. (2000) in association with Isomap. Several of the discussed articles noted that one of the most popular quality assessment methods for DR is the average agreement rate, which compares the k-neighbourhoods in the high- and the low-dimensional space (Akkucuk and Carroll, 2006; Lee et al., 2014; Lee and Verleysen, 2009).

The comparison of local neighbourhood structures in many of the discussed quality assessment methods emphasises the need for assessment of the preservation of local geometry during DR, rather than assessment of the global topology. Lee et al. (2014) even excluded spectral methods such as Isomap, MVU and LLE from their study as these methods have difficulty with clustered data and tend to deform.

The present study focused on global topological measures, and not geometrical properties, in quality assessment, and aimed to complement some of the previously discussed methods. The topological aspects have rarely been addressed in this context in the existing literatures. An article on this topic that is very closely related to the present study and also used PH for the evaluation of DR schemes was undertaken by Rieck and Leitte (2015). Their analysis framework did not directly refer to Betti numbers but was based

on comparing persistence diagrams. Rieck and Leitte (2015) had slightly different goals as compared to the present research, and included several DR techniques; they used both Swiss roll data and real-world data. The history of quality assessment is given below.

TABLE 2.1: Quality assessment history

1962	Sheppard Diagram (SD)
1964	Kruskal Stress Measure (S)
1969	Sammon Stress (SS)
1988	Spearman's Rho (SR)
1992	Topological Product (TPr)
1997	Topological Function (TF)
2000	Residual Variance (RV)
2000	König's Measure (KM)
2001	Trustworthiness & Continuity (T&C)
2003	Classification error rate
2006	Local Continuity Meta-Criterion (Qk)
2006	Agreement Rate (AR)/ Corrected Agreement Rate (CAR)
2007	Mean Relative Rank Errors (MRRE)
2009	Procrustes Measure (PM)/Modified Procrustes Measure (PMC)
2009	Co-ranking Matrix (Q)
2011	Global Measure (QY)
2011	The Relative Error (RE)
2012	Normalization Independent Embedding Quality Assessment (NIEQA)
2014-2015	Norm Concentration (Frobenius norm between the gram matrices)

2.2 Experiments

The approach of using PH for topology evaluation during DR can be applied to any manifold learning method and to manifolds of any dimension. However, some non-linear DR methods, such as t-SNE (Van der Maaten and Hinton, 2008), focus on disentangling and visualising data, while other methods try to preserve global topological or geometrical properties of a connected underlying manifold. Accordingly, the focus of the present study was on two representatives of the latter set of methods; t-SNE and related methods were not included.

The experiments employed two of the most well-known manifold learning techniques, namely, Isomap and LLE, which both use an underlying k-nearest neighbour graph on the manifold data.

The task was to obtain 2-dimensional embeddings of the Swiss roll and the heated Swiss roll, and six of their variations of increasing complexity with 1, 3 and 7 holes (Fig. 2.1).

One aim was to demonstrate the effectiveness of using PH for evaluating manifold learning. This was achieved by comparing evaluations using the common measure of residual variance (Tenenbaum et al., 2000) with our proposed Betti number analysis. Another aim was to demonstrate how to use Betti numbers to estimate the minimum number of sample points required for successful application of DR, and to determine an optimal k-nearest neighbour parameter k for each case.

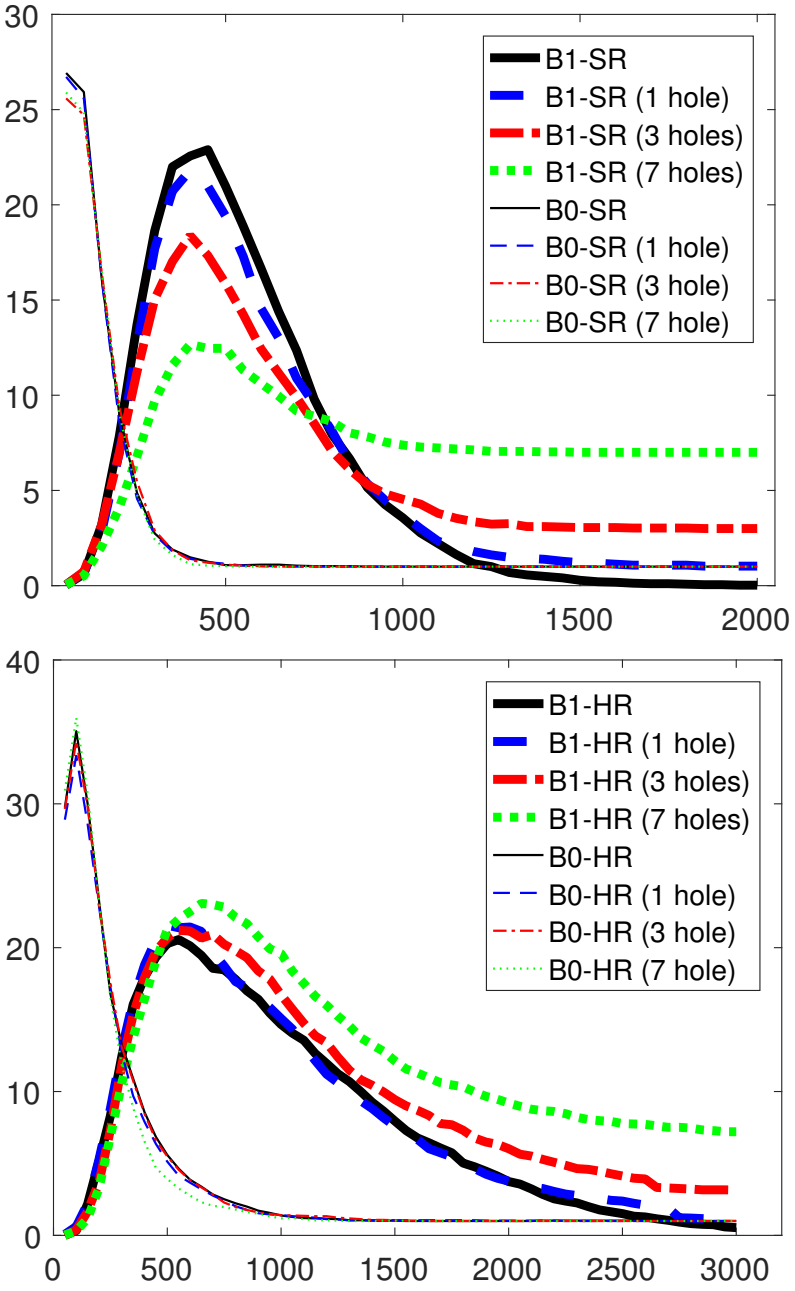


FIGURE 2.2: (a) Betti numbers of the Swiss Roll data in 3-dimensions dependent on the number of sample points; (b) Betti numbers of the Heated Swiss Roll data in 3-dimensions dependent on the number of sample points. Betti number behaviour dependent on the number of sample points (x-axis) before dimensionality reduction. The results for B_0 in both graphs show that for sample sizes below 500, the point cloud is not yet recognised as one connected manifold but as a set of many disconnected components. For 1000 or more sample points, B_0 converges to 1 indicating correctly that each manifold consists of one connected component. The curves for B_1 converge correctly to 0, 1, 3 or 7 for the manifolds with the corresponding number of holes. Each curve represents the average of the same 40 samples but with a different number of holes. The numerical results indicate that the B_0 and B_1 curves converge more slowly the more complex the manifold is.

2.2.1 Software

The experiments for manifold learning either used the original software (in Matlab) as provided by the authors of Isomap (Tenenbaum et al., 2000) and LLE (Roweis and Saul, 2000a) or their implementation in the toolbox provided by van der Maaten et al. (2009). Both methods use a k-nearest neighbour graph and a suitable parameter k had to be selected upfront for each method and manifold.

The Betti numbers for the example manifolds were calculated before and after DR using the Javaplex software package for PH, as described by Tausz et al. (2014b). The computational demand of calculating the simplicial complexes and Betti numbers can be very high and investigations are currently underway to achieve efficient computations (Otter et al., 2017). Javaplex is well-documented and suitable for smaller data sets, as used in the present study. It offers several options to calculate PH and then generates barcode diagrams and associated Betti numbers as output (Tausz et al., 2014b). We chose the Vietoris-Rips option in Javaplex to calculate the PH. A critical parameter that requires calibration is the 'maximum filtration value' R. Tausz (2012) recommended to start with a small value and then scale it up gradually in a series of tests, while observing the barcode diagrams, until essential persistence bars can be identified. This process was followed for each data set in the present study in a series of pilot experiments. Finally, we settled on $R = 3.5$ for the data sets before DR and $R = 4.7$ for the embedded point clouds after DR.

2.2.2 Data

This study used eight synthetic datasets of different geometrical and topological complexity (Fig. 2.1).

The 'Swiss Roll' dataset (SR) (Fig. 2.1a) is a two-dimensional strip that is curled as a spiral embedded in $\mathbb{R}^3$ (Tenenbaum et al., 2000; van der Maaten et al., 2009). Let $x = (x_1, x_2) \in [-1, +1]^2$ be uniformly distributed. Then a parameter representation of SR

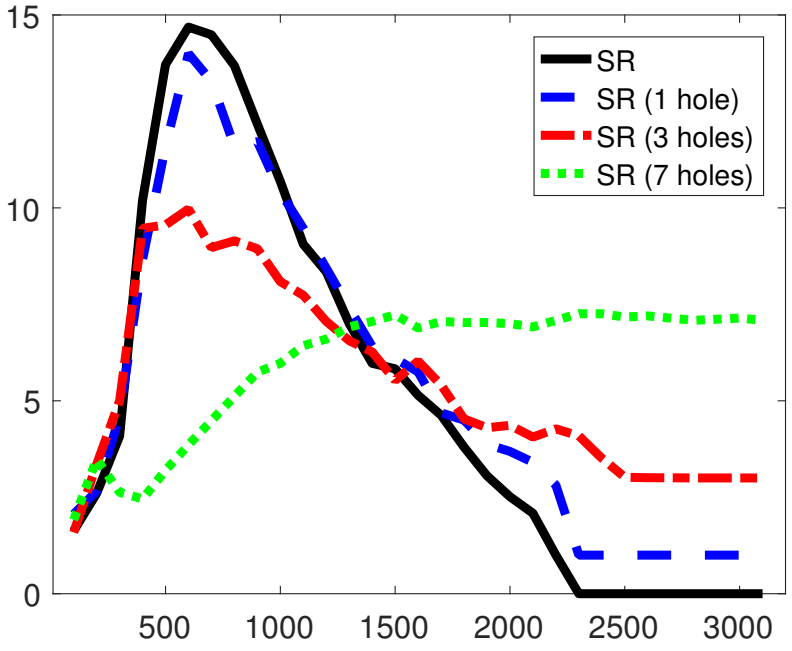
where each coordinate of the manifold depends on a single latent variable is given by:

$$\begin{bmatrix} (1 + 2\pi\sqrt{x_1})\cos(2\pi\sqrt{x_1}) \\ (1 + 2\pi\sqrt{x_1})\sin(2\pi\sqrt{x_1}) \\ 20x_2 \end{bmatrix} \tag{2.1}$$

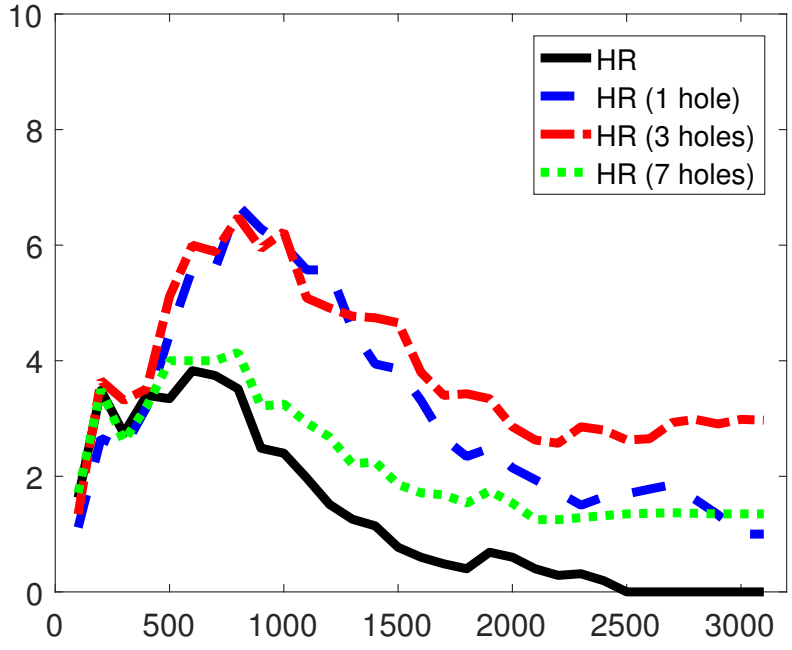
The 'Heated Swiss Roll' (HR) (Fig. 2.1e) is a variation of SR with added curvature in the z-direction (Lee and Verleysen, 2007). A parameter representation where the first two coordinates depend on both latent variables x_1 and x_2 is given by:

$$\begin{bmatrix} (1 + (2x_2 - 1)^2)2\pi\sqrt{x_1}\cos(2\pi\sqrt{x_1}) \\ (1 + (2x_2 - 1)^2)2\pi\sqrt{x_1}\sin(2\pi\sqrt{x_1}) \\ 20x_2 \end{bmatrix} \tag{2.2}$$

To scale the topological complexity of the data, we used an additional three variations of the SR data inspired by the so called 'Japanese flag manifold'; Fig. 2.1a-2.1h show SR and HR with 0, 1, 3, and 7 holes, respectively.



(A)



(B)

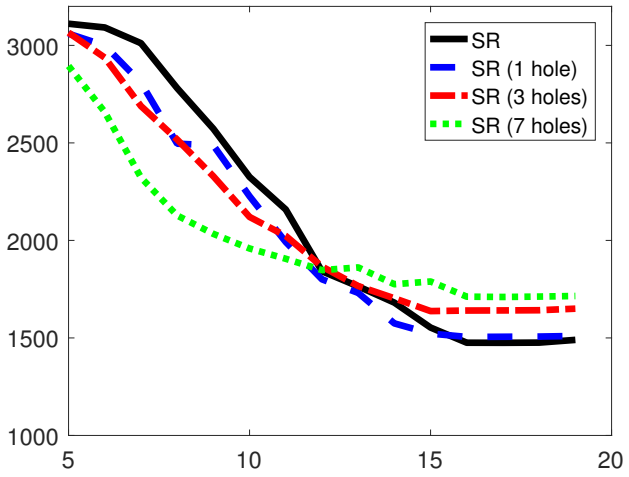
FIGURE 2.3: (a) B_1 for the embedded SR with different numbers of holes; (b) B_1 for the embedded HR with different numbers of holes. Betti number behaviour dependent on the number of sample points after dimensionality reduction with 7-Isomap. The plots show B_1 is dependent on the number of sample points for each of the eight data sets. Each curve is the average of B_1 for 18 embedded manifolds. With increasing numbers of sample points, B_1 for most manifolds converged to the correct value. An exception is the case of HR with 7 holes in (b), where most embeddings failed and the mean curve ended up lower than 7.

2.2.3 Analysis

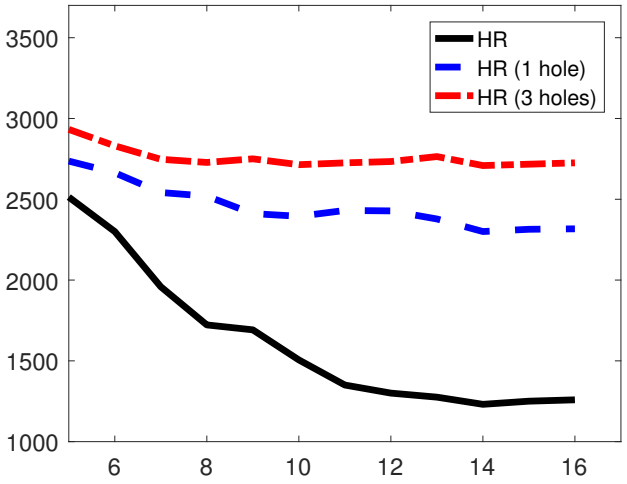
The impact that DR can have on the manifolds is seen in Fig. 2.5 and 2.6; these figures show successful and unsuccessful Isomap embeddings. The following experiments aim to quantify this visual observation. They compare the topological properties of our data sets before and after DR.

2.2.4 Topological Analysis Before Dimensionality Reduction

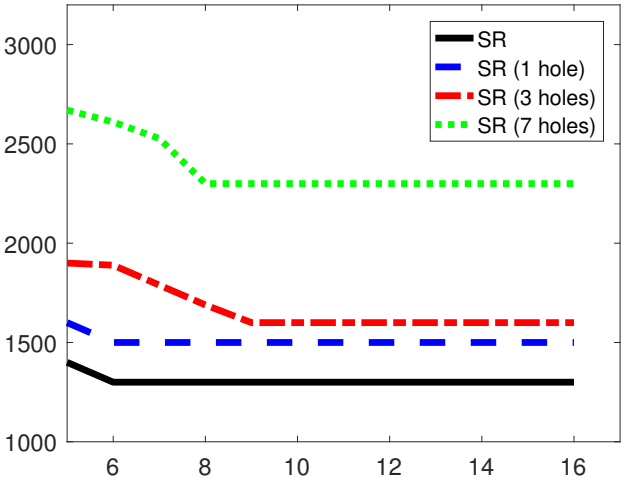
Fig. 2.2 shows (on the y-axes) Betti numbers B_0 and B_1 calculated for the eight data sets of Fig. 2.1 before application of manifold learning. Each curve shows the mean for 40 random samples. None of the considered manifolds had 3-dimensional cavities, and therefore, B_2 was always 0 and is not displayed. The results in Fig. 2.2 show that B_0 (= number of connected components) quickly converges to 1. B_0 converges faster for SR than for the more complex HR. The count of circular holes, B_1, converges to the expected values of 0, 1, 3 and 7 after about 1700 sample points for SR and after 2100 sample points for HR. These results show that the convergence rates can be slower for geometrically and topologically more complex manifolds.



(A) Isomap



(B) Isomap



(C) LLE

FIGURE 2.4: (a)Isomap; (b) Isomap; (c) LLE. The lines in the graphs show the lower bound on the sample size (ordinate) above which B_1 converges correctly after application of manifold learning. The experiments were run for several k (abscissa). The graphs show that more sample points are required if the manifold is more complex. Due to the very small standard deviation, the error bars were not visualised.

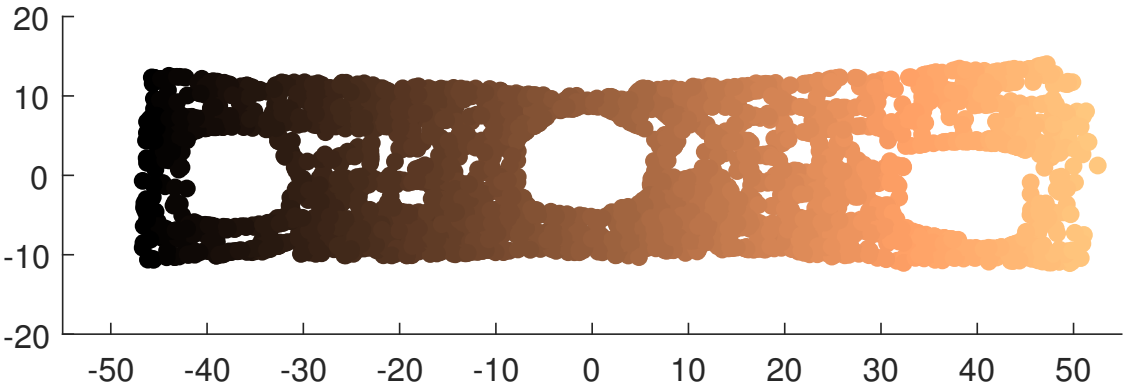
2.2.5 Analysis of Manifold Data after Dimensionality Reduction

In order to detect and maintain the topology type of manifolds represented by point cloud data before and after DR, a certain amount of data is required. If there are not enough data points, the manifold cannot be recognised and its topology cannot be calculated correctly. Fig. 2.3 shows Betti number B_1 (y-axis) for the 2-dimensional embeddings of our test manifolds with regard to sample size (x-axis). All embeddings were conducted using Isomap with $k = 7$. Isomap can be very unstable, and to reduce the impact of outliers, each curve shows the average of the 18 best runs out of 30. Results obtained for LLE were similar to those with Isomap and are therefore not displayed.

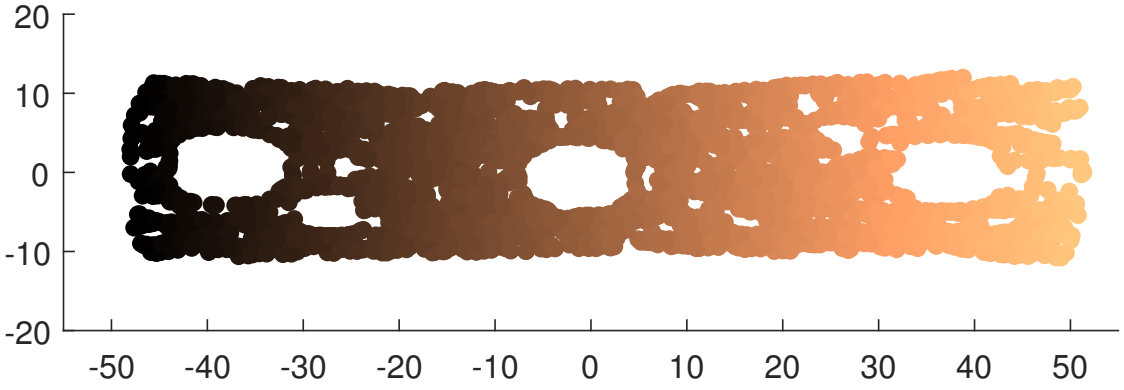
After DR, B_1 behaves similarly for seven different versions of SR and converges to the expected values when the sample size comes close to 3000. Most of the displayed curves go up first because, for low sample sizes, the manifold has many holes. The green-dotted curve in Fig. 2.3a behaves slightly differently because it represents SR with 7 holes, and for low sample sizes, the 7 holes seem to merge into one big hole. Most embeddings of HR with 7 holes (Fig. 2.1h) behaves slightly different because it represents HR with 7 holes, and for low sample sizes, the 7 holes seem to merge into one big hole. Most embeddings of HR with 7 holes (Fig. 2.3b).

We also calculated the residual variance for all embeddings with Isomap using the tool that comes with the Isomap software. The residual variance is calculated using $1-R^2(D_M, D_L)$ where R is the standard linear correlation coefficient taken over all entries of the two distance matrices, D_M are the estimated geodesic distances on the manifold before embedding and D_L are the Euclidean distances after embedding (Tenenbaum et al., 2000). The residual variance is a useful measure to determine in which dimension to embed a manifold.

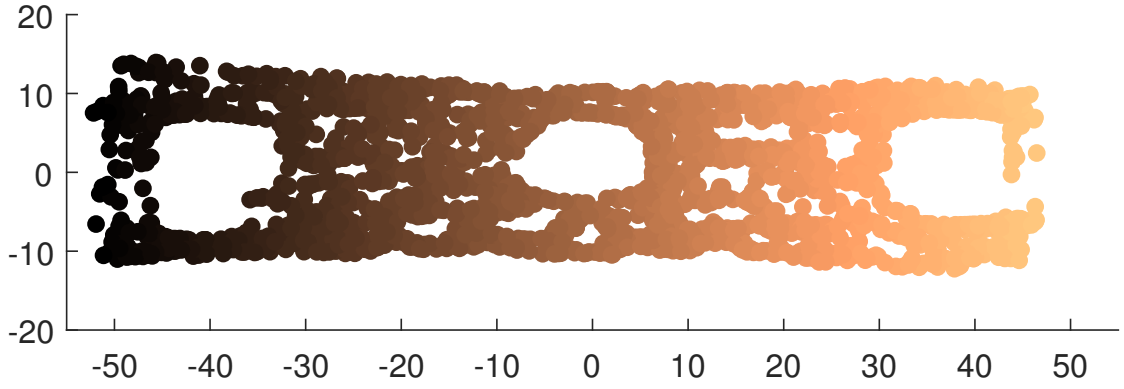
In our experiments, we aimed at determine if an embedding is successful or not from a topological point of view. The following examples highlight the strength of our proposed topological approach in comparison to the residual variance.



(A) Successful embedding

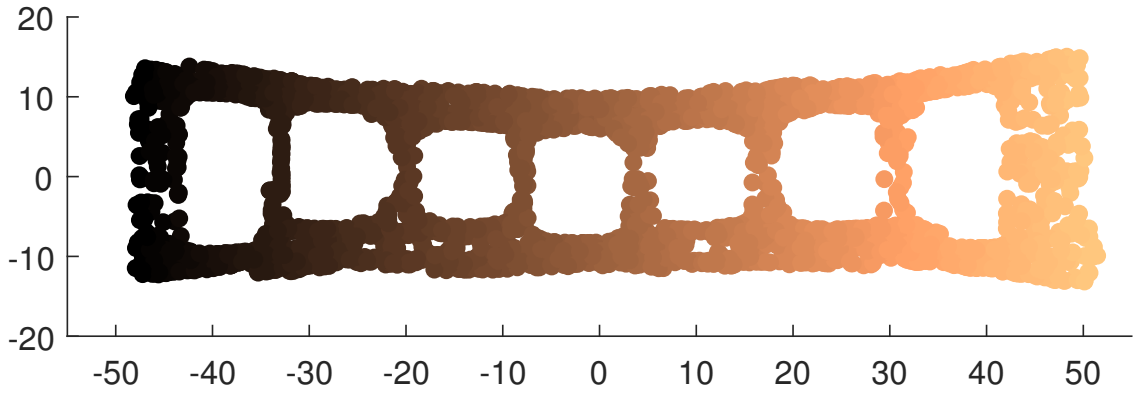


(B) A significant 4th hole occurred on the left,

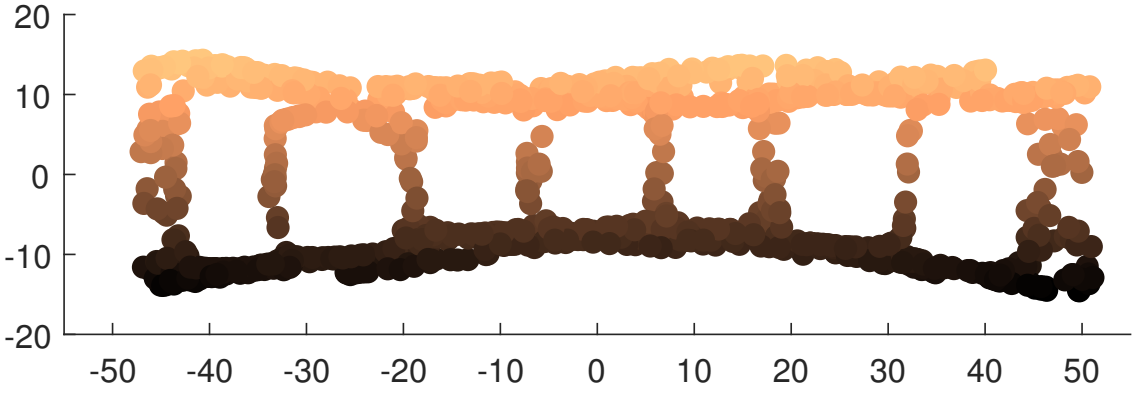


(C) The hole on the right was ripped open.

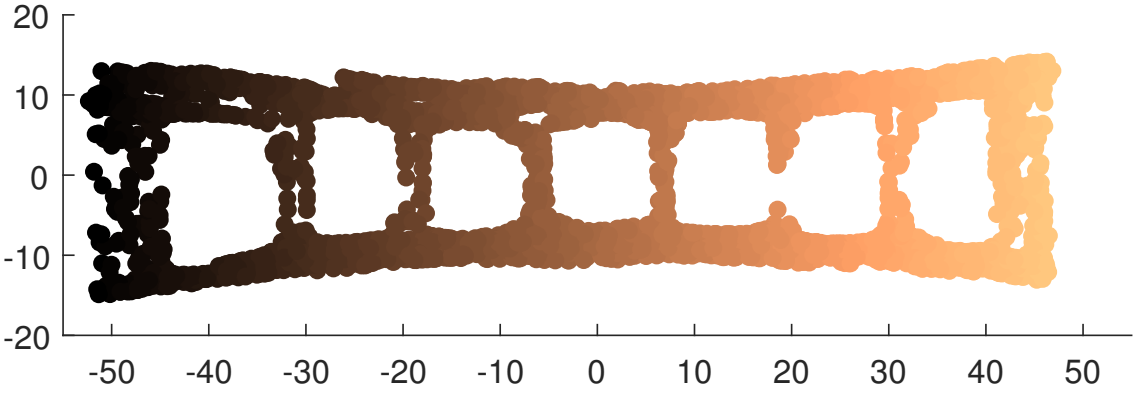
FIGURE 2.5: Two-dimensional Isomap embeddings (with $k = 7$) of SR with 3 holes; (a) a 3100 point Swiss Roll sample that was successfully embedded from 3D into 2D; (b) a 3100 point Swiss Roll sample with 3 holes where a 4th hole occurred on the left during embedding; (c) a 2100 point Swiss Roll sample with 3 holes where the hole on the right was ripped open during embedding.



(A) Successful embedding of a 3100 point sample



(B) Successful embedding of a 1000 point sample



(C) A 3100 point sample where hole 5 and 6 merged

FIGURE 2.6: Two-dimensional Isomap embeddings (with $k = 7$) of a Swiss Roll with 7 holes; (a) a 3100 point sample that was successfully embedded from 3D into 2D; (b) a 1000 point sample that was successfully embedded; (c) a 3100 point sample where the 5th and 6th hole were merged during embedding.

Example 1 Fig. 2.5a shows a successful embedding of SR with 3 holes represented by a point cloud of 3100 sample points. We repeated this embedding with Isomap ($k = 7$)

30 times for different draws of the 3100 sample points. The mean residual variance of this experiment was $1.32 \cdot 10^{-3}$ (std $0.33 \cdot 10^{-3}$). The mean Betti number B_1 for this experiment was 3.2 (std 0.41). Interestingly, the example in Fig. 2.5a had $B_1 = 3$ as expected but a relatively high residual variance of $2.28 \cdot 10^{-3}$, while the example in Fig. 2.5b had $B_1 = 4$ and a relatively low residual variance of $0.89 \cdot 10^{-3}$. This indicates that the topological measure can detect the accidentally occurring fourth hole in Fig. 2.5b while the residual variance cannot detect it. The values for the residual variance indicate, in this case misleadingly, that the embedding in Fig. 2.5b would be of better quality than the one in Fig. 2.5a.

Example 2 When reducing the sample size of SR with 3 holes to 2100 points, holes can break open, seen on the right side in the example in Fig. 2.5c. The residual variance of this example was 0.0021 and close to the mean of the batch of 30 samples and did not indicate that anything was wrong with the embedding. However, the assessment using Betti numbers returned $B_1 = 2$ as a warning that one hole was lost.

Example 3 Fig. 2.6a and 2.6b show that successful embeddings of SR with 7 holes are possible for samples of 3100 points but also for 1000 points (Fig. 2.6b). Both examples resulted in $B_1 = 7$. Fig. 2.6c had 3100 sample points and resulted in $B_1 = 6$ because two holes seemed to merge into one hole. For all three examples in Fig. 2.6, the residual variance was close to the mean of the batch and did not indicate any topological differences in the outcome.

2.2.6 Topological Analysis of the Manifold Data after Dimensionality Reduction: Dependency of B_1 Convergence on Sample Size and k-Nearest Neighbour Parameter

Fig. 2.4 shows how the final convergence of B_1 was affected by different k in Isomap and LLE. For each experiment an average of 30 runs was used. The standard deviations were relatively high in the range 350-550. The graphs indicate how the different values for k (x-axis) relate to different sample sizes (y-axis) at the point of convergence of B_1. The

ideal neighbourhood parameter k for Isomap on the test manifolds appears to be in the range 14-16. For LLE $k = 6$ for SR and SR with one hole, and $k = 8$ or 9 for SR with 3 or 7 holes (Fig. 2.4c). All graphs confirm that test surfaces with more holes require higher sample sizes.

2.3 Chapter Summary

This study described a validation approach for non-linear DR techniques using PH. In a series of pilot experiments, the topology of point cloud data before and after DR was compared using the calculation of PH Betti numbers. As this only uses the data before and after embedding, it can, in principle, be applied to any DR method. The study focused on Isomap and LLE; that is, two established manifold learning methods that use a k-nearest neighbour graph and try to respect the global topology of the data.

Data was sampled from different versions of increasing complexity of the Swiss roll. The experiments showed that the more complex the manifold is in terms of curvature or topology, the more sample points are required to calculate its topology. It was demonstrated that the proposed method can help to provide a lower bound for the number of sample points required to allow correct manifold learning (Fig. 2.4).

We identified examples where the proposed method could detect a topological change caused by the embedding, but the traditional measure of residual variance could not.

Another contribution was the description of a method that helps to decide on a suitable k for Isomap and LLE.

The experiments were restricted to variations of the Swiss Roll, but could be extended in future research to real-world data sets, and could be applied to manifolds of any dimension.

A general issue for manifold learning or associated quality evaluations occurs when the distribution of sample points is highly heterogeneous and does not capture the topological or geometrical characteristics of the underlying manifold. If, for example, certain areas of an image manifold are missing or cannot be captured in the data, then even a large increase in the number of sample points cannot fix the problem and the global topology

of the manifold may remain unknown. Therefore, in the present study, we assumed that the sample points of our simulated data were approximately evenly distributed.

A limitation of the presented approach is that PH cannot detect if an embedding of a manifold causes rips in the edge of a manifold, nor can it detect geometrical distortions that do not change the connectivity or topology type of the manifold.

In summary, with its ability to detect topological changes in any dimension, the proposed approach complements the methods reviewed in Section 2.1 and could be used in conjunction with them to achieve a more comprehensive quality assessment of manifold learning. Using combinations of the proposed and the reviewed approaches to evaluate embeddings of topologically complex high-dimensional real-world data is a task for future investigations.

Chapter 3

Topology Calculation for 3-dimensional Data

Topology and its various benefits are well understood within the context of 2-dimensional (2D) data sets. However, requirements in 3-dimensional (3D) applications have yet to be explored. It is clear that, with the rapid increase in the amount of data produced, the availability of efficient tools to analyse these data is of great importance. Due to its ability to extract essential topological features of the parsed data, PH is becoming a widely-used method. PH was introduced by Edelsbrunner et al. (2000) and has drawn much attention as it robustly extracts the topological structure of data. To study these data quantitatively requires efficient algorithms for processing large 3D images and for extracting topological and geometrical measures. After successful calculation of PH for a manifold, we wanted to apply our methods to a 3D real-world data set. One of the most fundamental descriptions of a structure is via homology: the mathematical characterisation of connectivity, including connected components, independent loops and enclosed voids. An essential ingredient to use homology in the study of experimental and computational data is to build a filtration (a nested sequence of cell-complexes) that captures the topology of the data concerning a parameter, usually a length-scale, that dictates the order in which complexes are added. Topological features (such as a hole through an object) are born at some parameter value and are later merged or filled in at a larger value. Features that are created and destroyed almost simultaneously are considered noise. Features that persist over longer parameter

ranges are deemed to be more critical. While algorithms with good practical running times have been proposed (Chen and Kerber, 2011), exact computation of persistence for sizeable 3D stack image data remains a challenge due to the massive memory requirements.

The under usage of PH in real world applications is because it is very computationally costly. The general PH algorithm (Edelsbrunner and Harer, 2010) takes $O(n^3)$ even for small sized data (e.g., 64× 64× 64). In addition to the higher complexity, there are three further issues: (1) the memory consumption of the currently available implementations, like Javaplex and Ripser[1], is very large even for a small point cloud; (2) the focus of several applications is on data of higher dimensions, e.g., 4D, 5D or higher; and (3) not only does the data lie in 4-dimensional, 5-dimensional or higher dimensions, the data are also point cloud data, which makes the computation even more complex.

In the previous chapter, we used Javaplex and Ripser to calculate the PH of the SR and HR datasets. We also described the optimal sample points required for the dataset in order to obtain stable topology. In the case of 3D or higher real-world data, the existing algorithm does not scale well with the increase in sample points, thus introducing larger computational times and memory inefficiency. In a later section, we will provide the detailed analysis of our experiments.

The contributions of the present study are as follows:

- We attempted to calculate Betti numbers in PH to evaluate their impact on the topology type of the manifold that underlies a given set of points.
- We used a 3D stack image and calculated PH with deep learning.
- We described the full analysis report for Javaplex with the computation cost.
- We were unable to complete the whole experiment because of the computational cost; this study attempted to use a Convolutional neural network to obtain the PH in a supervised manner.

[1]https://github.com/Ripser/ripser

3.1 2D Image Slices to Point Cloud

Images can be representative of complex materials. Processing of an image can be performed quickly and in exquisite detail. By definition, image processing is a field that is concerned with the computation and information contained in images. This chapter aims to analyse the images of complex material through PH, which can build a bridge between algebraic topology and applications. However, sometimes, the images can be of high resolution, and this can cause a problem for the computation of PH. We aimed to calculate the correct PH of the whole 3D data. The data we have used to calculate PH was 3D complex material data from Department of Mechanical Engineering, The University of Newcastle. The datasets were the Micro-Computed tomography (MCT) image slice of complex material. This type of complex materials is called syntactic foams. According to Taherishargh et al. (2017), researchers are working to produce these materials cost-effectively. If we can explore the topology of these kinds of material, it would be easier to produce these materials. This motivates us to use this data set.

These data set are consist of image slices. The data used was slices of one cylinder of complex material. These are often called stack images. Each image represents a slice of complex material. Each of the slice images is of 744×684 pixels. To analyse this size of an image through Javaplex and Ripser takes a substantial amount of time and storage. That's why these images must be converted to point cloud.

To convert each slice image to point cloud data, we first must convert each image into a binary image with a threshold of 0.25.

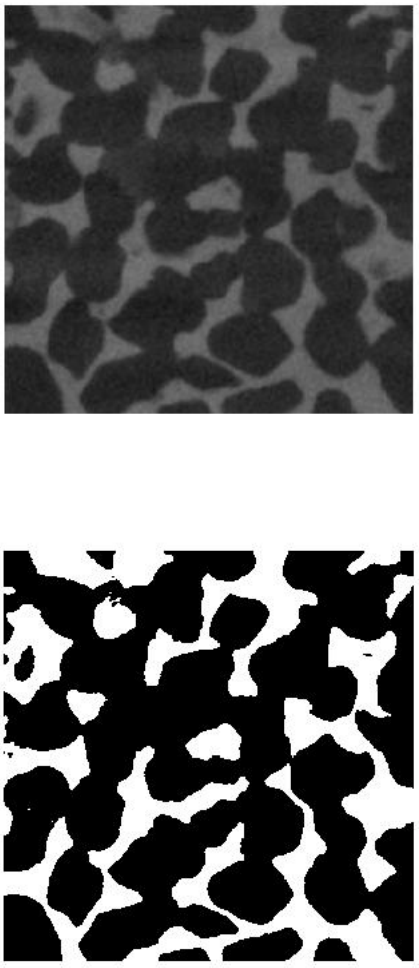


FIGURE 3.1: (a) Original Image; (b) after binarisation, black pixels denote the holes in the image and white pixels are the complex material

Fig. 3.1 shows only a sub-image from the middle of the original image 250×250. Then, we find the distribution of the pixels where the pixel value equals to 1, and begin sampling according to the distribution. After sampling, the image becomes a point cloud dataset with coordinates of the data points. We used the random sampling technique to obtain the point cloud from the distribution. After we converted the image to the respective point cloud data set, we applied Javaplex and Ripser to obtain the PH. As we want to apply our previous methods explained in Chapter 2 and know about the minimum sample points required to achieve correct topology, we have showcased the two different version of

point cloud to visualise the distortion of the topology with the number of sample points. Fig. 3.2 shows the point cloud of the slice image with 4000 and 40000 sample points.

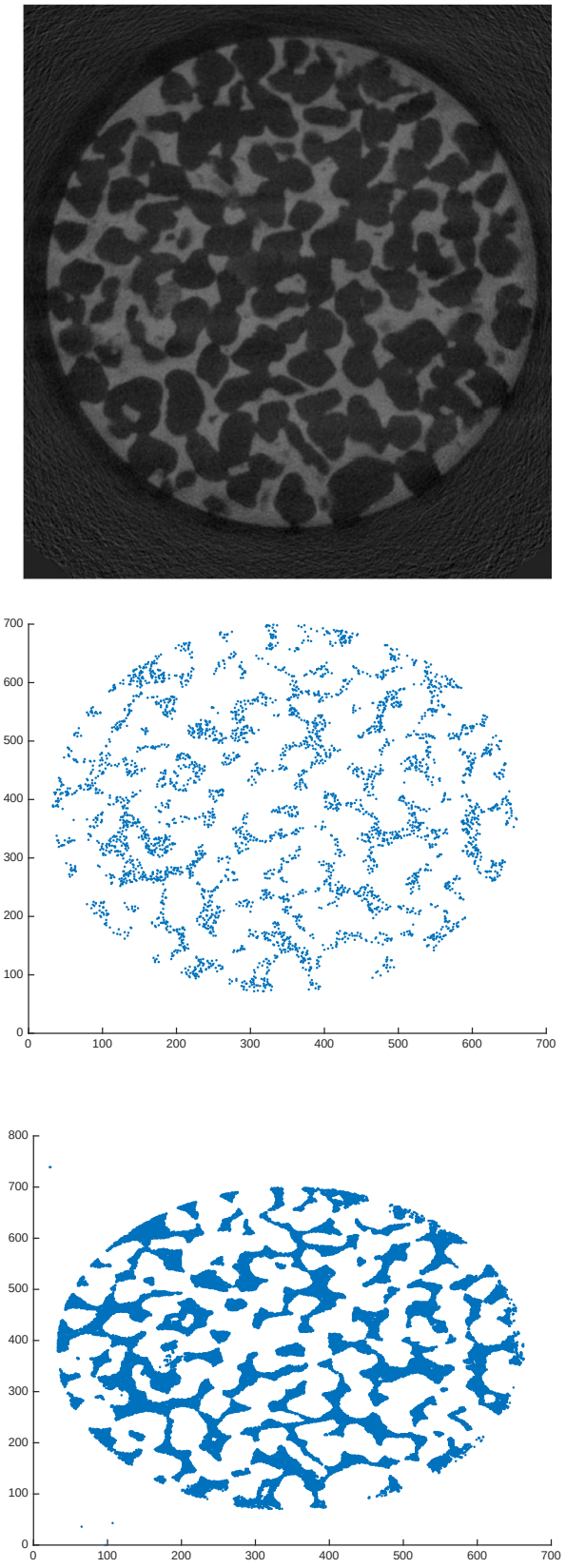


FIGURE 3.2: Slice image converted to point cloud (a) Original slice image, (b) 4000 sample points and, (c) 40000 sample points.

After we converted the image to the respective point cloud data set, we applied Javaplex and Ripser to obtain the PH.

3.1.1 2D Slices to 3D Data

2D slices of point cloud data are used to generate 3D data of complex material. We put every point cloud slice vertically to each other, at a distance of 1 cm. We then tried to feed the 3D point cloud data to Javaplex to obtain the topological analysis for the given data set. We first started with the 4000 sample points per slice. Each slice has been converted into 4000×2 matrix where it has the x and y coordinates of 4000 sample points. Similarly all the 775 slices are converted to similar matrices. When we merged all the 775 slices of size 4000×2, the point cloud data became of size $4000 \times 2 \times 775$. We have used 4000 sample points for each slice of the cylinder. Javaplex required more than 500 GB of RAM to compute the PH of the 3D data. It is hard to visualise 3D data from the 2D slices because: 1) all the 2D images are of the same dimension; hence, the 3D volume can hold all of them in a rectangular cube or cylindrical shape; 2) the majority of the pixels in each of the 2D images has a 3D spatial relationship, and it is difficult to visualise if each of the 2D images is of some random distribution.

Fig. 3.3 shows the point cloud data generated from the complex material. Part (c) of Fig. 3.3 shows the actual complex material in 3D. Part (b) shows the point cloud generated from the complex material. Part (a) is the top view of a slice image after converting it to a point cloud.

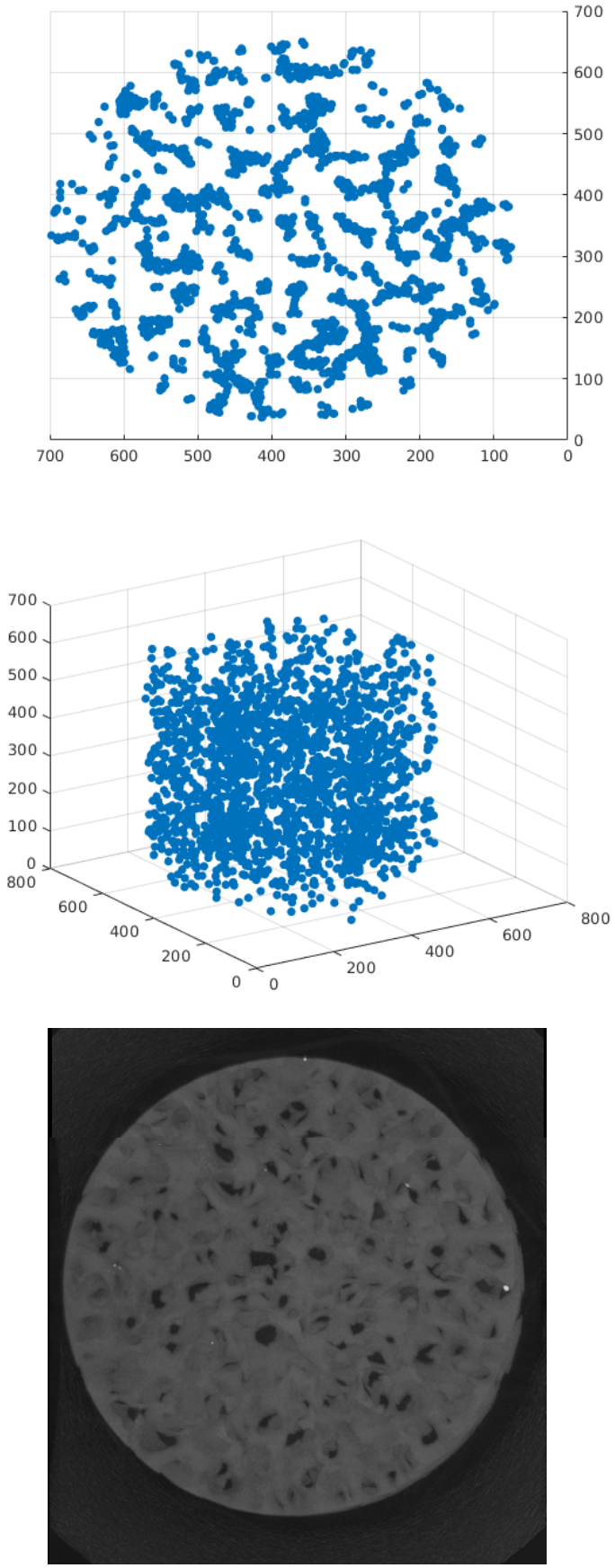


FIGURE 3.3: (a) Original slice point cloud with (b) $4000 \times 2 \times 775$ point cloud data and (c) 3D view of the original data.

3.1.2 Experimental Set-up

One aim was to demonstrate the effectiveness of using PH for evaluating the use of Betti numbers to estimate the minimum number of sample points required for 3D complex material. This resulted in successful calculation of the minimal volume that must be considered when determining material specific topological characteristics. The computational demands of calculating simplicial complexes and Betti numbers can be very high, and researchers are currently investigating ways to achieve efficient computations within a suitable time complexity (Otter et al., 2017). Javaplex offers several options to calculate the PH and then generates barcode diagrams and associated Betti numbers as output (Adams et al., 2014). We chose the VietorisRips option in Javaplex to compute the PH. As we know from Chapter 2, one of the critical parameters that requires calibration during the experiments is the 'maximum filtration value' R. In the case of SR, we can easily test and estimate the R value, which has been explained in Chapter 2. However, for 3D data, the estimation of the R value with pilot experiments is very time-consuming. However, we used the inbuilt functions of Javaplex, 'createRandomSelector' and 'landmark selector', to determine the R for the 3D data; eventually, we settled on $R \approx 9000$ for the data set.

3.2 Analysis

We started our experiment with 3D point cloud data of size 4000×3. From Fig. 3.4, it can be observed that B_0 in the graph indicates that for sample sizes below 500, the point cloud is not yet recognised as one connected manifold, but as a set of many disconnected components. For 1000 or more sample points, B_0 converges to 1 indicating correctly that the 3D data consists of one connected component. However, the curves for B_1 do not indicate successful convergence for the manifolds with the corresponding number of holes. Each curve represents the average of the same 10 3D data samples, but with different examples. The numerical results indicate that the B_0 and B_1 curves converge slower the more complex manifold is. Fig. 3.5 shows the bar code diagram for the same data obtained from Javaplex.

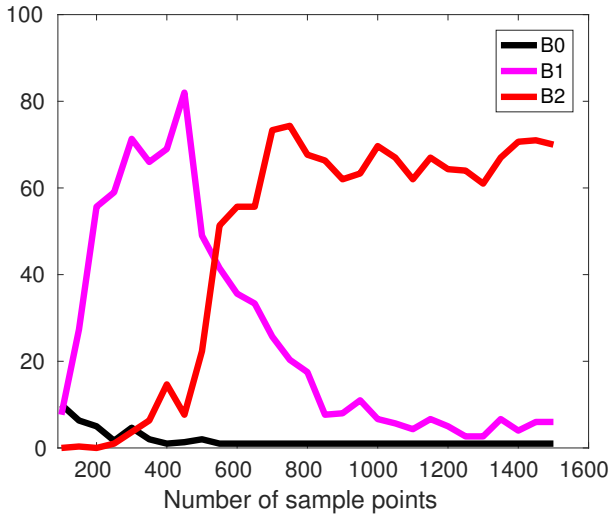


FIGURE 3.4: Betti number behaviour dependent on the number of sample points (x-axis).

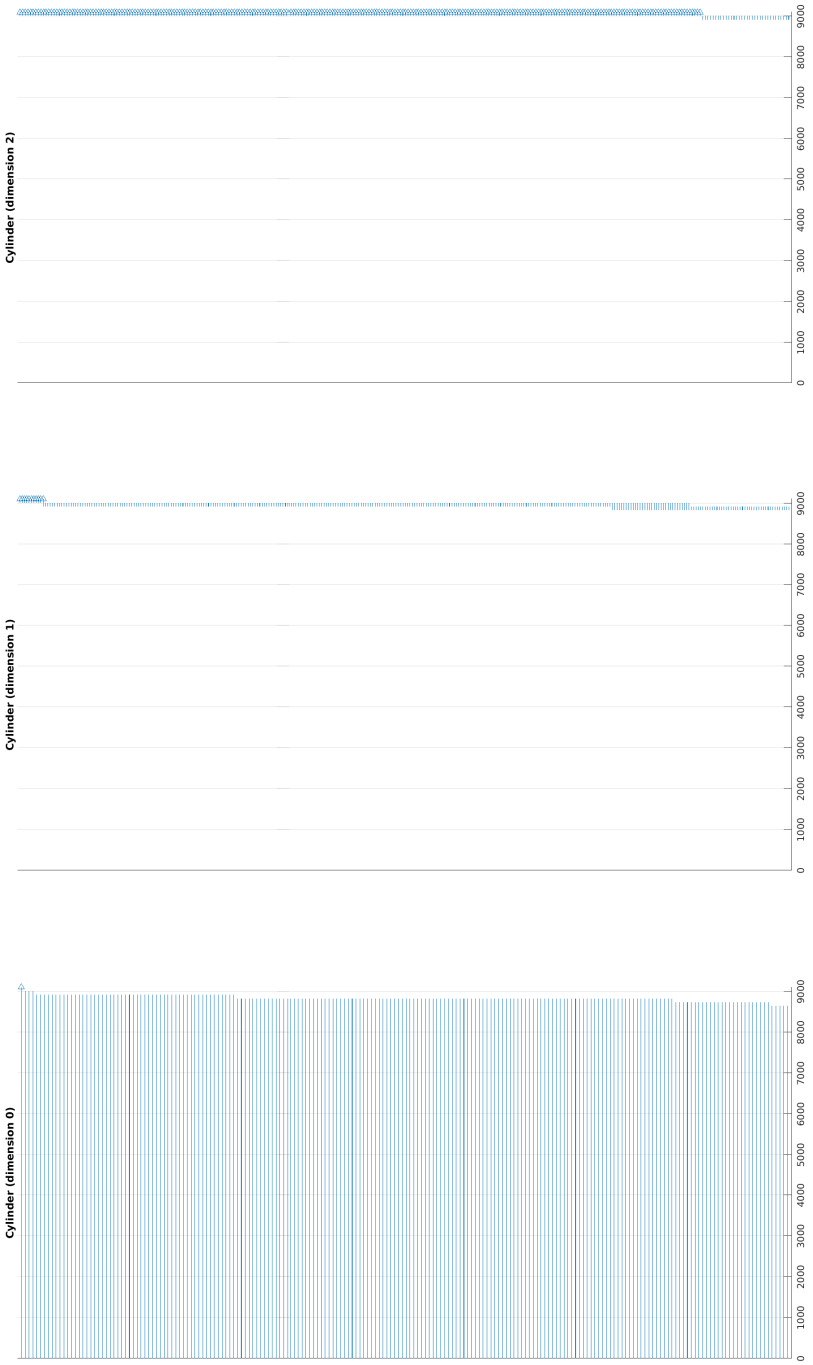


FIGURE 3.5: Bar code diagram for 3D data (x-axis).

However, these experiments required more physical memory with increasing numbers of sample points. This phenomenon occurs due to the increasing number of simplexes as the number of sample points increases. For 800-900 sample points, Javaplex was running for 18 hours and using 48 GB RAM, and the number of simplexes for 800 sample points with 3D data was around ≈ 335485, which grows exponentially as the number of sample points increases. From Fig. 3.6, we can clearly see as we increase the sample points, the number of simplices increase exponentially. Results are obtained experimentally from Javaplex.

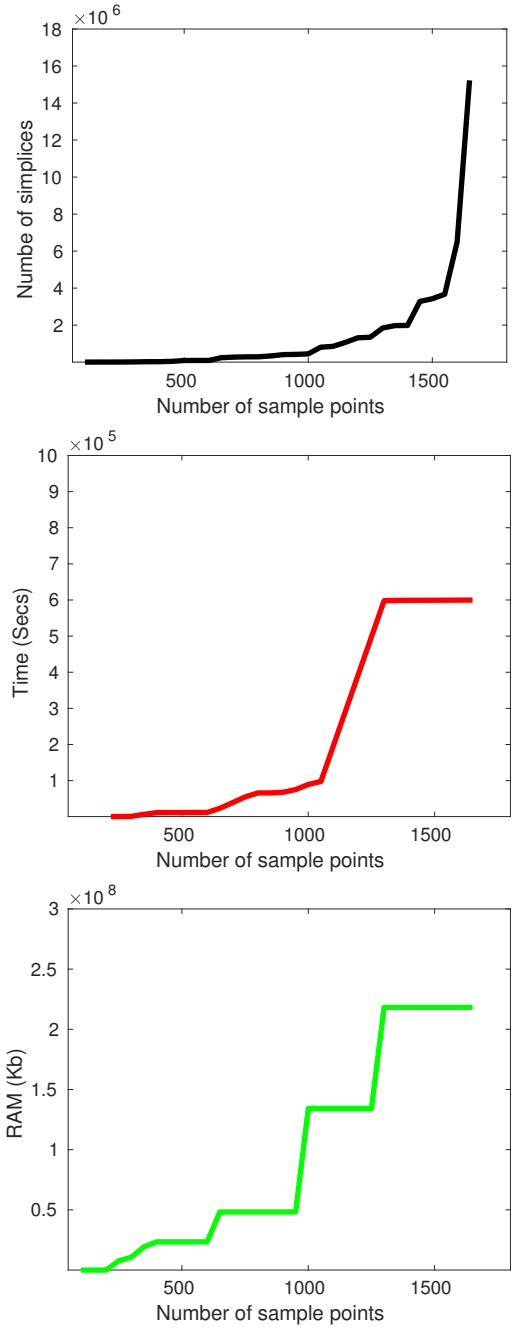


FIGURE 3.6: (a) Number of simplexes, (b) time to calculate the homology, and (c) physical memory required for PH calculation for 3D data set. The results are shown for up to 1700 sample points, after which computation of PH failed due to the huge memory requirement.

From Fig. 3.6, it can be seen that, requirements in memory and time increase rapidly with the number of sample points. For example, for 300 sample points, the number of simplices were 10000, and for 600 sample points, the numbers of simplices became 90000. Javaplex

failed because it was requesting more than 128 GB of RAM after running for almost 139 hours. We then increased the Java-heap size to 400 GB and RAM requirement to 500 GB to rerun the experiment with a 400 hour limit. However, this was only able to give a result for up to 1700 sample points. Processing becomes stranded up to 400 hours. From Fig. 3.6, it can be seen that for successful PH calculation for 3D data, we require $\approx$ 166 hours and $\approx$ 218 GB of RAM. However, we were using only a low-resolution stack image data of 600×600 per slice. How can we solve this problem with less computation cost? How can we detect the holes or Betti numbers for 3D data?

3.3 Deep Learning for Topological Analysis of Data

Although deep learning (DL) techniques had great success in computer vision and machine intelligence, their application to 3D data sets has been unexplored under topological complexity. Recent studies (Cang and Wei, 2017; Hofer et al., 2017) have used DL techniques with topology properties to predict bimolecular properties and topological signatures. However, (Cang and Wei, 2017) only used DL for 3D biological data. There is some recent literature describing the ability to 'vectorise' the space of the barcode, where it can then be fed to standard DL techniques (Adcock et al., 2013). Moreover, Adams et al. (2017) converted a persistence diagram to a 'persistence image', which can be later vectorised for use with various learning techniques. Later, Hofer et al. (2017) used a persistence diagram to compute parametrised projection, which can be used in DL techniques. However, all of these approaches are applicable for 2D data. Can we use 3D point cloud to understand topological signatures? To investigate this, we could simply generate (lots of) simulated data and test appropriate deep nets in a supervised manner.

In this chapter, we explore deep learning architecture capable of reasoning the topological structure of a data set. The 3D voxel grids or collection of images renders data unnecessary voluminous and causes issues. Point cloud data are simple and unified structures that avoid the combinatorial irregularities and complexities of meshes, and thus are easier to learn from. To get uniform results from convolutional architecture, most researchers typically transform the point cloud to collection of images before feeding them to deep net

architecture. However, complex material data can be more voluminous and thus we tried to convert the images to the point cloud data.

3.3.1 Topological Data Analysis for 2D Using Deep Nets

Instead of using a topological signature as data, we used the images as input data to a deep neural network to detect topology of the point cloud. For example, Hofer et al. (2017) used a persistent diagram as a feature after vectorisation. Instead of doing that, we created 2D point cloud data with various holes in it. We then converted the 2D point cloud to an image as input data for the Convolutional Neural Network (CNN). Each image was labelled with B_0 and B_1, as for 2D, B_0 and B_1 are essential for topological analysis. This technique has been used for all of the 2D point clouds in the current study. As a first step in our DL experiment, we create the point cloud data set by:

$$\begin{bmatrix} 2(x_1) \\ 2(x_2) \end{bmatrix} \tag{3.1}$$

In the Eq. 3.1 we take $x = (x_1, x_2) \in [a, b]$ be randomly distributed. In all of our experiments, we took $a = 0$ and $b = 8$. Using Eq. 3.1, the point cloud data set is generated with any number of sample points $(1000, 2000, .., 4000)$ using the distribution $[a, b]$. Then, we insert the various topological features to the point cloud. Below we have explained the algorithm to insert various 1-dimensional holes.

Algorithm 1 Topological Structure

Input: Point cloud $coords2$, no of holes n.

Pick one point from the point cloud randomly as centre of hole c.

Initialize μ.

for $i = 1$ **to** n **do**

Pick one point from the point cloud as centre of hole c.

If c is first centre and distance from image $margin <= 2$

Add to the list of centre of holes $centers$.

$centers \leftarrow c$.

Else

Check for other points with distance as we do not want to intersect.

Take random radius between $r = [1, 2]$.

repeat

Calculate the distance from $c \in centers$.

If $dis_c < r$

Remove the point from the $coords2$.

until $centers = empty$

end for

With the help of Algorithm 1, we generated many point clouds with various numbers of holes, positions and radius. Then we saved them as tiff images.

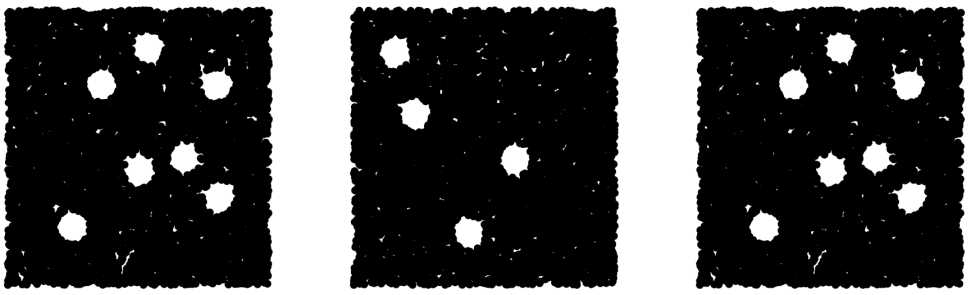


FIGURE 3.7: (a)7 holes, (b) 4 holes, (c) 7 holes with different positions

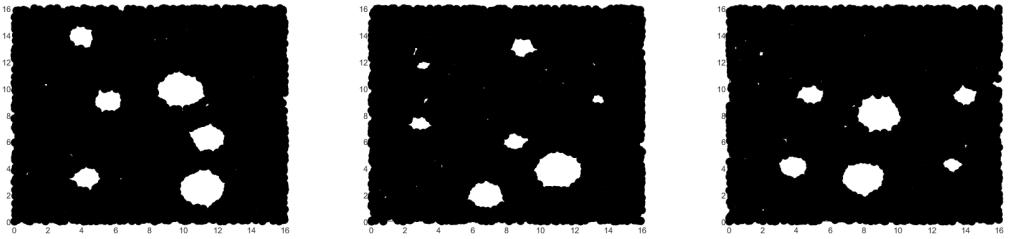


FIGURE 3.8: (a) 6 holes; (b) 7 holes; (c) 6 holes with different positions and variable sizes

Fig. 3.7 shows a sample of the first set of input data which was used for training in the CNN. Each image data has been labelled regarding B_0 and B_1. For example, for Fig. 3.7(a), $B_0 = 1$ and $B_1 = 7$. We generated 20000 images for this category and tried to predict the Betti numbers using a CNN. The results showed that CNN can predict Betti numbers with 100% accuracy. To increase the complexity of our data, we introduced more complex structures, like islands, into our 2D data set. The idea behind this was to ensure the network can not only detect the object but also understand the topological structure of the data. This study attempted to make the B_0 and B_1 values variable for each of the images in order to make learning complex for the CNN. Fig. 3.8 shows the data with variable size holes.

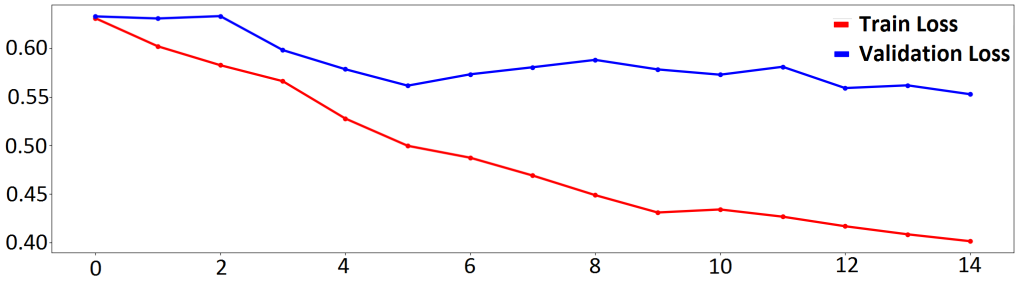


FIGURE 3.9: Train and Validation loss curves for 2D stack images for 15 epochs (x-axis)

3.3.2 Model

Although the full network is shown in Fig. 3.10, the essential architectural choices are the input layer and box size, which depend on the image size. We have a three stage

convolutional network to work with slices of images. The first layer consists of a 2D convolutional layer with 32 filters, each sized (5,5). This is followed by a max-pool 2D layer with pooling size of (2,2). The same configuration is then used again for another three 2D convolutional layers with 64, 128 and 256 filters, respectively. The last stage is a standard two-layer fully connected neural network with 1024 units and 20% drop out. This is a simple approach to capture the hole count among the filtration directions. We used cross-entropy loss to train the network for 15 epochs for the 2D data.

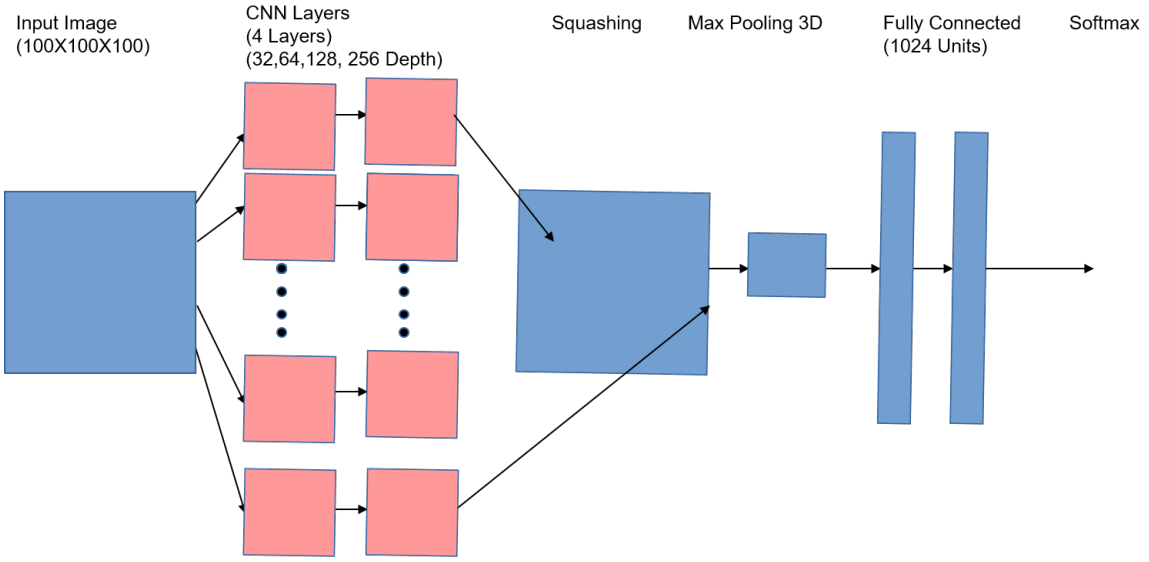


FIGURE 3.10: Example of basic network architecture with double convolutional layer. Four convolutional layers (Conv1, Conv2, Conv3, Conv4), one max pooling layer, a fully connected layer, and lastly, a fully dense layer, where a final softmax is performed. Cross entropy loss is used and trained with Tensorflow with Adam Optimiser

The current CNN model follows the Fig. 3.10 architecture. The model used Conv1, Conv2, Conv3 and Conv4, respectively. From the picture, we can observe that the predicted output approximates the pattern of the ground truth image well.

3.3.3 Result for 2D Images

A summary of the experimental results on the test set can be seen in Fig. 3.9. Models with 32, 64, 128 and 256 filters outperformed the models with 15 filters. The best model had four layers, 32, 64, 128 and 256 filters, and a 1024 hidden dimension in the fully connected layers. We have used ReLU function for each neuron activation. According to

Krizhevsky et al. (2012), ReLU function $f(x) = max(0, x)$, train several times faster in convolutional neural network. In our experiment, the ReLU activation function performed the best and was used for all other experiments. The success of 2D convolutional neural networks on this dataset verifies the assertion stated earlier, that the model is robust to rotation and noise. The best architecture can beat the best baseline model by an accuracy of 100%. However, the experiments showed that for the same hole size dataset, CNN can easily detect Betti numbers which are very trivial. That is why we introduced the variable hole size in the data set, which can be difficult for the neural network. Fig. 3.8 shows the data set with variable hole sizes. The best mode's rotational invariance is seen in Fig. 3.9. Table 3.1 shows the final result with the 2D data sets.

TABLE 3.1: Convolutional Neural Network accuracy for 2D data

Image Size	Sample Points	accuracy
100×100	2000	60.4%
250×250	2000	77.8%
250×250	4000	81.9%
250×250	8000	88.9%
600×600	4000	91.4%
600×600	8000	94.7%
600×600	10000	99.7%

3.3.4 Checking B_1

Although the previous 2D CNN model gave good accuracy with simple holes in the data set, a neural network model cannot provide understanding of the topology of the data. To further explore this, we created various forms of data, with various numbers of holes and islands, to defer the B_0 and B_1 numbers. Algorithm 2 explains how we put islands into the manifold data set.

With the help of Algorithm 2, we generated a different type of data set, which can be topologically disconnected.

Algorithm 2 Topological Structure with Islands

Input: Point cloud $coords2$, no of holes n.
Pick one point from the point cloud randomly as centre of hole c.
Initialize μ.
for $i = 1$ **to** n **do**
  Pick one point from the point cloud as centre of hole c.
  If c is first centre and distance from image $margin <= 2$
  Add to the list of centre of holes $centers$.
  $centers \leftarrow c$.
  island $\leftarrow$ Choose randomly from $centers$.
  Else
  Check for other points with distance as we do not want to intersect.
  Take random radius between $r = [1, 2]$.
  Take inside radius for islands between $r_1 = [1, 2]$.
  repeat
    Calculate the distance from $c \in centers$.
    If $dis_c < r$
    Remove the point from the $coords2$.
  until $centers - islands = empty$
  repeat
    If $dis_c < randdis_c > r_1$.
    Remove the point from the $coords2$.
  until $islands = empty$
end for

From Fig. 3.11 we can see that, not only have we have changed the number of holes, but we have also changed the number of components in the data set. For example, Fig. 3.11(a) and (d) has the same number holes. However, their B_0 is two and five, respectively, as they have a different number of islands. We have used the same network architecture for the island data sets. Table 3.2 shows the summary of the results obtained from the island data sets.

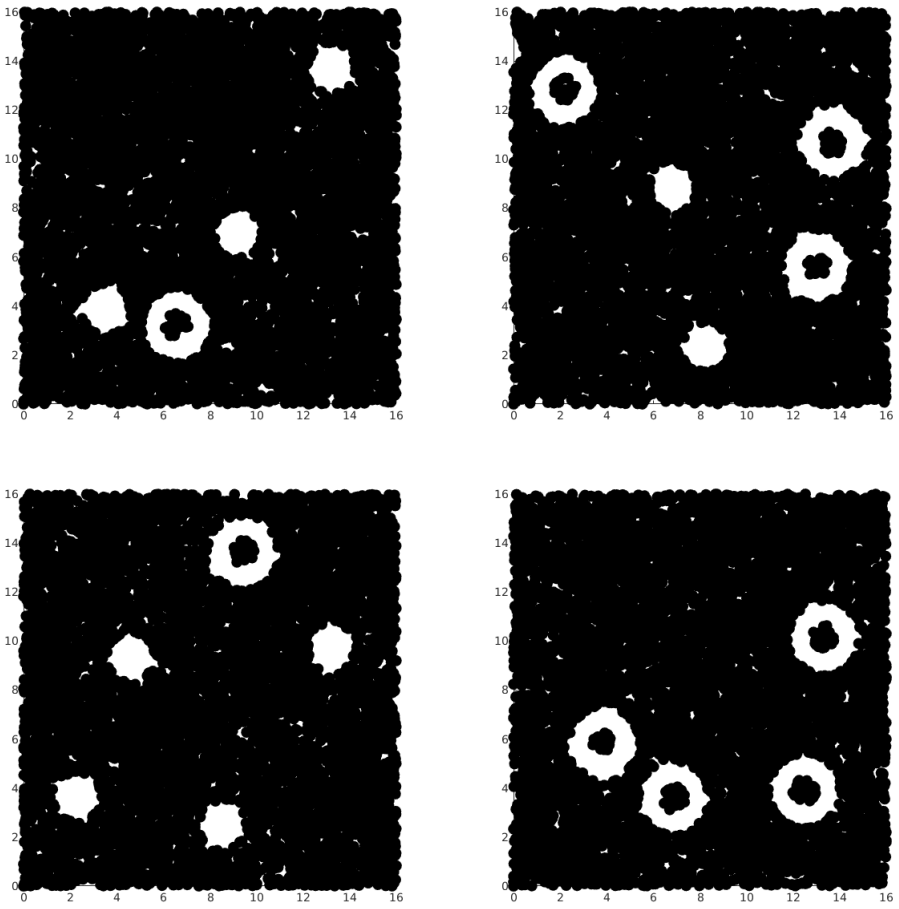


FIGURE 3.11: (a)4 holes and 1 island, (b) 5 holes and 3 islands, (c) 5 holes and 1 island, (d) 4 islands

TABLE 3.2: Convolutional Neural Network accuracy for 2D island data

Image Size	Sample Points	accuracy
100×100	2000	59.4%
250×250	2000	74.8%
250×250	4000	79.9%
250×250	8000	84.9%
600×600	4000	88.8%
600×600	8000	91.4%
600×600	10000	98.9%

3.4 3D Data

The primary objective of this study was to represent the difficulties encountered if we consider 3D point cloud data for topology detection. Previous analysis has shown that there are very high computational demands when attempting to detect correct topology of a data set with a high number of sample points. Experiments on retinal images have been carried out on the DRIVE (Staal et al., 2004) dataset, which includes 40 eye fundus images and contains manual segmentation of the blood vessels by expert annotators. Much recent research has shown that network topology can be detected using CNN (Ventura et al., 2017). However, only 2D image data has been used to detect path level topology in the Massachusetts Roads dataset. In our study, we want to detect topology of geometrical shapes or manifolds. To understand the PH of a 3D point cloud data set, we have synthetically generated manifolds with different shapes and different sizes of holes inside the manifold, in order to vary B_1 and B_2.

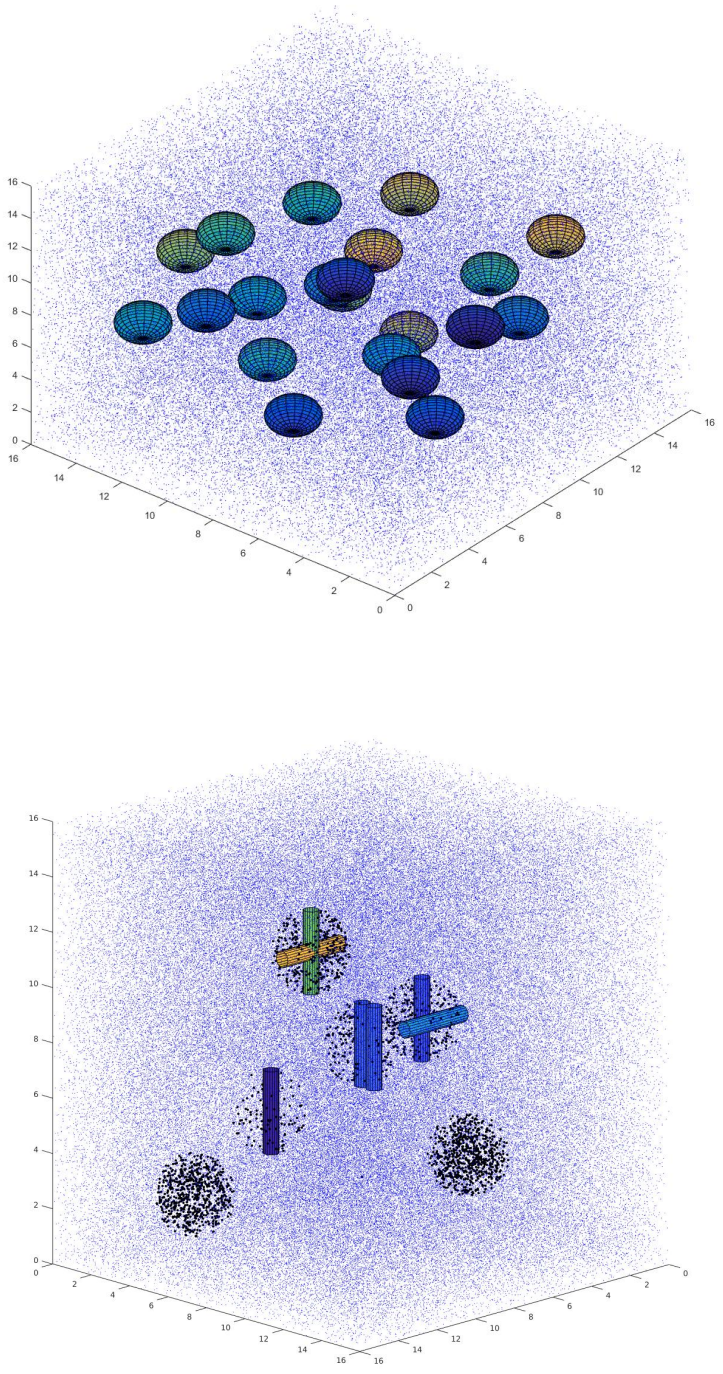


FIGURE 3.12: (a) 3D manifold with only random bubbles; (b) 3D manifold with cylinder and torus.

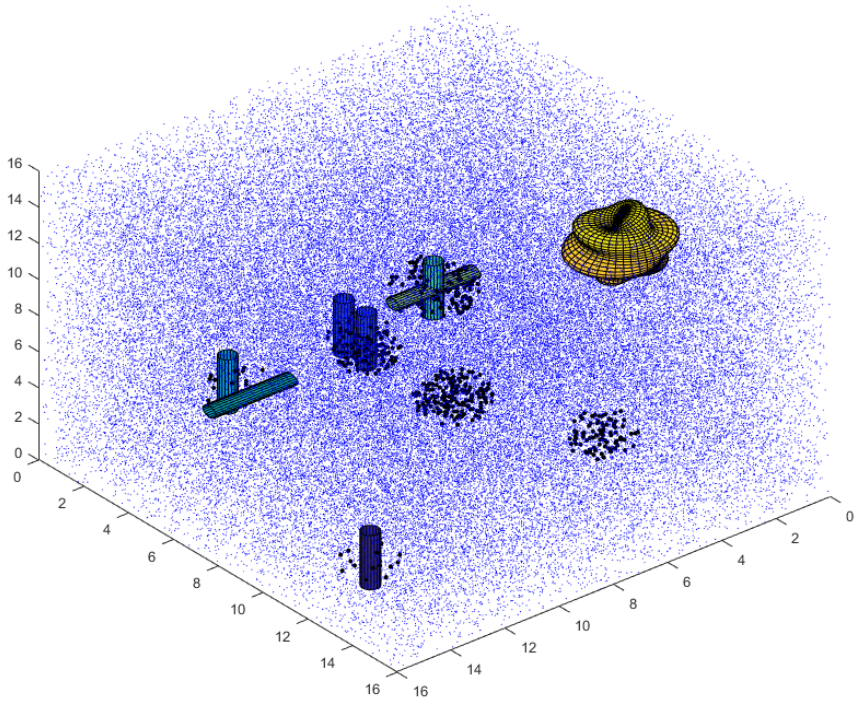


FIGURE 3.13: 3D manifold with cylinder, torus and reshaped sphere

Fig. 3.12(a), 3.12(b) and 3.13 show the different 3D data generated with our algorithm. The data were generated randomly with a hollow sphere, tunnels and a combination of two tunnels and a reshaped sphere. The idea behind reshaping a sphere is to force the network to understand the topology rather than the object. Then, every data sets was labelled with B_0, B_1 and B_2 numbers.

3.5 Experimental Protocol

Fig. 3.14 illustrates the network architecture used for 3D object topology detection in our study. Note that the 3D point cloud data share one input layer. As we used 3D point cloud data, we had three coordinates and converted the point cloud to $100 \times 100 \times 100$ voxel data. We used 1000000 total sample points to convert the point cloud to voxel data. The convolution operation operates with kernels of size $100 \times 100 \times 100$ and a stride of 2.

We used 3D CNN for our experiment. Max-pooling operates along the filter dimension. We used cross-entropy loss to train the network with Adam optimiser and a mini-batch size of 2 for 300 epochs. Every 20th epoch, the learning rate (initially set to 0.001) was halved.

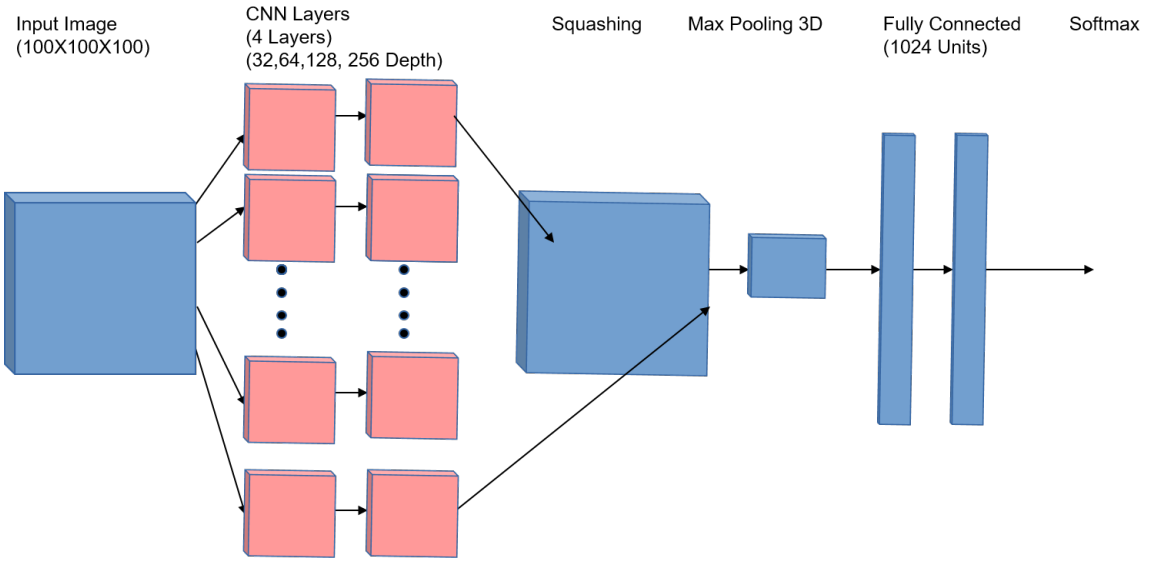


FIGURE 3.14: Example of the basic network architecture with double 3D convolutional layer. Four 3D convolutional layers, one max pooling layer, a fully connected layer and finally, a fully dense layer, where a final softmax is performed. Cross entropy loss was used and training was performed with Tensorflow with Adam optimiser.

TABLE 3.3: Convolutional Neural Network accuracy for 3D

Image Size	Sample Points	Accuracy
10×10×10	1000	29.4%
20×20×20	8000	31.8%
30×30×30	27000	47.6%
50×50×50	125000	74.9%
80×80×80	900000	88.8%
100×100×100	1000000	94.4%

3.5.1 Extension of the Model for 3D Image Data

The 3D convolutional neural network architectures can easily extract the features from volumetric images of the 3D MNIST data set (Irvin, 2017). These models can be robust and can be used for any perturbed data set. We attempted to implement the 3D convolutional model to recognise the number of holes present in the 3D image data.

3.5.2 Pre-processing of the Data

Each image was manually annotated for each hole present in the image with the help of a Hough transform based on the gradient field. The images were given the range of the 'sample radius' to detect all the holes. All points within the sample radius of a hole are sampled as a positive sample. An equal number of negative samples are randomly sampled outside the hole radius. A convolutional neural network is trained using the hole centres and negative samples. From Fig. 3.15 it can be seen that positive samples were correctly generated along with negative samples.

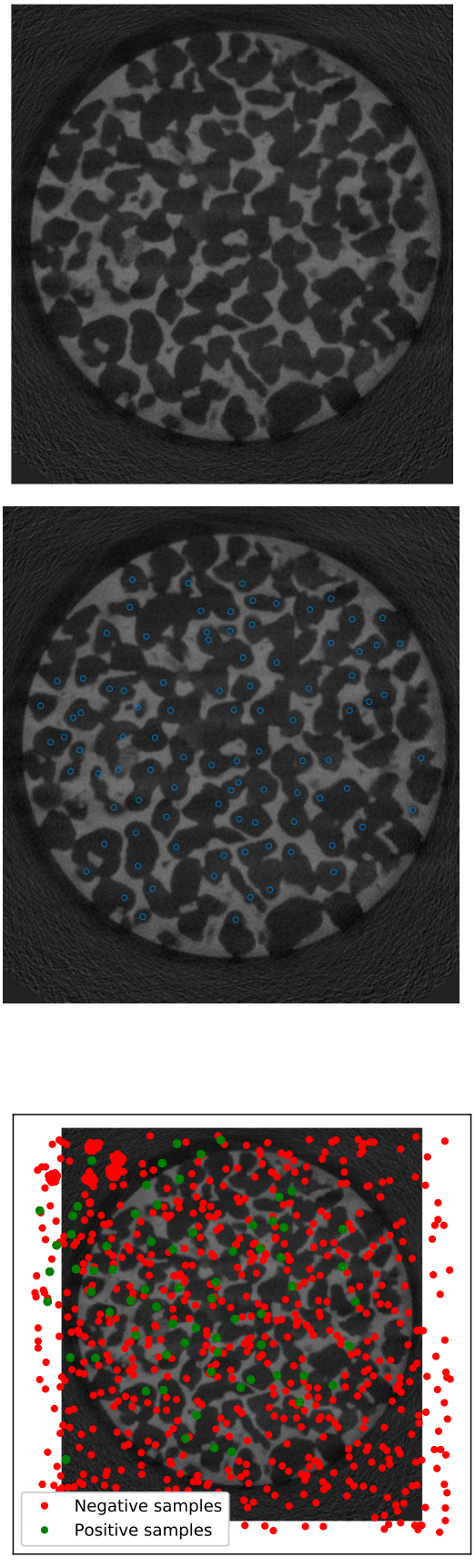


FIGURE 3.15: (a) Original image; (b) image with holes in the centres; (c) image with negative samples (red) and positive samples (green).

The following data was used as input for the CNN model for 3D. We obtained 70% accuracy using CNN when compared with the deterministic results of Javaplex. For both the cases we have used 2000 sample points. The synthetically generated 3D data sets were labelled individually by hands as Javaplex was not able process this data due to memory limitation.

3.6 Chapter Summary

This study has presented an approach towards learning topological properties of point cloud data. Our particular realisation of this idea, i.e., as an input layer to deep neural networks, not only enables us to learn the topology of point cloud data, but also to use this as additional (and potentially complementary) inputs to existing deep architectures. However, one drawback of the proposed approach is the understanding of PH beyond three dimensions; in fact, other strategies might be possible and better suited in certain situations. In summary, we argue that our experiments show substantial evidence that topological features of data can be beneficial in many learning tasks of random point cloud data, but do not necessarily replace existing methods.

Algorithm 3 3D Topological Structure

Input: Point cloud $coords2$, no of holes n.
Pick one point from the point cloud randomly as centre of hole c.
Initialize μ.
for $i = 1$ **to** n **do** Include 3D hole.
Pick one point from the point cloud as centre of hole c.
If c is first centre and distance from image $margin <= 2$
Add to the list of centre of holes $centers$.
$centers \leftarrow c$.
Else
Check for other points with distance as we don't want to intersect.
end for
for $j = 1$ **to** n **do**
Choose a random number between a 1 to 5.
switch a **do**
case 1 Include Torus object.
Calculate the distance from $c \in centers$.
If $dis_c < r$
Remove the point from the $coords2$.
case 2
Calculate the distance from $c \in centers$.
If $dis_c < r$
Remove the point from the $coords2$.
Take $rc = r/.5$ as cylinder radius and $h = r$ as height.
Put a cylinder at the centre c.
Include all the points inside the cylinder again to $coords2$.
case 3 Include 2-hole Torus
Calculate the distance from $c \in centers$.
If $dis_c < r$
Remove the point from the $coords2$.
Take $rc = r/.5$ as cylinder radius and $h = r$ as height.
Put two cylinder at the centres c and $c + 0.5$.
Include all the points inside the cylinder again to $coords2$.
case 4 Include crossed cylinder
Calculate the distance from $c \in centers$.
If $dis_c < r$
Remove the point from the $coords2$.
Take $rc = r/.5$ as cylinder radius and $h = r$ as height.
Put two cylinder at the centres c
Rotate one cylinder so that they crossed each other.
Include all the points inside the cylinder again to $coords2$.
case 5 Include Parallel cylinder
Calculate the distance from $c \in centers$.
If $dis_c < r$
Remove the point from the $coords2$.
Take $rc = r/.5$ as cylinder radius and $h = r$ as height.
Put two cylinder at the centres c and $c + 0.5$.
Rotate one cylinder so that they crossed each other.
Include all the points inside the cylinder again to $coords2$.
end for

Chapter 4

A Barrier Algorithm Approach for Optimisation Problems Over Non-Linear Manifolds

Optimisation over manifolds is a natural generalisation of smooth optimisation in $\mathbb{R}^n$. We take a new look at the optimisation techniques on manifolds. This chapter describes both the barrier method and the Newton backtracking method over manifolds to solve optimisation problems. There are many methods available to solve minimisation problems on Riemannian manifolds (Ji, 2007). These methods include the steepest gradient method, conjugate gradient method, Newton method and self-concordant functions. These methods can be used to solve problems over Riemannian manifolds with some modification. There are many past examples where the computation of these methods is not straightforward. For example, minimisation of a function over a sphere, where geodesic gradient and transformation are computed by non-linear functions and vector calculations. We implement some self-concordant functions, like the barrier method over manifolds, to solve specific optimisation problems.

In the case of a discrete optimisation problem, we need to choose the best solution from a set of points. This also includes the enumeration of the finite set of all feasible solutions, and comparison to obtain an optimal solution; this is rarely practical. However, discrete

problems can be solved using linear and non-linear programming methods. If the solution does not belong to the original data set, the approach rounds it to the closest solution in the set. Sometimes, with a favourable structure, the original problem formulation works well for the defined problem. We performed a similar treatment to the constraint as our constraint set is a manifold.

Our Contribution: To optimize a function in an abstract manifold is not convex optimization even if we consider Riemannian metric. If we could reformulate the constraint set with some convex properties it might have some empirical impact. To summarize, the key contributions of this chapter are the following :

i We use a logarithmic barrier method to manifold optimisation, for which we show how reformulation required based on the convexity, for starting point calculation.

ii Use of logarithmic barrier method with exact line search, which ensures small duality gap.

4.1 Background

As mentioned, computation over manifolds is not an easy task because calculation of geodesic distance is difficult. For this reason, there are several methods (Ji, 2007; Manton, 2002) which do not require consideration of any assumptions for the performance of optimisation on manifolds. In this chapter, we consider matrix manifold as our constraint set to optimise the linear function. This study presents novel algorithms that iteratively converge to a global minimum of a linear function $f(x)$ subject to the constraint. The results are obtained by explicating the constrained optimisation problem as an unconstrained one on a manifold using the logarithmic barrier function. The global optimum is assured by the assumption of the convex hull over the manifold. However, this assumption significantly reduces the complexity of the optimisation problem. We can reformulate the given problem with several convex properties; this might have a robust experimental impact. This provides the missing connection, and makes manifold optimisation not only match Matlab solver, but often outperform it.

Optimisation on manifolds is a fast-growing research topic in the field of non-linear optimisation (Boumal et al., 2014). The purpose is to provide efficient numerical algorithms to find (at least local) an optimiser for problems of the form

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq b_i,\ i = 1, \ldots, m. \\ & x \in M \end{aligned}$$

where the search space M is a smooth and differentiable manifold that can be endowed with a Riemannian structure.

For example, the **oblique manifold** $M = \{X \in R^{n \times m} : (X^T X) = 1_m\}$is a product of spheres. That is, $X \in M$ if each column of X has unit 2-norm in $\mathbb{R}^n$.

The (compact) **Stiefel manifold** is the Riemannian sub-manifold of orthonormal matrices,$M = \{X \in \mathbb{R}^{n \times m} : X^T X = \mathbb{I}_m\}$.

The **Grassmann manifold** $M = \{col(X) : X \in \mathbb{R}^{n \times m}\}$, where X is a full-rank matrix and $col(X)$ denotes the subspace spanned by its columns, is the set of subspaces of $\mathbb{R}^n$ of dimension m. Among other things, optimisation over the Grassmann manifold is useful in low-rank matrix completion.

The **special orthogonal group** $\mathrm{SO}(n) = \{X \in GL_n(\mathbb{R});\ X^T X = X X^T = I_n \text{ and } \det(X) = 1\}$ is the group of rotations, typically considered as a Riemannian sub-manifold of $\mathbb{R}^{n \times n}$. Optimisation problems involving rotation matrices occurring in robotics and computer vision belong to this manifold (Absil et al., 2009).

The set of **symmetric, positive semi definite, fixed-rank matrices** are also manifolds, $M = \{X \in \mathbb{R}^{n \times n} : X = X^T, rank(X) = k\}$. This space is tightly related to the space of Euclidean distance matrices X such that X_{ij} is the squared distance between two fixed points $x_i, x_j \in \mathbb{R}^k$ (Boumal et al., 2014).

Optimisation on manifolds is a natural candidate for the design of non-linear estimation algorithms. By operating directly on the low-dimensional search space the computational

costs can be kept proportionate to the complexity of the sought object. Riemannian optimisation generalises popular tools from continuous, unconstrained optimisation, such as gradient descent, Newton methods, trust-region methods, etc. However, from the classical Euclidean case to the field of Riemannian search spaces, information is lost in the process of convergence of the method. The same regularity conditions and global and local convergence results are established in a mature theory laid out by (Absil et al., 2010). Apparently, non-convex optimisation problems are still difficult to solve, and the relevance of the reached optimisers often depends on the quality of the initial starting point of the solution.

Traditional methods, like steepest descent, are not only robust in solving optimisation problems, but they almost always successfully converge to a local minimum. However, these algorithms converge slowly (Manton, 2002). This method was first introduced to manifolds by Luenberger (1972, 1973) and Gabay (1982). In the early nineties, this method was performed on problems in systems and control theory by Brockett (1993), Helmke and Moore (2012), Smith (2013) and Mahony (1994).

Compared to the steepest descent method, the Newton method has a quadratic convergence rate. In 1982, Gabay (1982) extended the Newton method to a Riemannian sub-manifold of $\mathbb{R}^n$ by updating iterations along the geodesic. Other independent work to extend the Newton method on Riemannian manifolds has been performed by Smith (1994) and Mahony (1994) restricted to the compact Lie group, and by Udriste (1994), limited to convex optimisation problems on Riemannian manifolds. Edelman et al. (1998) also introduced a Newton method for the optimisation of orthogonality constraints - Stiefel and Grassmann manifolds. Dedieu et al. (2003) also studied the Newton method to find the zero of a vector field on general Riemannian manifolds.

Even though Newton's method has a faster quadratic convergence rate, it requires computation of the inverse of a symmetric matrix, called the Hessian, consisting of the second order local information of the cost function. Therefore, it increases the computational cost.

When considering large-scale optimisation problems with sparse Hessian matrices, quasi-Newton methods encounter difficulties. The conjugate gradient method is used for solving

such problems as it avoids computing the inverse of the Hessian (Jiang et al., 2007).

The barrier method approach with exact line search is used for optimisation problems over matrix manifolds. First, general properties of the barrier method in Euclidean space are derived. Based on this, a logarithmic barrier method is proposed for optimisation of functions; this guarantees that the solution falls within any given small neighbourhood of the optimal solution in a finite number of steps. This method only requires convexification for the starting parameter. We introduced a different invocation of the barrier method to solve optimisation problem over manifolds. An example evaluation of an optimisation problem is also given to illustrate the proposed concept and algorithm. The first algorithm based on the logarithmic barrier method was coupled with Armijo's condition for choosing the step size at each iteration (Polak, 2012). According to Polak, the steepest descent method is nearly non-implementable for the calculation of step-size. The second type of algorithm derived in this chapter is based on the traditional Newton algorithm. The novel component of our study is that the nearby neighbourhood to which the barrier method or Newton method is applied changes at every cycle. There are several advantages of applying both the barrier method and Newton calculations in our method. The logarithmic barrier method guarantees that the solution should lie within the constraint region. Backtracking sort calculations coupled with Armijo's condition quite often manage to converge to a local minimum. Newton method calculations, using second-order derivatives, can accomplish quadratic convergence. This quick convergence produces several disadvantages to the whole algorithm. There will be no guarantee that the algorithm will converge to a minimum, rather, it can converge to a closest critical point, which can be a local minimum or saddle point (Manton, 2002). However, our assumption of the convex hull over the manifold ruled out this possibility of a local minimum. The operation of numerical Hessian and gradient with Armijo's condition provides a decrease in the objective function, which is expected to achieve the final solution.

4.2 Problem Set-up

The main motivation for our method comes from the need to solve a continuous, unconstrained optimisation problem. The steepest descent and gradient descent methods are

well known available algorithms. M can be considered as a collection of points without any specific manifold structure, which turns M into a topological set. Then, we can apply the notion of neighbours, which can determine if the real-valued function on M is smooth or not. The real-valued function f on the constraint set M defines the goal of the optimisation problem, and is termed the objective function (Absil et al., 2010). Computing the minimum value of f is our goal.

Problem.
Given: a manifold $\mathcal{M}$ and a function $f : \mathcal{M} \to \mathbb{R}$.
Solution : a feasible x^* of $\mathcal{M}$ such that there is a small neighbourhood B of x^* in $\mathcal{M}$ where $f(x^*) \leq f(x)$ for all $x \in B$.
Our algorithm solves this problem over manifold $\mathcal{M}$ iteratively. Given a starting point $x^0 \in \mathcal{M}$, our algorithm produces a sequence of $(x^k)_{k\geq 0}$ in $\mathcal{M}$ which converges to x^*. The convex hull assumption over $\mathcal{M}$ assures that x^* does not give us a local minimiser.

According to Absil et al. (2009), the simplest approach to optimise a differentiable function is to continuously search for a point in the descent direction where the gradient vanishes. These points where $\bigtriangledown f = 0$ are called stationary points or critical points of f. The closest numerical analogy is the class of optimisation methods that uses line-search procedures and which can calculate gradient in the descent direction. However, on a manifold, there is the notion of moving in the direction of a tangent vector, while staying on the manifold. Conceptually, a retraction R at x, denoted by R_x, is a mapping from tangent space $T_x\mathcal{M}$ to $\mathcal{M}$ with a local rigidity condition that preserves the gradient at x. According to LaValle (2006), the tangent space $T_x\mathcal{M}$ at a point p on an manifold $\mathcal{M}$ is an n-dimensional hyperplane in $\mathbb{R}^m$ that best approximates $\mathcal{M}$ around p, when the hyperplane origin is translated to p.

Definition 1 (retraction with projection). A retraction on a manifold $\mathcal{M}$ is a smooth mapping R from the tangent bundle $T\mathcal{M}$ onto $\mathcal{M}$ with the following properties. For all x in $\mathcal{M}$, let R_x denote the restriction of R to $T_x\mathcal{M}$. Then,

- $R_x(0) = x$, where 0 is the zero element of $T_x\mathcal{M}$.
- The distance $d_{p\to x} = \min\|\mathbf{p} - \mathbf{x}\|$ (Absil and Malick, 2012).

- The angles between vectors are preserved, i.e., $\forall x, y \in \mathcal{T}_x\mathcal{M}, \langle Tran_p(t_x), Tran_p(t_y)\rangle = \langle x, y\rangle$.

Proof: for a manifold, any two points can be joined by a piecewise smooth curve segment. Thus they admit a natural differentiable structure which can be defined in Riemannian metric (Absil et al., 2009). The collection of all points and their respective tangent spaces is called a tangent bundle $T\mathcal{M}$, which is geodesically complete. Since $\mathcal{M}$ is geodesically complete, given a point $p \in T_x\mathcal{M}$, there exists a unique $x \in \mathcal{M}$, such that the distance $d(p, x)$ is minimal.

Since $\mathcal{M}$ is smooth, one of its properties is that, in the case of parallel transport of point x to y, the angles between vectors are preserved (Ji, 2007).

4.3 Manifolds

In this study, along with the SR and HR data set which has been explained in Chapter 2, a torus data set was also used. A torus data set is created by the equation below:

$$y = \begin{bmatrix} (2 + \cos(x_1)) \cdot \cos(x_2) \\ (2 + \cos(x_1)) \cdot \sin(x_2) \\ sin(x_1) \end{bmatrix} \tag{4.1}$$

where $0 \leq x_1 \leq 2\pi$ and $0 \leq x_2 \leq 2\pi$. With this parametric equation, the torus surface is isotropic in the plane spanned by the coordinates y_1 and y_2.

4.4 Starting Point Calculation

For any optimisation problem, the choice of a starting point plays an important role. Interior methods for non-linear and quadratic programming perform poorly, or in some cases even fail, if we choose an incorrect starting point (Gertz et al., 2004). To overcome this problem, heuristics have been used to obtain a starting point (Mehrotra, 1992). The

starting solution x^0 is either defined by the user or set to $0(x^0 = 0)$. In our approach, we determined our starting point by:

Algorithm 4 Starting point x^0

Input: data $\mathcal{M}$, size m.

Get manifold data from Eq. 2.1, say $\mathcal{M}$.

Calculate the convex hull $\mathcal{C} = \left\{ \sum_{i=1}^{N} \lambda_i P_i \right\}$

where $\lambda_i \geq 0,\ \forall i\ \sum_{i=1}^{N} \lambda_i = 1$ *and* $P_i \in \mathcal{M}$.

Get the Chebyshev centre of the convex hull $\mathcal{C}$, say x^c.

Find closest point x^{c*} by retraction of x^c to $\mathcal{M}$ (Absil et al., 2009; Boumal, 2014).

$x^0 \leftarrow x^{c*}$.

After we obtain the initial point, we begin to evaluate the gradient and Hessian of the given function f (Ji et al., 2008).

4.5 Optimisation over Manifolds

In Euclidean optimisation, the main goal is to find a descent direction and then to perform a line-search to search minimum and ensure convergence. On a manifold, the descent direction is calculated in tangent space. Given a descent direction, we use the exact line search method on the tangent space and retract the final point to the manifold. Descent direction over manifold is simply the direction on the tangent space. The notion of tangent vector at a point $x \in M$ can be defined when M is a sub-manifold of a Euclidean space η (Absil et al., 2010; Hosseini and Sra, 2015). As we make the assumption of a convex hull, $\mathcal{M}$ automatically becomes Euclidean space where we can perform gradient and Hessian. The set of all tangent vectors at x is defined as tangent space to $\mathcal{M}$ at x denoted by $\mathcal{T}_x\mathcal{M}$. A point $p \in \mathcal{C}$ can be transported to the tangent vector space $\mathcal{T}_x\mathcal{M}$ on the manifold and expressed as $Tran_p(t_x)$ from Definition 1. Now, the first, second and third order derivatives of f are defined as follows:

$$f_x^{'}(p) = \bigtriangledown f(Tran_p(t_X)) \tag{4.2}$$

$$f_x^{''}(p) = \bigtriangledown^2 f(Tran_p(t_X)) \tag{4.3}$$

$$f_x^{'''}(p) = \bigtriangledown^3 f(Tran_p(t_X)) \tag{4.4}$$

The gradient and Hessian of f at $p \in \mathcal{C}$, are denoted by $grad_p f$ and $hess_p f$ respectively and given by:

$$grad_p f = f_x^{'}(p) \tag{4.5}$$

$$hess_p f = f_x^{''}(p),\ x \in \mathcal{T}_x\mathcal{M} \tag{4.6}$$

4.5.1 Barrier Function

Given $c \in \mathbb{R}^n$ and $\alpha_i \in \mathbb{R},\ i = 1, \ldots, m$, we consider the following convex programming problem

$$\begin{aligned} \underset{x}{\text{minimise}} \quad & c^T x \\ \text{subject to} \quad & f_i(x) \leq \alpha_i,\ i = 1, \ldots, m. \end{aligned}$$

where all functions $f_i(x), i = 1, \ldots, m.$ are convex. To apply the interior point method to this problem, we must construct the self-concordant barrier for the domain. Let us assume that there exists standard self-concordant barriers $F_i(x)$ for the inequality constraints $f_i(x) \leq \alpha_i$. Then, the resulting barrier function for this is

$$F(x) = \sum_{i=1}^{m} f_i(x) \tag{4.7}$$

Given a sequence $\{\mu_t\}$ such that $\{\mu_t\} > 0$, we minimise the new function

$$f(x, \mu_t) = \frac{1}{\mu_t} c^T x + F(x) \tag{4.8}$$

in sequence and use the solution of the current minimisation problem as the initial guess for the next optimisation (Ji, 2007). As μ_t goes to zero in the limit, we obtain the minimum of the original problem. In fact, according to Boyd and Vandenberghe (2004), it is not necessary to obtain the exact minimum of the cost function $f(x, \mu_t)$ for every given μ_t. A common method used in practice is to perform one step or several steps of Newton method and then choose a different μ.

Gradient-based methods are preferred for large scale problems given that they avoid computing the inverse of the Hessian matrix. Since $c^T x$ is linear and $F(x)$ is standard self-concordant, $f(x, \mu_t)$ is still standard self-concordant for all $\mu_t > 0$ (Nesterov, 2013). As we take the convex hull assumption over the manifold, we can use the barrier self-concordant function to optimise the objective function.

The barrier function we used in our algorithm is

$$F(x) = -log\{f_i(x) - \alpha_i\}. \tag{4.9}$$

We used the logarithmic barrier because as $f_i(x) - \alpha_i$ tends to 0, $log\{f_i(x) - \alpha_i\}$ goes to negative infinity. This introduces a gradient to the function being optimised which favours less extreme values of x, while having relatively low impact on the function away from these extremes. Another advantage of logarithmic barrier functions is that they are less computationally expensive, depending on the function being optimised (Burer et al., 2003).

Algorithm 5 Solver x^*

Input: Function $f : c^T x$, constraint $\mathcal{M}$, size m

x^0 = call procedure starting point with $\mathcal{M}$

Initialize μ.

for $i = 1$ **to** MAX_ITERATION **do**

Make the problem unconstrained by using Eq. 4.8.

Calculate the gradient from Eq. 4.3. And $-f_x^{'}(p)$ is the descent direction.

repeat

Initialize the step length $s \leftarrow 1$

Find a feasible point x^k and perform backtracking line search.

Transport x^k to manifold $\mathcal{M}$ by $Trans_{x^k}(t_y)$, where $y \in \mathcal{M}$ using Definition 1.

Update step size s. This step is the Armijo-Goldstein condition.

until min $f_i(x) < 0$

Update μ until $f(x^*, \mu_t) \leq f(x^k, \mu_t)$

end for

4.5.2 Preliminaries of Primal-dual Approach

Primal-dual scheme is mainly used to obtain a feasible integral solution for an ILP by solving its LP relaxation. For a linear program, there always exists a dual linear program. Further, the dual of the dual linear program is the original linear program. The original LP is referred to as the *primal LP*. In order to use this approach, the primal LP (a minimisation problem, in this case) has to be in the following standard form:

$$\begin{aligned} \text{Minimise} \quad & \mathbf{c}^T\mathbf{x} \quad \text{subject to} \\ A\mathbf{x} \quad & \geq \quad \mathbf{b}; \quad \text{N constraints} \\ x_k \quad & \geq \quad 0; \quad \forall k \in \{1, 2, \ldots, M\}\,. \end{aligned} \tag{4.10}$$

From the primal LP, the corresponding *dual LP* (which becomes a maximisation problem) can be written as follows:

$$\begin{aligned} \text{Maximise} \quad & \mathbf{b}^T\mathbf{y} \quad \text{subject to} \\ A^T\mathbf{y} \quad & \leq \quad \mathbf{c}; \quad \text{M constraints} \\ y_l \quad & \geq \quad 0; \quad \forall l \in \{1, 2, \ldots, N\}\,. \end{aligned} \tag{4.11}$$

Observe that, each constraint in the primal problem has a corresponding variable in the dual problem, and vice-versa. More details about relations between primal and dual problems can be found in Vazirani (2013).

We now list some of the well-known results from Vazirani (2013) that will be used in the subsequent part of this section to prove the optimality gap.

Theorem 4.1. ***LP Duality Theorem.***
The primal LP has a finite optimum if, and only if, the corresponding dual LP has a finite optimum. Moreover, if $\mathbf{x}^* = [\ x_1^* \quad x_2^* \quad \cdots \quad x_M^*\ ]^T$ *and* $\mathbf{y}^* = [\ y_1^* \quad y_2^* \quad \cdots \quad y_N^*\ ]^T$ *are optimal solutions for the primal LP and the dual LP respectively, then it holds that*

$$\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*.$$

Theorem 4.2. ***Weak Duality Theorem.***

If $\mathbf{x} = [\ x_1 \quad x_2 \quad \cdots \quad x_M\ ]^T$ *and* $\mathbf{y} = [\ y_1 \quad y_2 \quad \cdots \quad y_N\ ]^T$ *are feasible solutions of the primal LP and the dual LP, respectively, then it holds that*

$$\mathbf{c}^T\mathbf{x} \geq \mathbf{b}^T\mathbf{y}.$$

Theorem 4.3. ***Complementary Slackness Condition:*** *Let* $\mathbf{x}$ *and* $\mathbf{y}$ *be the primal and the dual LP feasible solutions respectively. Then* $\mathbf{x}$ *and* $\mathbf{y}$ *are both optimal if, and only if, both of the following conditions hold*

Primal complementary slackness condition:

$\forall k \in \{1, 2, \ldots, M\}$*: either* $x_k = 0$ *or* $\sum_{l=1}^{N} a_{kl}y_l = \mathbf{c}_k$

Dual complementary slackness condition:

$\forall l \in \{1, 2, \ldots, N\}$*: either* $y_k = 0$ *or* $\sum_{k=1}^{M} a_{kl}x_k = \mathbf{b}_l$

where a_{kl} *is an element of* A *in* k^{th} *row and* l^{th} *column.*

For the problem under consideration, we need to find a feasible solution for an linear program as we make the assumption of convex hull:

$$\begin{array}{llll} \text{Minimise} & \mathbf{c}^T\mathbf{x} & \text{subject to} & \\ & A\mathbf{x} \quad \geq \quad \mathbf{b}; & \text{N constraints} & \\ & x_k \quad \in \quad \mathbb{R}; & \forall k \in \{1, 2, \ldots, M\} & \end{array} \tag{4.12}$$

In such cases, the primal-dual schema is used with linear relaxation (shown in Eq. 4.10) for the ILP (shown in Eq. 4.12). As we need integer values for $\mathbf{x}$, Theorem 4.1 to Theorem 4.3 cannot be used directly. In such cases, since we require only a feasible solution, the complementary slackness conditions (shown in Theorem 4.3) can be relaxed. Approximation algorithms that are designed using the primal-dual schema commonly ensure that one of the conditions shown in Theorem 4.3 holds and the other is suitably relaxed. In the following theorem, two different situations are captured — one for relaxing each condition. However, if primal conditions can be ensured, then set $\alpha = 1$, and if dual conditions can be ensured, then set $\beta = 1$ (Vazirani, 2013).

Theorem 4.4. *Let* $\mathbf{x}$ *and* $\mathbf{y}$ *be the feasible solutions for primal and dual LPs respectively, that satisfy the relaxed complementary slackness conditions stated above. Then it holds:*

$$\mathbf{c}^T\mathbf{x} \leq (\alpha\beta)\mathbf{b}^T\mathbf{y} \tag{4.13}$$

The optimality gap of any optimization problem can be defied as :

$$gap = c^T x - b^T y \tag{4.14}$$

4.6 Experiments

We performed numerous experiments to examine the correctness of our method. Every test was performed 20 times for each data set, as the dataset was generated randomly. After that, for each optimisation problem, we used 30 *Max Iteration* to perform line search during the barrier method over the manifold. The line search can be stopped earlier if the gradient becomes zero or very close to zero (tolerance). In our experiment, we set tolerance to $1e^{-10}$.

Below we report the accurate comparison of our method on Swiss roll data. We generated two different types of Swiss roll data and a torus from Eq. 2.1 and 4.1. These are referred to as standard Swiss roll, heated Swiss roll and torus. First, we minimised a linear function $c^T x$ over these three manifolds, where c is the coefficient of the $\vec{x}$.

$$f(x) : c^T x \rightarrow c_1x_1 + c_2x_2 + ... + c_nx_n \tag{4.15}$$

Next, to verify our algorithm, we attempted to minimise a paraboloid function over the torus manifold. We generated the non-linear parabolic function in such a way that the function passed by the origin. Thus, we know that the minimum point of the function is at the origin. However, the origin in our torus manifold does not include the origin in the sample point set. According to our algorithm, the optimal solution comes to the inner surface of the torus as expected. In this experiment, a 3-dimensional parabolic function

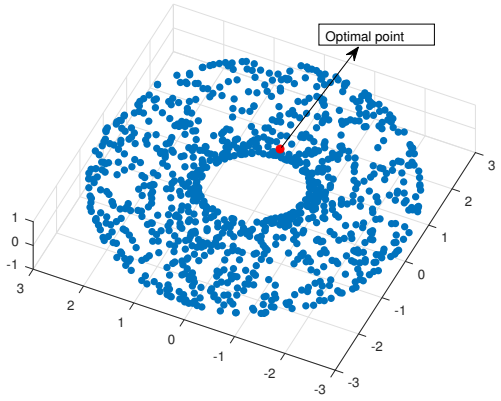


FIGURE 4.1: Optimal point over manifold for non-linear function 4.16

TABLE 4.1: Optimal result for paraboloid function over various sample torus

POINTS	OUR APPROACH	SOLVER	TIME(SEC)
50	0.440	0.440	15.5
100	0.442	0.442	15.77
200	0.449	0.449	15.26
300	0.449	0.449	15.40
400	0.449	0.452	15.45
500	0.456	NIL	15.52
700	0.456	NIL	15.85
800	0.456	NIL	15.99
900	0.456	NIL	16.21
1000	0.456	NIL	16.29

was used to verify our algorithm, which can be given by:

$$z = c_1 {x_1}^2 + c_2 {x_2}^2 \tag{4.16}$$

where x_1 and x_2 are uniformly distributed over$[-1, +1]$. Table 4.1 gives a clear picture of our experiment. We chose $c_1 = c_2 = 0.1$ for Eq. 4.16. Then we calculated the optimum value for the function z. The dataset used for this experiment was Torus which is represented in Fig. 4.2. However, after 500 sample points, Matlab solver stopped giving results saying the solution was unbound. Fig. 4.1 shows the actual optimal point if we try to optimize the non-linear function (4.16) over a torus.

Table 4.2, Table 4.3 and Table 4.4 present the comparison between our method and the

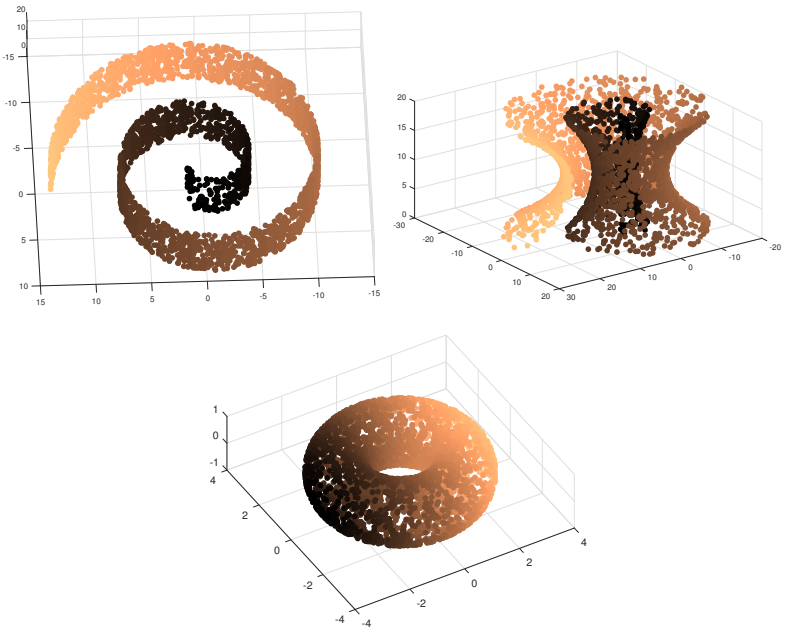


FIGURE 4.2: Two data sets of increasing complexity: (a) SR; (b) HR; (c) torus; (d) sphere

TABLE 4.2: Optimal result for linear function over various sample SR

POINTS	OUR APPROACH	SOLVER	TIME(SEC)
50	-6.54	-6.65	15.2
100	-6.27	-6.65	15.22
200	-6.13	-6.78	12.52
300	-6.43	-6.78	15.40
400	-6.69	-6.84	15.45
500	-6.56	-6.86	15.52
700	-6.58	-6.86	15.85
800	-6.74	-6.86	15.99
900	-6.74	-6.96	16.21
1000	-6.74	-6.96	16.29

standard Matlab solver. We compared our results to a various number of sampled points starting from 10. We did the same for the Matlab solver. For SR, the accuracy of our method with the standard solver was in the range of 98%. But in the case of HR, the accuracy varied from 90-100% for various sampled points, and for torus, the accuracy was around 85%. However, we can see from the Table 4.1 that as the number of sample points increases, Matlab started to give no solution. From Table 4.4 and Table 4.3, it can be seen that Matlab does not give any result for 700 and 800 sample points. Fig. 4.2 shows all variety of data set we have used.

Table 4.3: Optimal result for linear function over various sample HR

Points	Our Approach	Solver	Time(Sec)
50	-4.32	-6.34	12.36
100	-8.34	-8.30	15.22
200	-8.34	-8.34	15.23
300	-8.34	-8.62	15.22
400	-8.62	-8.62	15.22
500	-8.62	-8.62	15.22
600	-8.62	-8.62	15.22
700	-8.62	Nil	15.22
800	-8.62	Nil	15.22
900	-8.62	-8.62	15.22
1000	-8.62	-8.62	15.22

Table 4.4: Optimal result for linear function over various sample Torus

Points	Our Approach	Solver	Accuracy
50	-6.32	-6.32	15.07
100	-6.32	-6.32	15.31
200	-6.33	-6.33	15.12
300	-6.33	-6.33	15.12
400	-6.32	-6.33	14.74
500	-6.31	-6.34	15.24
600	-6.06	-6.34	15.23
700	-6.31	Nil	15.19
800	-6.52	Nil	15.10
900	-6.52	-6.52	15.39
1000	-6.52	Nil	15.34

From Table 4.2, Table 4.3 and Table 4.4 we can clearly see that, for each and every sampled manifold, we calculated the optimal solution for Eq. 4.15.

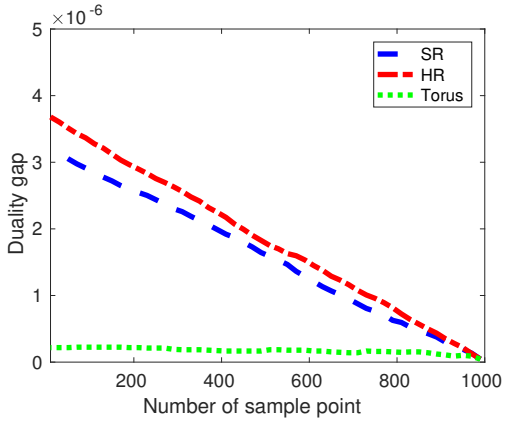


FIGURE 4.3: Optimality gap for each sampled manifold for given function $c^T x$.

From Fig. 4.3 we can see that, for all the sampled manifolds, the optimality gap for the objective linear function is between 10^{-6} to 10^{-8}. We ran several experiments for the same linear function and took the average of the optimality gap for every sampled manifold. From the graph it can be seen that as we increase the number of samples in the data set, we reduce the optimality gap for our optimisation problem, and at 1000 sample points it is very close to zero. This phenomenon indicates that if we increase the number of sample points, it behaves like a curved surface, and as explained in section 2, it can join as a piecewise smooth curve segment. We then obtain more accurate results.

We are achieving this optimality gap, which is very close to the actual optimal solution, in substantially less time.

From Fig. 4.4, we can say that the average time needed to find each optimal value for a given linear function under constraint $\mathbb{M}$ is 15 to 16 seconds. The time taken by each experiment is considered less for the optimisation problem. The bar graph 4.4 shows the time analysis for each data set we used as a constraint. This shows a randomly selected matrix manifold and the time taken to reach the optimal solution.

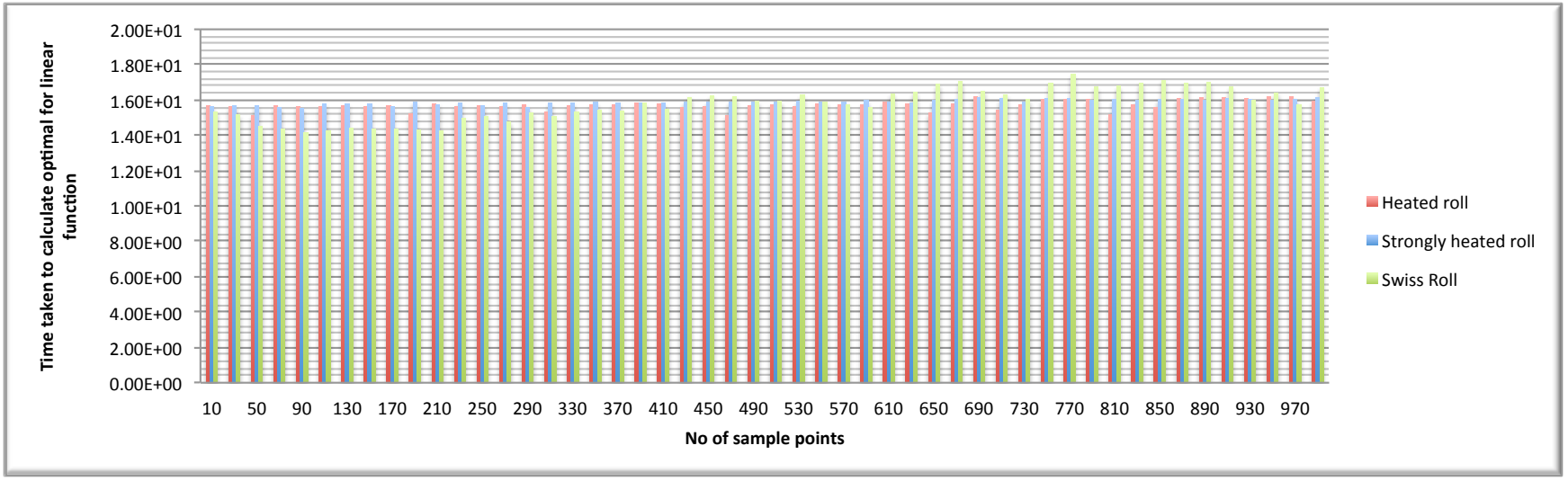


FIGURE 4.4: Comparison of time taken for different data sets.

4.7 Chapter Summary

In summary, optimisation on manifolds depends on the exploitation of tools of differential geometry to build optimisation schemes on manifolds, and then the conversion of these abstract geometric algorithms into practical numerical methods. This can be applied to problems that can be rephrased, such as optimising a differentiable function over a manifold. We close by pointing out that optimisation of real-valued functions on manifolds, as formulated in Problem 1, is not the only place where numerical optimisation and differential geometry meet. We have achieved comparatively good results with the synthetic data set. Other examples can be found in Absil (2009); Dedieu et al. (2005); Nesterov and Nemirovski (2008); Zhao (2010); these studies all describe the same connection between geometry and optimisation. Three significant but easy optimisation problems are given to illustrate the efficiency of the proposed concept and algorithm.

Chapter 5

Improving Operational Performance in Service Delivery Organisations Using a Meta-heuristic Task Allocation Algorithm

Efficient allocation of tasks to employees is crucial in SDOs. It can facilitate the achievement of SLAs, utilises the employees well and improves operational performance. Task allocation is a challenging problem that addresses the inter-dynamics of tasks and employees, and requires consideration of factors such as diversity, utilisation, skills and fairness. The combination of task deadlines and associated SLA requirements adds another dimension to the complexity of the data and problem. In this Chapter, we propose a Tabu search algorithm for efficient task allocation. The algorithm is aware of employee utilisation, productivity and fairness. We evaluate the proposed algorithm using real-world transaction processing data from a large SDO. The results show that the proposed approach can reduce deadline misses substantially, in comparison to the organisation's current approach.

In this Chapter, we address the problem of efficiently allocating tasks to resources in a SDO, with the objective of allocating tasks in a real time environment. We attempt to obtain final allocations within a certain optimality gap. We also consider several other constraints, like that the number of baseline violations is within a specified limit (KPI/SLA-aware), the assigned workload is fair among the resources, that utilisation of each resource should be within the specified lower and upper bounds to ensure that all the resources are well utilised (utilisation-aware) and productivity of each resource is within a specified upper limit, as described in (Mulla et al., 2016).

Provided an ILP-based solution for the problem, ILP-based solutions take a substantially increased amount of time as the number of resources and number of tasks increases. For example, the ILP-based approach does not provide any feasible solution (within two hours) for a scenario where we have 22 resources and 2000 tasks. As SDOs need an allocation matrix for the resources in (near-)real time, an ILP-based solution cannot be deployed in practice. Therefore, we propose a meta-heuristic-based approach in this paper. It involves two steps:

1. We convert the problem to a meta-heuristic-based problem in which Tabu search algorithms have been proven to be effective and fast.
2. We propose a Tabu-search (TS) based algorithm.

This chapter is organized as follows. Section 5.2 presents the notations and the system model. Our task allocation approach based on Tabu search is presented in Section 5.4. Section 5.5 presents the experimental set up and discusses the efficacy of the proposed solution when applied to real-world data from a large SDO. Finally, Section 5.6 summarize the chapter.

5.1 Related Work

The problem of task allocation has been well studied (Cheng and Sin, 1990; Coffman Jr et al., 1984; Dowsland and Dowsland, 1992; Graham et al., 1979; Lawler et al., 1982;

Mokotoff, 2001; Price, 1982; Wiese et al., 2013) with a focus on assigning tasks to machines (non-human resources). These studies cannot be applied to the problem under consideration since they cannot handle some of the human related aspects, such as productivity and utilisation of employees.

Other studies (Ernst et al., 2004; Krishnamoorthy and Ernst, 2001; Lapègue et al., 2014; Prot et al., 2015; Smet et al., 2013) have specifically looked into the problem of allocation of tasks to employees. Ernst et al. (2004) studied the problem of assigning employees to shifts based on an estimate of the skills required for the jobs in those shifts. Krishnamoorthy and Ernst (2001) discussed a class of personnel task scheduling problems (that arise in restoring applications), mostly related to scheduling of employees in different shifts and assigning different tasks to a set of shifts. Smet et al. (2013) proposed a hybrid-heuristic approach in which the objective was to assign the tasks such that a minimum number of resources was used, a problem clearly different from ours. This approach considers skills of employees and demand while recommending reallocation. The allocation is performed considering the availability of employees, skills of employees, skills required to perform a task and start and end date for the task. Other studies (Lapègue et al., 2014; Prot et al., 2015) examined the problem with a fairness objective while ensuring the regulatory requirements. These studies did not consider the effect of allocation on operational KPIs, nor did they consider key human factors such as employee productivity and utilisation. Thus, none of these studies can be directly applied to the problem under consideration in this work. Thus, none of these works can be directly applied to the problem under consideration in this work.

Several mechanisms have been proposed in studies by Fields et al. (1992); Fletcher et al. (2012); Hamadi and Quimper (2007); Mitra et al. (2001); Powell et al. (1999); Su (2002) who performed allocation of tasks to employees using an allocation method that was skill- and fairness-aware. Many of these techniques (Fields et al., 1992; Mitra et al., 2001; Powell et al., 1999; Su, 2002) were not utilisation-, productivity- and SLA-aware (e.g., minimising the number and magnitude of baseline time violations). In Fletcher et al. (2012), an engine was configured to determine assignment of a resource to a task considering that the task and the resource profile proposed. This took into account resource utilisation. However, it was not productivity-aware, and since there was no deadline for

tasks, its objective was different than ours. In Hamadi and Quimper (2007), a constraint programming based allocation technique was proposed. It performed skill-based allocation and considered fairness while trying to minimise the cost. However, the allocation strategy was not productivity-aware or utilisation-aware, and did not focus on minimising deadline violations.

In most SDOs, in order to effectively process the incoming transactions, one needs to efficiently allocate these transactions to employees on a daily basis. Several parameters need to be considered for such an allocation exercise, some of which are the skills required to perform a transaction, the fairness in terms of workload allocation to employees, the committed SLA in terms of the number and magnitude of baseline violations, the utilisation of employees and the performance of employees, among others. Although there exists several available allocation techniques on the market, most of them try to solve the problem by considering just one or two of the following aspects, and none of them address the problem holistically by considering all of these aspects.

Skill based-allocation: To process a transaction, an employee needs to have certain pre-defined skills. These skills may differ depending on the type of transaction. Allocating a transaction to an employee who does not possess the required skills may result in SLA misses, deterioration in performance of employees and employee dissatisfaction. Generally, different types of transactions that an organisation needs to process and the respective skills required to process the same are known in advance. Therefore, when performing the allocation, it must be ensured that the transactions are assigned to employees with the right skills.

Number and magnitude of baseline violations: Typically, each transaction type has an estimated time within which the processing of a transaction of that type needs to be completed (generally referred to as the baseline time of the transaction type). Ideally, each transaction must be finished within its baseline time. However, in reality it may not be possible for several reasons, including high workload, improper transaction assignment, etc. Therefore, one of the ways in which SLAs are set is to put an upper limit on the number of transactions that can violate the baselines and to minimise the magnitude of such baseline violations as much as possible.

Fairness: The allocation scheme must ensure that the transactions are assigned fairly among the available employees. One way to achieve this would be to make sure that, while assigning the transactions, the workload of each employee can only vary within a specified range from the average workload of all employees. In other words, we can ensure that no employee has a workload beyond a certain limit from the average workload.

Utilisation of employees: Often, it can happen that although the allocation is fair, employees in the group can be under- or over-utilized, which is not a desired scenario. It is crucial to utilise these resources efficiently to reduce cost, thereby increasing profit. Any task allocation scheme needs to ensure that the employees are utilised well, i.e., enough work is allocated to each employee to keep them occupied most of the time. At the same time, we need to make sure that employees are not overloaded with too much work.

For example, it is possible for the allocated work to keep the employees busy for only 50% of their available time, or sometimes when there are too many transactions to handle, an over allocation of work which can keep employees busy say 150% of their available time, implying that employees are overloaded with work, and they may have to work beyond their available hours to complete the work. The first scenario indicates that the group is overstaffed and the second scenario indicates that the group is understaffed. A good allocation scheme should try to avoid such scenarios and one way it can do so is by posing explicit lower and upper bounds on utilisation of each employee.

Incentive-aware allocation: In an inferior allocation scheme, it is possible that certain employees who are performing extremely well with respect to certain transaction types (say, an employee is finishing these transactions well within their respective baselines) might be repeatedly allotted transactions of the same type. This may give an unfair advantage for these employees to be more and more productive compared to their peers, in terms of the number of transactions that they finish in a given time. Typically, incentive calculation in organisations is dependent on productivity; such biased allocation can give unfair advantage to certain employees by allowing them to claim a major share of the team's incentive, leaving almost nothing for others. A good allocation scheme should address this issue by giving equal opportunity for all employees to be productive by providing unbiased allocation of tasks.

Dynamic re-allocation: The performance of employees may vary depending on the time of the day. For example, performance of some employees may improve as the day progresses, while it may reduce for others. So, with one time allocation, say at the beginning of the day, it may not be possible to keep some of the above discussed metrics in check. Therefore, a good allocation scheme needs to dynamically re-allocate transactions depending on the real-time performance of employees.

Table 5.1, provides an integer programming model to obtain the optimal solution for the given resources and task. However, as we increase the number resources and tasks, ILP takes an increased amount of time to calculate the allocation matrix. In our solution, we always obtain the allocation matrix in less than a minute. However, we have some duality gap in our solution compared with the optimal solution obtained from (Mulla et al., 2016), which is acceptable if we address the problem as a real-time problem. Table 5.1 summarises the available state-of-the-art assignment algorithms and the contributions of this work.

Resources	Work	Skill-aware and Fairness-aware	KPI/SLA-aware	Resource utilisation-aware	Resource productivity-aware	Approximation guarantee
Machines	(Graham et al., 1979; Price, 1982) (Cheng and Sin, 1990; Mokotoff, 2001) (Lawler et al., 1982) (Coffman Jr et al., 1984) (Dowsland and Dowsland, 1992) (Wiese et al., 2013)	Partial[a]	Partial[b]	No	No	Partial[c]
Employees	(Krishnamoorthy and Ernst, 2001) (Ernst et al., 2004) (Smet et al., 2013) (Lapègue et al., 2014) (Prot et al., 2015)	Partial	Partial	No	No	No
Employees	(Su, 2002) (Powell et al., 1999) (Mitra et al., 2001) (Fields et al., 1992)	Partial	No	No	No	No
Employees	(Fletcher et al., 2012)	Yes	No	Yes	No	No
Employees	(Hamadi and Quimper, 2007)	Yes	No	No	No	No
Employees	(Mulla et al., 2016)	Yes	Yes	Yes	Yes	Yes
Employees	**This work**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

[a] "Partial" in this column indicates that several works are skill-aware but not fairness-aware and several works are fairness-aware but not skill-aware, while several works are both skill- and fairness-aware and others are neither skill- nor fairness-aware.
[b] "Partial" in this column indicates that only a few works are KPI/SLA-aware although those SLAs might be different than what is considered here (i.e., minimising the number and the magnitude of task deadline violations).
[c] "Partial" in this column indicates that only a few works have proven approximation guarantees.

TABLE 5.1: Summary of available state-of-the-art task assignment algorithms along with the contributions of this work (Mulla et al., 2016)

5.2 System Model

We consider the problem of allocating a set $\Pi = \{\pi_1, \pi_2, \ldots, \pi_n\}$ of n tasks to a set $R = \{r_1, r_2, \ldots, r_m\}$ of m employees (referred to as *resources*) in a services delivery setting. Each task π_j is characterised by a processing time and a *baseline time* b_j (also referred to as *deadline*). The processing time of a task depends on the resource that processes it and the processing time of π_j when processed by r_i is denoted by t_{ij}, where $j \in \{1, 2, \ldots, n\}$ and $i \in \{1, 2, \ldots, m\}$. A task π_j that cannot be processed by a resource r_i (e.g., if the resource does not have the skills to process it) is modelled by setting t_{ij} to ∞. Ideally, processing of each task π_j has to be completed within b_j time units.

For each resource π_i, we define the following terms. The workload w_i of π_i is defined as:

$$w_i \stackrel{\text{def}}{=} \sum_{\pi_j \in \Pi(i)} t_{ij} \tag{5.1}$$

where $\Pi(i)$ is the set of tasks assigned to r_i. Informally, it is the sum of the processing times that the resource takes for each task (t_{ij}) that is assigned to it. The utilisation u_i of π_i is defined as:

$$u_i \stackrel{\text{def}}{=} \frac{w_i}{s} \tag{5.2}$$

where s is the duration of the shift in which π_i works. Informally, utilization of a resource indicates the fraction of time the resource will be occupied with the assigned tasks in a shift. The productivity p_i of π_i is defined as:

$$p_i \stackrel{\text{def}}{=} \frac{\sum_{\pi_j \in \Pi(i)} b_j}{\sum_{\pi_j \in \Pi(i)} t_{ij}} \tag{5.3}$$

Informally, it is the ratio of the amount of expected time in which π_i needs to complete all the tasks assigned to him/her to the amount of actual time in which π_i will complete those tasks.

We also use a few other below mentioned notations in this section. Upper limit on the productivity of any resource is denoted by $\bar{p}$ and upper limit on the number of baseline violations is denoted by $\bar{v}$. The lower and upper bounds on the utilisation of any resource are denoted by $\underline{u}$ and $\bar{u}$, respectively. (*Bar* is used over a letter to denote the upper bound

and *underbar* is used to denote the lower bound). The average workload $\tilde{w}$ is defined as: $\tilde{w} = \frac{1}{m}\sum_{i=1}^{m} w_i$. The lower and upper bounds on the workload of a resource are $(1-\bar{w})\times\tilde{w}$ and $(1+\bar{w})\times\tilde{w}$, respectively.

5.3 ILP-based Task Allocation

In this section, we provide the the Integer linear programming solution provided by (Mulla et al., 2016), `Ialloc`, for efficiently allocating tasks to resources considering some of the constraints specific to SDOs. It is based on solving ILP and works as follows.

First, solve the ILP formulation (on x_{ij} variables) shown in Fig. 5.1 using one of the standard solvers (such as IBM ILOG CPLEX [1] and Gurobi Optimizer [2]).

$$\text{Minimize } \sum_{j=1}^{n}\sum_{i=1}^{m} x_{ij} \times obj_{ij}$$

$$\text{where } obj_{ij} = \begin{cases} -\log\left(\frac{v_{mag}\times b_j - t_{ij}}{b_j\times(v_{mag}-1)}\right) & \text{if } t_{ij} \geq b_j \\ 0 & \text{otherwise} \end{cases}$$

subject to the following constraints:

I1.	$\forall R \in R : \sum_{i=1}^{m} x_{ij} = 1$
I2.	$\sum_{j=1}^{n}\sum_{i=1}^{m} x_{ij} \times \mathbf{1}_{+}(t_{ij} - b_j) \leq \bar{v} \times n$
I3.	$\forall r_i \in R : \lvert w_i - \tilde{w}\rvert < \bar{w} \times \tilde{w}$
I4.	$\forall r_i \in R : \frac{\sum_{j=1}^{n} x_{ij}\times b_j}{\sum_{j=1}^{n} x_{ij}\times t_{ij}} < \bar{p}$
I5.	$\forall r_i \in R : \sum_{j=1}^{n} x_{ij} \times t_{ij} \geq \underline{u} \times s$
I6.	$\forall r_i \in R : \sum_{j=1}^{n} x_{ij} \times t_{ij} \leq \bar{u} \times s$
I7.	$\forall \pi_j \in \Pi, \forall r_i \in R : x_{ij} \in \{0,1\}$

$$\text{where } \mathbf{1}_{+}(f) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } f > 0 \\ 0 & \text{otherwise} \end{cases}$$

FIGURE 5.1: Integer Linear programming formulation for assigning tasks in Γ to resources in Π

The ILP formulation tries to maximise the operational KPIs by keeping the number of baseline violations within the defined limit and further by minimising the magnitude of these violations. The objective function is an approximation of an *indicator function*. This function sets a nonlinear cost for the magnitude of baseline violations. In this study, we have set this threshold to $v_{mag} \cdot b_j$ according to their paper which implies that if the

[1] http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/
[2] http://www.gurobi.com/products/gurobi-optimizer

magnitude of violation exceeds this value, a prohibitively large cost is acquired. Hence, the formulation increasingly tries to minimise the magnitude of violations as much as possible.

In the ILP shown in Fig. 5.1, each variable x_{ij} indicates the assignment of task π_j to resource r_i. Constraint $I1$ combined with constraint $I7$ enforces that each task is to be allocated exactly one resource. Constraint $I2$ enforces that the number of baseline violations is within the defined limit. Constraint $I3$ specifies that the workload of any resource or employee is within a specified range of the average workload of the team. Constraint $I4$ enforces that the productivity of any resource is within the designated threshold. Constraint $I5$ and $I6$ enforce lower and upper bounds on the utilisation of any resource. Finally, constraint $I7$ specifies the range for x_{ij} indicator variables.

Using the solution output by the solver, the tasks are assigned to resources or employees as follows. If $x_{ij} = 1$ then assign task π_j to resource r_i.

If the problem state turns out to be infeasible, we need to update the thresholds on productivity, utilisation and workload, and try to solve the formulation with the new thresholds; this process is performed iteratively until the solver outputs a feasible solution. The changing of thresholds can be performed efficiently with the help of the solver.

5.4 Tabu Search

Tabu search is generally implemented as a single search trajectory direct search method. The concept was originally proposed by (Glover, 1990), and since then, this research technique has been used in many applications. Tabu search has been applied to discrete combinatorial optimisation problems such as graph colouring and Travelling Salesman. Tabu search is initiated at a feasible starting point within a solution. After that, it identifies sequences of moves and whilst that process is executed, a candidate list is generated. An evaluation process can determine whether the member belongs to the list or not. Tabu search has three main advantages (Glover, 1990) : (1) the use of flexible attribute-based design to permit better evaluation criteria and historical search information to exploit the problem more thoroughly; (2) an associated mechanism of control based on the interplay between conditions that constrain and free the search process; and (3) intensification

strategies einforce move combinations and solution features historically. The below flow chart describes how Tabu search has been applied to our problem.

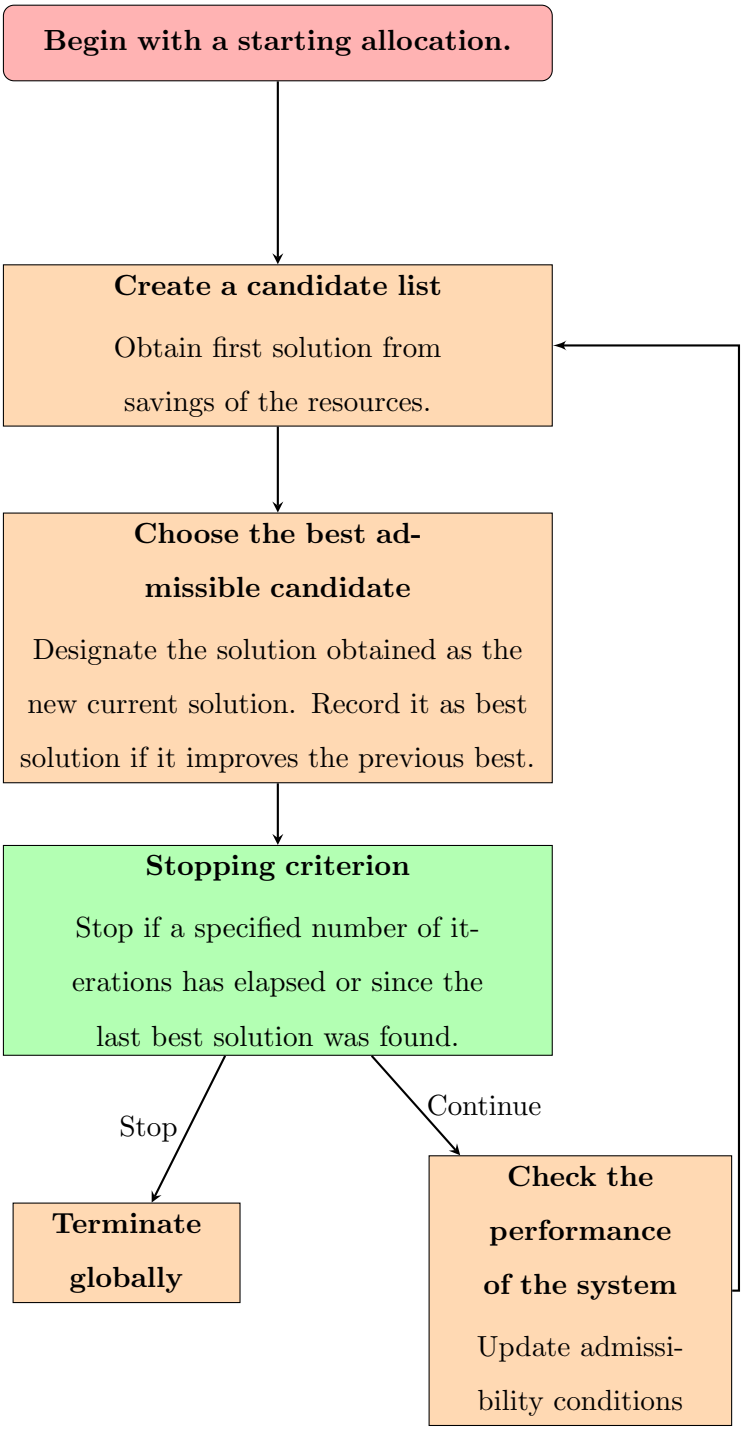


FIGURE 5.2: Workflow of Tabu Search

We used the TS approach to refine the solution obtained by using the given heuristic algorithm. Although a TS-based algorithm is proposed in (Liang et al., 2002) for the orienteering problem, our algorithm is different from this. Our algorithm works for the resource allocation matrix with the score of each resource, which is different from the graph discussed in (Liang et al., 2002). In (Zhou et al., 2015), a TS-based algorithm is used for a wireless relay network where the problem was simplified as an orienteering problem. The TS algorithm was then applied to obtain the best path from the current with a penalty function of budget and cost of path. Furthermore, our algorithm only cares about the last assignment for each resource in the allocation matrix. Thus, the proposed algorithm is more suitable for our problem.

5.4.1 Tabu Search-Based Task Allocation

To ensure that the system was non-biased in terms of team lead action, we used this approach. To ensure it was successful and to compare our result with the original ILP (Mulla et al., 2016), we extracted real-time data from the organisation. We refer to employees as resources here.

Fig.5.2 describes how we applied Tabu search in our task allocation problem. According to TS, we first need a solution to start with. In our method, we started with the allocation given from the savings of the resources. Saving of a resource can be calculated by:

$$savings_i = b_j - t_{ij} \tag{5.4}$$

where b_j is the baseline of the task and t_{ij} is the time line of the task j assigned to resource i. This t_{ij} has been taken from a random distribution of the same task performed by various resources. We calculated the savings of resources and whichever resource had the largest saving (ensuring the resource is able to handle this type of work), was allocated that task in the first step. However, if the savings of all the resources are negative i.e., $t_{ij} \geq b_j$, we assign the task to the resource who has the least negative savings. Then, we check the performance of the initial allocation. We calculate the utilisation, productivity, workload and number of violations from the allocation matrix $\mathbb{X}$. The calculation of these

Algorithm 6 Tabu

function TABU(R, Π, t, b, s) ▷ Where R - Resource array, Π - Set of Task, t - Timeline matrix, b - Baseline matrix, s - Shift time ▷ i - Resource index, j - Task index, $\mathbb{X}$ - Allocation matrix
    Initialize $\mathbb{X}$
    Call PERFORMANCEEVAL($\mathbb{X}, t, b$)
    do
        if $\mathbb{X}_{ij} == 1$ and $t_{ij} > b_j$ **then**
            $V\_task.append(j)$
        end if
        for each j in V_task **do**
            for each i in R **do**
                if $t_{ij} \neq -1$ **then**
                    if $U_i \leq \bar{u}\, and\, P_i \leq \bar{p}\, and\, W_i \leq \bar{w}$ **then**
                        $Tabu\_list_j.append(i)$
                    end if
                end if
            end for
        end for
        $\mathbb{X}, allocation\_update$ from TABUSEARCH($Tabu\_list, \mathbb{X}, t, b$)
    while $allocation\_update \neq 0$
    return $\mathbb{X}$
end function

vectors has been discussed in Eq. 5.1, 5.2, and 5.3. After that, we make a list of all violated tasks to perform our TS-based algorithm.

To validate or make the allocation close to the optimal solution, we work with only violated tasks in our TS-based algorithm. Each violated task from the first allocation $\mathbb{X}$ is taken and a candidate list of resources who can do that task is created. Before adding the resource to the candidate list, we need to check the current utilisation, productivity and workload of the resource. The current values of these metrics should be less than $\bar{u}$, $\bar{p}$ and $\bar{w}$, which are user driven input to the system. The resource will then be added to the current set of resources to perform the violated task. Then, we call the TS algorithm with the Tabu list and current allocation $\mathbb{X}$. Before going into the TS algorithm, we need to evaluate a score to choose the best candidate from the *Tabu list*. A score can be evaluated by the summation of workload and productivity with the penalty of the solution, i.e.,

$$score_i = (norm(p_i) + norm(w_i)) + pen_i \tag{5.5}$$

Algorithm 7 Tabu Search

function TABUSEARCH($Tabu\_list, \mathbb{X}, t, b$)▷ Where $Tabu\_list$ - Set of candidate list, $\mathbb{X}$ - Allocation matrix ▷ i - Resource index, j - Task index, $score_i$ - score of resource from Eq. 5.5
  $allocation\_update = 0$
  for $iteration = 1$ to $max_iterations$ **do**
    $tabu\_update \leftarrow 0$
    $\phi \leftarrow 1$
    for each j in $Tabu\_list$ **do**
      $\mathbb{X}' \leftarrow \mathbb{X}$
      select $i^{new} \in Tabu\_list[j]$ with the best $score_{i^{new}} - score_{i^{cur}}$
      $\mathbb{X}'_{i^{cur}{}_j} \leftarrow 0$
      $\mathbb{X}'_{i^{new}{}_j} \leftarrow 1$
      Call PERFORMANCEEVAL($\mathbb{X}', t, b$)
      Call PERFORMANCEEVAL($\mathbb{X}, t, b$)
      if $magV' \leq magV$ **then**
        $\mathbb{X} \leftarrow \mathbb{X}'$
        $\phi \leftarrow \phi/2$
        $allocation\_update \leftarrow 1$
        $tabu\_update \leftarrow 1$
        $W \leftarrow W\_temp$
        $P \leftarrow P\_temp$
      end if
    end for
    if $tabu\_update == 0$ **then**
      $\phi \leftarrow \phi * 2$
    end if
  end for
  return $\mathbb{X}, allocation\_update$
end function

where pen_i is the penalty associated with the solution, and is given by

$$pen_i = \begin{cases} 0, & \text{if } b_j \leq r_i \\ \phi * (r_i - b_j), & \text{if } b_j > r_i \end{cases} \tag{5.6}$$

where r_i is the remaining time of the resource from its shift and ϕ is a penalty parameter dynamically updated during the search. The parameter ϕ is initialised at a value of 1. However, the algorithm is robust with respect to this parameter.

Next, we call the TS algorithm with the Tabu list. First, we generate the score for each resource present in the Tabu list for the particular task. Then, we select the best resource

for the task and assign the task to that particular resource. We fix the Tabu list elements, which are obtained from the previous function, and will be updated in the score function if assignment is being performed. We update the $\mathbb{X}$ with the new selected assignment. Then, we again call the "PerformanceEval" function to check if the obtained solution (magnitude of violation) is better or not. If yes, we keep the updated assignment and update the W, P, U. Otherwise, we go to the next violated task. After we loop through all the tasks in the violated task list, if we obtain any updated elements in $\mathbb{X}$, we repeat the whole experiment and try to improve our result iteratively. Thus, we approach the optimal solution.

5.5 Experimental Results and Discussion

The proposed task allocation mechanism was implemented, and in this section, we discuss the results of applying our approach on real-world data from a transaction processing business unit within a large SDO. The resources in the organisation are skilled with regard to the processes. Different resources are skilled to execute different processes, and the proficiency levels of resources for the skills can vary. Each team lead is responsible for handling transactions (of various types) pertaining to certain processes. The team lead monitors the volume of transactions arriving every day (that he/she is expected to handle) and allocates them to resources in his/her team based on their skills/proficiency and the complexity of the transactions. The team lead manages all these mentally, day-in and day-out. Each transaction type comes with a *baseline* (unit of time) within which transaction instances of that type are expected to be completed. Any transaction extending beyond the baseline is considered to be violating the baseline KPI. We keep track of this KPI via the *number of violations* metric, which measures the number of transactions that violated the baseline. We also keep track of the *magnitude of violation*, the magnitude of time by which the baseline is violated. For example, if the baseline of a transaction type is 100 time units and if an instance of that transaction type took 125 time units, then the magnitude of violation is 25 time units. It is important to keep track of this because the penalty that the organisation needs to pay also depends on the magnitude of violation.

We applied the proposed approach on the transactions handled by several team leads to assess the efficacy of our approach. We present the results on the transactions handled by four team leads, TL_1, TL_2, TL_3, and TL_4 for a period of six months, January to June 2016. In order to study the efficacy of our approach, we need to estimate the likely number of violations and the magnitude of violations if the transactions are assigned to resources. To enable this, we extracted the *historical processing time distributions* for each of the employees for the various transaction types. For any transaction assigned to a resource, we randomly sample[3] the time from that resources' processing time distribution for the transaction type (pertaining to that transaction). Before taking the randomly sampled processing time from the distribution, we also discarded the outliers values. That means we discarded the processing times that were much less than that of the baseline (e.g., 25% of baseline). We did this because these outlier values can affect the standard productivity and utilisation values. Assuming that the resource would take the sampled time had he/she been assigned that transaction, we check if that time exceeds the baseline. If so, we record that as a violation and capture the magnitude by which it exceeds the baseline. For each transaction, we do this assignment *five* times and take the average metrics along with their confidence intervals.

The initial assignment for TS is done using a simple heuristic. For each transaction from Π, we check and assign the transaction to the resource who has the largest saving for that transaction, where saving is defined by:

$$savings_{ij} = b_j - t_{ij} \tag{5.7}$$

Here b_j signifies the baseline time for the transaction and t_{ij} denotes the sampled time for that transaction for resource i. If all the resources have negative savings for a particular task, we assign the task to the least negative savings. Thus, the initial allocation matrix $\mathbb{X}$ obtained is then subjected to TS, as explained in Section 4.

Table 5.2 presents the characteristics of the transactions handled by different team leads and compares the current manual approach with our proposed automated task allocation.

[3]Other sampling mechanisms such as weighted sampling, where more weights to sampling processing times from the recent past is applied. A detailed analysis and discussion of different sampling mechanisms is beyond the scope of this work.

It can be seen that the current approach leads to a significant number of violations for the team leads, with the percentage of violations being in the range of 14-46%. Using our proposed Tabu allocation, the deadline violations were in the range of 2-24%. On average, the number of violations was improved by 60% across all the team leads. Fig. 5.3 depicts the comparison of the proposed approach with the current approach. Fig. 5.3(a) depicts the percentage of violations for the current approach (dotted lines) for the four team leads and the average percentage of violations, along with the 95% confidence intervals (over five runs) for the proposed Tabu-based allocation (solid lines). Similarly Fig. 5.3(b) and Fig. 5.3(c) depict the average utilisation and productivity for the current approach (dotted lines) and the average utilisation and productivity, along with the 95% confidence intervals (over five runs), for the TS-based allocation (solid lines). As can be seen, our approach outperformed the current approach. The percentage of violations was consistently much smaller than the current approach. Further, the utilisation of resources was smaller than that of the current approach. In other words, due to the new allocation, the current resources are able to finish the tasks much earlier. These resources can be better utilised by assigning them other tasks or reallocating them to other groups. Furthermore, the productivity of resources is higher than that of the current approach.

TABLE 5.2: Comparison of the number of violations and their magnitude for the transactions handled by four team leads from a large service organisation, for a duration of six months in the year 2016, where Tabu-based allocation results in less violation than the current allocation

TL_1	**# Res.**	**# Trans.**	**Current allocation**				**Tabu-based allocation**			
			% Viol.	**Cum. Mag. Viol. (in secs)**	**Avg. Util.**	**Avg. Prod.**	**Avg. % Viol.**	**Cum. Mag. Viol. (in secs)**	**Avg. Util.**	**Avg. Prod.**
Jan	22	308	16.55	23154	12.79	114.72	10.00	11132	11.93	115.59
Feb		100	14.00	11253	11.94	114.73	5.00	3077	11.05	112.30
Mar		111	21.62	62229	18.00	108.34	16.21	11064	14.94	102.50
Apr		1620	19.38	140358	36.69	112.34	6.54	46075	32.19	124.70
May		2700	30.52	324849	36.17	106.14	15.11	134256	31.53	121.83
Jun		2346	27.32	234297	35.67	108.31	12.79	109459	31.30	121.80
TL_2										
Jan	17	5742	22.34	563531	46.35	127.21	3.03	79614	30.48	193.98
Feb		4782	28.37	572090	46.30	118.75	2.80	83481	28.36	186.89
Mar		6072	26.96	634317	53.54	114.03	2.81	120633	30.10	233.67
Apr		6647	27.26	732574	50.25	119.36	2.45	92278	28.78	207.77
May		6493	24.29	457616	44.01	131.65	10.08	121145	29.94	187.24
Jun		5164	23.99	424336	37.88	192.02	5.80	53523	24.27	209.18
TL_3										
Jan	32	3737	30.61	985369	53.54	115.17	13.28	429000	43.59	135.58
Feb		2888	35.53	854770	51.17	109.16	13.40	241739	39.45	140.03
Mar		2116	37.43	532662	39.69	109.27	17.72	245589	34.40	134.52
Apr		1825	33.60	607748	40.77	111.12	14.85	151599	32.68	133.90
May		1933	35.75	674532	35.75	111.61	24.21	355669	36.76	120.57
Jun		2198	35.21	935117	41.93	108.25	18.47	292266	33.69	150.83
TL_4										
Jan	22	1731	45.40	671828	37.68	95.29	20.92	226034	27.90	116.73
Feb		1824	36.51	548773	51.46	105.99	14.65	194856	40.20	165.33
Mar		1317	39.48	490642	42.75	140.01	14.27	146027	31.63	140.01
Apr		1488	43.88	470765	54.15	101.34	16.20	144945	39.97	128.46
May		2979	37.70	939586	50.31	91.58	19.40	342126	39.98	118.40
Jun		4560	42.40	793910	46.14	99.30	17.34	297356	34.98	129.12

Cum. Mag. Viol. - Cumulative Magnitude Violation.
Avg. Util. - Average Utilisation.
Avg. Prod. - Average Production.

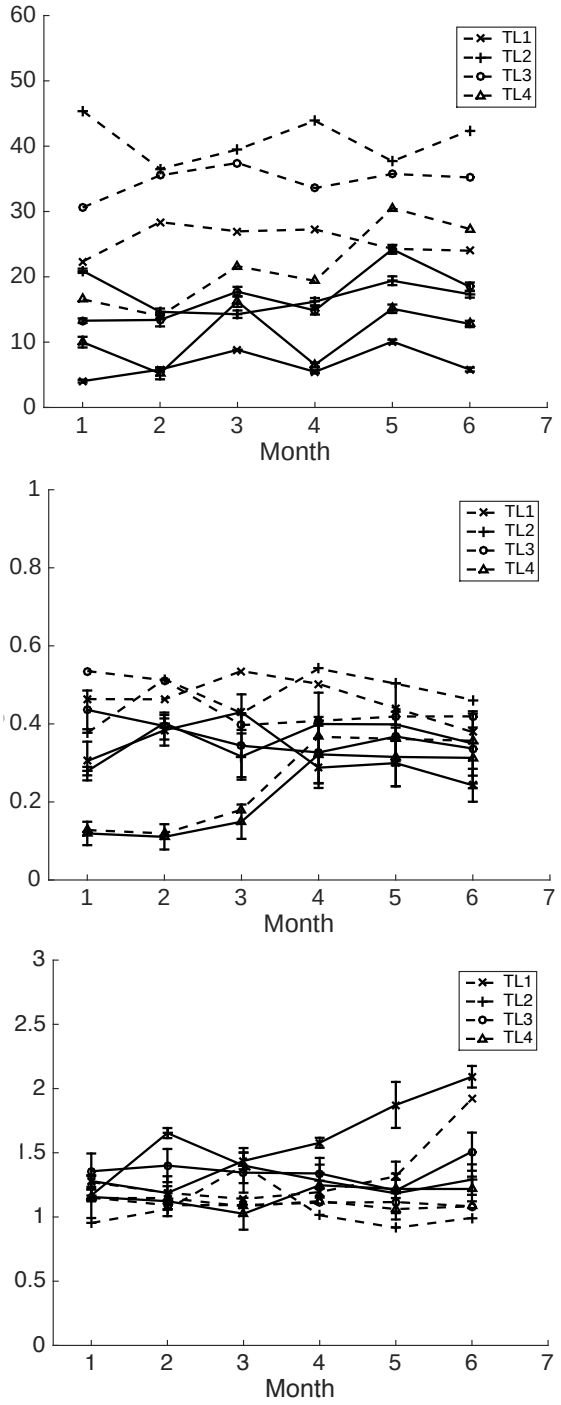


FIGURE 5.3: Comparison of percentage of violations, average utilisation and productivity metrics for the four team leads, for both the current approach and the proposed Tabu-based approach. The solid line in the figures corresponds to the Tabu-based approach while the dotted line represents the current method. (a) percentage of violations; (b)utilisation; (c) productivity.

In our evaluations, the proposed approach was able to reduce the number of violations

by 20% and the magnitude of violations by up to 75%. Furthermore, it increased the productivity of resources by 10% and the average workload and utilisation was reduced by 35%, compared to the currently practiced manual task allocation in the organisation.

Our proposed approach is computationally tractable with running times of less than three minutes. Fig 4.4 shows a comparison of the time taken in ILP and our approach for one team lead, TL1, over one month of data. Each day, the team lead encounters a different number of transactions. For example, the graph (Fig. 4.4) shows that on the 22^{nd} day there were 24 resources with 200 transaction; our TS approach only takes 6.5 seconds with 98% confidence in percentage of violation. As another example, for a team lead who has 1006 tasks in his/her pool and 20 resources, according to Fig. 4.4, ILP takes more than one hour to assign tasks to the resources, but with our approach, the assignment takes less than 10 minutes. Table 5.3 explains the detail results regarding time comparison of ILP and our approach.

Table 5.3: Running time comparison for Integer Linear programming and Tabu for TL1

Day_No	nResources	nTasks	AvgRunTime(s) ILP	AvgRunTime(s) Tabu
1	34	336	845.84	156.77
2	35	449	1353.83	300.27
3	31	460	1052.11	442.41
4	32	546	1567.87	669.44
5	3	43	1.98	0.1
6	29	373	788.37	179.83
7	33	549	1467.64	207.84
8	30	558	1429.09	369.61
9	29	466	1047.74	187.78
10	28	583	1243.24	194.67
11	5	103	7.87	0.42
12	28	398	732.70	105.02
13	37	572	1990.59	346.20
14	34	537	2255.12	265.02
15	34	640	2033.95	366.25
16	34	449	1440.04	161.74
17	5	91	1.61	0.42
18	27	393	692.26	167.42
19	32	398	979.41	184.46
20	31	586	1537.53	885.53
21	34	593	1858.50	419.28
22	3	42	2.78	0.36
23	32	415	1031.79	356.98
24	34	522	1455.91	775.76
25	31	529	1341.17	386.81
26	26	437	670.00	162.52
27	4	87	5.43	1.56
28	32	466	1095.15	225.84
29	35	606	2111.94	624.82

5.6 Chapter Summary

In this chapter, we studied the problem of automated allocation of tasks to human resources in a SDO setting with the goal of improving the operational KPIs and the processing of high dimensional data. For this problem, we proposed a meta-heuristic-based task allocation scheme which aims to minimise the number of tasks missing the deadlines and to minimise the magnitude by which they are missed, taking into consideration the resource skills, utilisation, productivity, fairness and KPIs while allocating the tasks. In the experimental evaluations of transaction data from a large services organisation, the proposed approach reduced the number of deadline misses and the magnitude of violations by 75%. As future work, we intend to design an algorithm that dynamically reallocates tasks periodically by closely monitoring the real-time performance of resources. Although, the data would be very high dimensional and complex for real-world applications. Further, in future work, various sampling strategies can be used to select the processing time for a particular task, instead of a random distribution as in the current work.

Chapter 6

Discussion and Conclusion

This chapter summarises the motivation behind this thesis and presents a discussion of the main outcomes. Evidence for the achievement of the research aims introduced in Chapter 1 is provided, and future work is proposed to address the limitations of this research.

In a world where big data analytics is gaining more and more importance, the investigation of detailed characteristics of point cloud data has become a fundamental task. If we assume that there is an underlying manifold structure to a point cloud, methods that can help to analyse and understand the data by determining the manifold's dimensionality, topology and degree of non-linearity are of increasingly higher value.

In computer science and machine learning research, where we talk about high-performance computing, fast computing and parallel computing, the traditional ideas in homology calculation cannot perform well with the amount of noisy, high-dimensional point cloud data. Such data also becomes computationally costly while processing, which is perhaps an even more significant challenge.

At a time where neural networks have become more potent in object recognition, the work of this thesis focused on recognising an object's topology from sample data without detecting the object itself. This path has not been accessible to date. Successful learning of simple topology proved challenging and required development of particular techniques for avoiding instabilities in the learning process. Without these techniques, early attempts to learn topology resulted in poor prediction outcomes. Later, these insights made it possible

to develop a high-performing model for complex topology data with a large point cloud. However, the data sets were generated by a script.

Since directly inferring geometric structure represented by point cloud from complicated barcodes is not possible, we based our PH approach on supervised learning; consequently, our results are not strongly influenced by the size of the dataset. More generally, we believe that PH provides an essential new tool for attacking difficult topology analysis problems for point cloud data. While much remains to be understood regarding the interface between PH and machine learning, we hope that this study helps to provide some insight into the neural network to understand the topology of the data by exploring the impact on concrete, topologically complex data. Another major problem that will arise as we construct more sophisticated models is the problem of more complex topology data in a higher dimension.

This thesis presented a series of investigations into these topics, with particular focus on evaluating the performance of manifold learning, topology analysis and optimisation on manifolds represented by point cloud data. In each part of the research project, a new method for addressing several of these aspects was proposed and tested. This resulted in the development of several point cloud data-based methods. Their potential impact, limitations and possible directions for future research will be discussed in the following sections.

6.1 Validation of Non-linear Dimensionality Reduction Methods (Chapter 2)

In investigating manifold learning techniques, the primary work is to project the data onto a low-dimensional surface. However, manifold learning is restrictive in the sense that those surfaces can be non-linear. What if the best representation lies in some weirdly shaped surface? Non-linear DR techniques cannot recognise the topology distortion that occurs in the data. This study obtained a sub-optimal representation of the data in lower dimensions. Although methods like Isomap and LLE create k-nearest neighbour graphs for DR, this does not help us to understand topological change. This disadvantage is practically off-set to correct DR method. The main advantage of PH for point cloud data

is that it can detect the topology distortion during embedding, where residual variance can fail. This makes PH most useful for embedding of either ad-hoc or very high dimensional point cloud data.

To our best knowledge, non-linear manifold learning algorithms like Isomap and LLE are controlled by the tuning parameters k, which defines the neighbourhood size around each data point. Making k too large or too small can change the nature and dimensionality of the embedding solution. Another issue is how to deal with new data points. All of these algorithms are batch algorithms as they rely on neighbourhood graphs to model the local structure of the data manifold (Izenman, 2012; van der Maaten et al., 2009). These methods require one to rerun the entire algorithm on a data set consisting of the original data augmented by the new points. Thus, all the manifold learning algorithms suffer from not being generalisable (Izenman, 2012). This research, which may improve the performance of embedding techniques for point cloud data, uses the nearest neighbourhood size k.

6.2 Topology Detection for 3D Data (Chapter 3)

Motivated to improve the performance of topology detection for large 3D point cloud data, we applied algorithms for constructing a useful combinatorial representation of topological information regarding high dimensional data. We generated a substantial amount of 2D and 3D data with different topological features. Motivated to improve the performance of topology detection of large 3D point cloud data, deep learning-based strategies were examined. Most general and supervised learning-based methods and convolutional neural networks have been used for identifying arbitrary topologies of large point clouds.

The secondary problem studied in this chapter was how to calculate correct Betti numbers in PH for large point cloud data. The idea behind the use of these point cloud data is to apply these methods to real-world applications. In this study, we proposed a CNN-based neural network architecture for processing large point cloud data sampled in a Euclidean space. This study converted the point set into voxels which is useful for understanding topological features such as holes, bubbles or their higher dimensional equivalents. To

handle the non-uniform point sampling issue, we used equal voxel sizes for 3D according to local point densities. These contributions enabled us to achieve correct PH calculations on challenging benchmarks of large 3D point clouds.

6.3 Optimisation Over Manifolds (Chapter 4)

Optimisation algorithms on manifolds have shown great ability to find solutions to non-convex problems in reasonable amounts of time. Research has been performed to solve many optimisation problems over manifolds. However, these studies have always considered Riemannian structure. To solve non-convex optimisation problems, whose search spaces can be of a Riemannian structure, we combined tractable relaxations (when available) with a Riemannian optimisation procedure (Boumal, 2014). In this study, we explored the problem without requiring the Riemannian structure. Given the nature and complexity of non-linearly constrained optimisation problems, it is ridiculous to assume that a single best algorithm will solve the problem. We expect there to be a variety of suitable algorithms, each occupying some importance for real-world applications. How well the proposed algorithm performs remains to be determined, but we feel it could be important. A promising class of problems are those involving high dimensional manifolds. The gradient system would require an iterative solver, but the theoretical basis of our algorithm does not require an accurate solution of the primal-dual equations, and only one system needs to be solved.

Although we have an exact line-search, we currently use a simple backtracking method for the curvilinear search. The line-search presumes we are minimising a logarithmic barrier function and that singularity is encountered somewhere along the search direction. The search does not fail when there is no singularity, but it can possibly treat this case better which can be future direction of this research.

The material of chapter 4 has not been published to date and some of it is still exploratory in its nature.

6.4 Further Work

The fields of topological data analysis and optimisation on manifolds are gaining growing attention in research and applications. As we argued in this thesis, tools are readily available to solve and analyse data processing problems on manifolds, and we contributed to some of them. But as is conventional with such research attempts, more questions are left unanswered at the end of the journey than at its onset.

It is essential to identify how to accelerate inference speed of our proposed network, especially for CNN layers, by sharing more computation in each local voxel. It is also of interest to find applications in higher dimensional spaces (4D, 5D and more) where CNN-based method can be tested.

The development, training and testing of the proposed methods on more extensive real-world data sets is an area of future investigation. There are many possibilities left to investigate such as the incorporation of more sophisticated techniques from machine learning to improve accuracy and other performance measures. With the availability of faster computers this area of research has the potential to develop highly competitive state-of-the-art algorithms. Another development related to this is the use of Betti numbers as a feature for understanding the connection between PH and machine learning. Our investigation has addressed this aspect by showing that neural networks can recognise a manifold's topology from data without having full access to all details of the manifold.

There are significant, open problems relating to topology learning for networks and medical data and to designing representations for objects and images. We do not want to specify the topology for each data category. Instead, we should learn the topology structure from point cloud data. Given the ability to acquire new training data, an attractive intermediate step is to learn topology specifications from more detailed labels, without any supervision.

We can be at the limits of current techniques for point cloud representation in high dimensions. The types of models that we would ideally like to build are substantially different from those that we can build in practice. To model topological properties in more detail, we will likely need to augment, or entirely replace, the low-level features that we currently use.

Further work in optimisation over manifolds would involve a more rigorous study of the steepest descent method to better understand the geometry of the sparse representation of the manifold, and hopefully to enable implementation of a deterministic algorithm. For independent component analysis, it is of interest to identify if the manifold-based methods presented here can perform as well on a more advanced problem, and perhaps to further study the potential of the interior point method with Hessian modification.

Appendix A

Publications During PhD Candidacy

Over the course of my candidacy, I was involved in and contributed to several works which are directly related to the main body of my thesis. These are listed here with a brief description of the topic and my contribution.

In Paul and Chalup (2017), a new PH method was proposed to validate the non-linear dimensionality reduction. I was the main author and also the main generator of ideas; I also developed the whole simulation.

In Paul et al. (2017), the primary results from my industrial experience were presented. I was a part of the ideas generation process and was involved in the design of several key parts of the architecture. I was also a significant contributor to the content of the article.

Rahul Paul, Stephan K Chalup. "Estimating Betti Numbers using Deep Learning ". In preparation

Rahul Paul, Stephan K Chalup. "A Barrier Algorithm Approach for Optimization Problems Over Non-Linear Manifolds ". In preparation

Appendix B

Code to generate data and figures

B.1 Swiss Roll data with all holes

```
NUM = 10000; % number of points considered
coords3 = zeros(3,NUM)';
coords4 = zeros(4,NUM)';
%% Swiss roll with holes
p=4;
roll = zeros(4,NUM);
for i = 1:NUM
    x1 = rand*p;
    x2 = rand;
    roll(1,i) = (1+(2*pi*sqrt(x1)))*cos(2*pi*sqrt(x1));
    roll(2,i) = (1+(2*pi*sqrt(x1)))*sin(2*pi*sqrt(x1));
    roll(3,i) = 20*x2;
    roll(4,i) = x1;
end
coords3 = [roll(1,:)', roll(2,:)', roll(3,:)'];
z_1 = 10;
x_1=  (1+(2*pi*sqrt(.5)))*cos(2*pi*sqrt(.5));
y_1 = (1+(2*pi*sqrt(.5)))*sin(2*pi*sqrt(.5));
x_2=  (1+(2*pi*sqrt(1)))*cos(2*pi*sqrt(1));
y_2 = (1+(2*pi*sqrt(1)))*sin(2*pi*sqrt(1));
x_3=  (1+(2*pi*sqrt(1.5)))*cos(2*pi*sqrt(1.5));
y_3 = (1+(2*pi*sqrt(1.5)))*sin(2*pi*sqrt(1.5));
x_4=  (1+(2*pi*sqrt(2)))*cos(2*pi*sqrt(2));
y_4 = (1+(2*pi*sqrt(2)))*sin(2*pi*sqrt(2));
x_5=  (1+(2*pi*sqrt(2.5)))*cos(2*pi*sqrt(2.5));
y_5 = (1+(2*pi*sqrt(2.5)))*sin(2*pi*sqrt(2.5));
```

```
x_6=  (1+(2*pi*sqrt(3)))*cos(2*pi*sqrt(3));
y_6 = (1+(2*pi*sqrt(3)))*sin(2*pi*sqrt(3));
x_7=  (1+(2*pi*sqrt(3.5)))*cos(2*pi*sqrt(3.5));
y_7 = (1+(2*pi*sqrt(3.5)))*sin(2*pi*sqrt(3.5));
center_1=[x_1,y_1,z_1];
center_2=[x_2,y_2,z_1];
center_3=[x_3,y_3,z_1];
center_4=[x_4,y_4,z_1];
center_5=[x_5,y_5,z_1];
center_6=[x_6,y_6,z_1];
center_7=[x_7,y_7,z_1];
hole_data=[];
temp=[];
for i= 1: size(coords3(:,1))
   dis_1(i) = sqrt(sum((coords3(i,:)-center_1).^ 2));
   dis_2(i) = sqrt(sum((coords3(i,:)-center_2).^ 2));
   dis_3(i) = sqrt(sum((coords3(i,:)-center_3).^ 2));
   dis_4(i) = sqrt(sum((coords3(i,:)-center_4).^ 2));
   dis_5(i) = sqrt(sum((coords3(i,:)-center_5).^ 2));
   dis_6(i) = sqrt(sum((coords3(i,:)-center_6).^ 2));
   dis_7(i) = sqrt(sum((coords3(i,:)-center_7).^ 2));
    if (dis_1(i)>4 && dis_2(i)>4 && dis_3(i)>4 && dis_4(i)>4 &&
    dis_5(i)>4 && dis_6(i)>4 && dis_7(i)>4)
    %% Remove the dis according to the hole number
       hole_data = [hole_data;coords3(i,:)];
       temp = [temp;roll(4,i)];
    end
end
coords3=hole_data;
```

B.2 Heated Roll data with all the holes

```
p=4;
roll = zeros(4,NUM);
for i = 1:NUM
    x1 = rand*p;
    x2 = rand;
    roll(1,i) = (1+(x2*2-1)^2)*2*pi*sqrt(x1)*cos(2*pi*sqrt(x1));
    roll(2,i) = (1+(x2*2-1)^2)*2*pi*sqrt(x1)*sin(2*pi*sqrt(x1)) ;
    roll(3,i) = 20*x2;
    roll(4,i) = x1;
end
%coords4 = heated';
```

```
coords3 = [roll(1,:)', roll(2,:)', roll(3,:)'];
z_1 = 10;
x_1= (1+(.5*2-1)^2)*2*pi*sqrt(.5)*cos(2*pi*sqrt(.5));
y_1= (1+(.5*2-1)^2)*2*pi*sqrt(.5)*sin(2*pi*sqrt(.5));
x_2= (1+(.5*2-1)^2)*2*pi*sqrt(1)*cos(2*pi*sqrt(1));
y_2= (1+(.5*2-1)^2)*2*pi*sqrt(1)*sin(2*pi*sqrt(1));
x_3= (1+(.5*2-1)^2)*2*pi*sqrt(1.5)*cos(2*pi*sqrt(1.5));
y_3= (1+(.5*2-1)^2)*2*pi*sqrt(1.5)*sin(2*pi*sqrt(1.5));
x_4= (1+(.5*2-1)^2)*2*pi*sqrt(2)*cos(2*pi*sqrt(2));
y_4= (1+(.5*2-1)^2)*2*pi*sqrt(2)*sin(2*pi*sqrt(2));
x_5= (1+(.5*2-1)^2)*2*pi*sqrt(2.5)*cos(2*pi*sqrt(2.5));
y_5= (1+(.5*2-1)^2)*2*pi*sqrt(2.5)*sin(2*pi*sqrt(2.5));
x_6= (1+(.5*2-1)^2)*2*pi*sqrt(3)*cos(2*pi*sqrt(3));
y_6= (1+(.5*2-1)^2)*2*pi*sqrt(3)*sin(2*pi*sqrt(3));
x_7= (1+(.5*2-1)^2)*2*pi*sqrt(3.5)*cos(2*pi*sqrt(3.5));
y_7= (1+(.5*2-1)^2)*2*pi*sqrt(3.5)*sin(2*pi*sqrt(3.5));
center_1=[x_1,y_1,z_1];
center_2=[x_2,y_2,z_1];
center_3=[x_3,y_3,z_1];
center_4=[x_4,y_4,z_1];
center_5=[x_5,y_5,z_1];
center_6=[x_6,y_6,z_1];
center_7=[x_7,y_7,z_1];
hole_data=[];
temp=[];
for i= 1: size(coords3(:,1))
    dis_1(i) = sqrt(sum((coords3(i,:)-center_1).^ 2));
    dis_2(i) = sqrt(sum((coords3(i,:)-center_2).^ 2));
    dis_3(i) = sqrt(sum((coords3(i,:)-center_3).^ 2));
    dis_4(i) = sqrt(sum((coords3(i,:)-center_4).^ 2));
    dis_5(i) = sqrt(sum((coords3(i,:)-center_5).^ 2));
    dis_6(i) = sqrt(sum((coords3(i,:)-center_6).^ 2));
    dis_7(i) = sqrt(sum((coords3(i,:)-center_7).^ 2));
     if (dis_1(i)>4 && dis_2(i)>4 && dis_3(i)>4 &&
     dis_4(i)>4 && dis_5(i)>4 && dis_6(i)>4 && dis_7(i)>4)
     %% Remove the dis according to the hole number
       hole_data = [hole_data;coords3(i,:)];
       temp = [temp;roll(4,i)];
    end
end
coords3=hole_data;
% Twist with hole
points = NUM;
p = 4;
%t=5;
```

```
length = 50;
twists = 3;
twist = zeros(4,points);
for i = 1:points
x1 = rand*p - p/2;
x2 = rand;
twist(1,i) = length*x2;
twist(2,i) = x1*2*pi*cos(pi*twists*x2);
twist(3,i) = x1*2*pi*sin(pi*twists*x2);
twist(4,i) = x1;
end
%coords4 = twist';
coords3 = [twist(1,:)', twist(2,:)', twist(3,:)'];
slot=length/7;
x_1= slot/2;
y_1=.5*cos(pi*twists*.5);
z_1=.65*sin(pi*twists*.65);
x_2 = x_1+slot;
x_3 = x_2+slot;
x_4= x_3+slot;
x_5 = x_4+slot;
x_6 = x_5+slot;
x_7 = x_6+slot;
hole_data=[];
temp=[];
center_1=[x_1,y_1,z_1];
center_2=[x_2,y_1,z_1];
center_3=[x_3,y_1,z_1];
center_4=[x_4,y_1,z_1];
center_5=[x_5,y_1,z_1];
center_6=[x_6,y_1,z_1];
center_7=[x_7,y_1,z_1];

for i= 1: size(coords3(:,1))
    dis_1(i) = sqrt(sum((coords3(i,:)-center_1).^ 2));
    dis_2(i) = sqrt(sum((coords3(i,:)-center_2).^ 2));
    dis_3(i) = sqrt(sum((coords3(i,:)-center_3).^ 2));
    dis_4(i) = sqrt(sum((coords3(i,:)-center_4).^ 2));
    dis_5(i) = sqrt(sum((coords3(i,:)-center_5).^ 2));
    dis_6(i) = sqrt(sum((coords3(i,:)-center_6).^ 2));
    dis_7(i) = sqrt(sum((coords3(i,:)-center_7).^ 2));
    if (dis_1(i)>2.5 && dis_2(i)>2.5 && dis_3(i)>2.5 &&
    dis_4(i)>2.5 && dis_5(i)>2.5 && dis_6(i)>2.5 &&
    dis_7(i)>2.5 )
       hole_data = [hole_data;coords3(i,:)];
```

```
        temp = [temp;twist(4,i)];
    end
end
coords3=hole_data;
```

B.3 Torus Data Set

```
NUM=4000
roll = zeros(4,NUM);
for i = 1:NUM
    x1 = (2*pi) * rand(1);
    x2 = (2*pi) * rand(1);
    roll(1,i) = (2+cos(x1))*cos(x2);
    roll(2,i) = (2+cos(x1))*sin(x2);
    roll(3,i) = sin(x1);
end
coords3 = [roll(1,:)', roll(2,:)', roll(3,:)'];
hold on
```

B.4 Sphere Data Set

```
NUM=4000
roll = zeros(4,NUM);
for i = 1:NUM
    x1 = (2*pi) * rand(1);
    x2 = (2*pi) * rand(1);
    roll(1,i) = (cos(x1))*cos(x2);
    roll(2,i) = (sin(x1))*cos(x2);
    roll(3,i) = sin(x2);
end
coords3 = [roll(1,:)', roll(2,:)', roll(3,:)'];
```

Bibliography

Absil, P.-A. (2009). Numerical representations of a universal subspace flow for linear programs. *Communications in Information and Systems*, 8(2):71–84.

Absil, P.-A., Mahony, R., and Sepulchre, R. (2009). *Optimization algorithms on matrix manifolds*. Princeton University Press.

Absil, P.-A., Mahony, R., and Sepulchre, R. (2010). Optimization on manifolds: Methods and applications. In *Recent Advances in Optimization and its Applications in Engineering*, pages 125–144. Springer.

Absil, P.-A. and Malick, J. (2012). Projection-like retractions on matrix manifolds. *SIAM Journal on Optimization*, 22(1):135–158.

Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., and Ziegelmeier, L. (2017). Persistence images: a stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35.

Adams, H., Tausz, A., and Vejdemo-Johansson, M. (2014). Javaplex: A research software package for persistent (co) homology. In *International Congress on Mathematical Software*, pages 129–136. Springer.

Adcock, A., Carlsson, E., and Carlsson, G. (2013). The ring of algebraic functions on persistence bar codes. *arXiv preprint arXiv:1304.0530*.

Adler, R. L., Dedieu, J.-P., Margulies, J. Y., Martens, M., and Shub, M. (2002). Newton's method on riemannian manifolds and a geometric model for the human spine. *IMA Journal of Numerical Analysis*, 22(3):359–390.

Akkucuk, U. and Carroll, J. D. (2006). Paramap vs. isomap: a comparison of two nonlinear mapping algorithms. *Journal of Classification*, 23(2):221–254.

Aupetit, M. (2007). Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing*, 70(7):1304 – 1330. Advances in Computational Intelligence and Learning.

Balasubramanian, M. and Schwartz, E. L. (2002a). The isomap algorithm and topological stability. *Science*, 295(5552):7–7.

Balasubramanian, M. and Schwartz, E. L. (2002b). The isomap algorithm and topological stability. *Science*, 295(5552):7–7.

Baraniuk, R. G. and Wakin, M. B. (2009). Random projections of smooth manifolds. *Foundations of Computational Mathematics*, 9(1):51–77.

Boumal, N. (2014). *Optimization and estimation on manifolds*. PhD thesis, Princeton University, NJ, USA.

Boumal, N., Mishra, B., Absil, P.-A., and Sepulchre, R. (2014). Manopt, a matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Brockett, R. W. (1993). Differential geometry and the design of gradient algorithms. In *Proc. Symp. Pure Math., AMS*, volume 54, pages 69–92.

Burer, S., Monteiro, R. D., and Zhang, Y. (2003). A computational study of a gradient-based log-barrier algorithm for a class of large-scale sdps. *Mathematical Programming*, 95(2):359–379.

Cang, Z. and Wei, G.-W. (2017). Topologynet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLOS Computational Biology*, 13(7):1–27.

Carlsson, G., Jardine, R., Feichtner-Kozlov, D., Morozov, D., Chazal, F., de Silva, V., Fasy, B., Johnson, J., Kahle, M., Lerman, G., et al. (2009). Topological data analysis

and machine learning theory. Technical report, Banff International Research Station for Mathematical Innovation and Discovery.

Chen, C. and Kerber, M. (2011). Persistent homology computation with a twist. In *Proceedings 27th European Workshop on Computational Geometry*, volume 11.

Cheng, T. and Sin, C. (1990). A state-of-the-art review of parallel-machine scheduling research. *European Journal of Operational Research*, 47(3):271–292.

Coffman Jr, E. G., Garey, M. R., and Johnson, D. S. (1984). Approximation algorithms for bin-packing—an updated survey. In *Algorithm Design for Computer System Design*, pages 49–106. Springer.

Cox, T. F. and Cox, M. A. (2000). *Multidimensional scaling*. CRC Press.

Dedieu, J.-P., Malajovich, G., and Shub, M. (2005). On the curvature of the central path of linear programming theory. *Foundations of Computational Mathematics*, 5(2):145–171.

Dedieu, J.-P., Priouret, P., and Malajovich, G. (2003). Newton's method on riemannian manifolds: covariant alpha theory. *IMA Journal of Numerical Analysis*, 23(3):395–419.

Dold, A. (2012). *Lectures on algebraic topology*, volume 200. Springer Science & Business Media.

Dowsland, K. A. and Dowsland, W. B. (1992). Packing problems. *European Journal of Operational Research*, 56(1):2–14.

Edelman, A., Arias, T. A., and Smith, S. T. (1998). The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353.

Edelsbrunner, H. and Harer, J. (2010). *Computational topology: an introduction*. American Mathematical Soc.

Edelsbrunner, H., Letscher, D., and Zomorodian, A. (2000). Topological persistence and simplification. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 454–463. IEEE.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., and Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27.

Fields, R. K., Quinn, P. R., and Blackley, T. (1992). System and method for making staff schedules as a function of available resources as well as employee skill level, availability and priority. US Patent 5,111,391.

Fletcher, B., Laithwaite, R. N. W., Selensky, E., and Sexton, P. (2012). Optimizing resource assignment. US Patent App. 13/664,338.

France, S. and Carroll, D. (2007). Development of an agreement metric based upon the rand index for the evaluation of dimensionality reduction techniques, with applications to mapping customer data. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 499–517. Springer.

Gabay, D. (1982). Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, 37(2):177–219.

Gashler, M., Ventura, D., and Martinez, T. (2011). Manifold learning by graduated optimization. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41(6):1458–1470.

Gertz, M., Nocedal, J., and Sartenar, A. (2004). A starting point strategy for nonlinear interior methods. *Applied Mathematics Letters*, 17(8):945–952.

Ghrist, R. (2008). Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75.

Glover, F. (1990). Tabu search: A tutorial. *Interfaces*, 20(4):74–94.

Gracia, A., González, S., Robles, V., and Menasalvas, E. (2014). A methodology to compare dimensionality reduction algorithms in terms of loss of quality. *Information Sciences*, 270:1–27.

Graham, R. L., Lawler, E. L., Lenstra, J. K., and Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5:287–326.

Hamadi, Y. and Quimper, C.-G. (2007). Allocating resources to tasks in workflows. US Patent App. 11/669,098.

Helmke, U., Hüper, K., Lee, P. Y., Moore, J., et al. (2004). Essential matrix estimation via newton-type methods. In *16th International Symposium on Mathematical Theory of Network and System (MTNS), Leuven*.

Helmke, U., Huper, K., and Moore, J. B. (2002a). Quadratically convergent algorithms for optimal dextrous hand grasping. *IEEE Transactions on Robotics and Automation*, 18(2):138–146.

Helmke, U. and Moore, J. B. (2012). *Optimization and dynamical systems*. Springer Science & Business Media.

Helmke, U., Ricardo, S., and Yoshizawa, S. (2002b). Newton's algorithm in euclidean jordan algebras, with applications to robotics. *Communications in Information and Systems*, 2(3):283–298.

Hirsch, M. W. (1976). *Differential Topology*. Springer-Verlag.

Hofer, C., Kwitt, R., Niethammer, M., and Uhl, A. (2017). Deep learning with topological signatures. In *Advances in Neural Information Processing Systems*, pages 1633–1643.

Hosseini, R. and Sra, S. (2015). Matrix manifold optimization for gaussian mixtures. In *Advances in Neural Information Processing Systems*, pages 910–918.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417.

Irvin, J. (2017). Convolutional neural networks for 3d mnist image classification.

Izenman, A. J. (2012). Introduction to manifold learning. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(5):439–446.

Ji, H. (2007). *Optimization approaches on smooth manifolds*. PhD thesis, The Australian National University.

Ji, H., Manton, J. H., and Moore, J. B. (2008). A globally convergent conjugate gradient method for minimizing self-concordant functions on riemannian manifolds. *IFAC Proceedings Volumes*, 41(2):14313–14318.

Jiang, D., Moore, J. B., and Ji, H. (2007). Self-concordant functions for optimization on smooth manifolds. *Journal of Global Optimization*, 38(3):437–457.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer Series in Statistics. Springer-Verlag, New York, second edition.

Kaczynski, T., Mischaikow, K. M., and Mrozek, M. (2004). *Computational Homology*, volume 157. Springer Science & Business Media.

Kaski, S. and Peltonen, J. (2011). Dimensionality reduction for data visualization [applications corner]. *IEEE signal processing magazine*, 28(2):100–104.

Krishnamoorthy, M. and Ernst, A. T. (2001). The personnel task scheduling problem. In *Optimization Methods and Applications*, pages 343–368. Springer.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kvernelv, V. B. (2013). Optimization on matrix manifolds with applications to blind source separation. Master's thesis.

Lapègue, T., Bellenguez-Morineau, O., and Prot, D. (2014). Methods for the shift design and personnel task scheduling problem. In *MOSIM 2014, 10ème Conférence Francophone de Modélisation, Optimisation et Simulation*.

LaValle, S. M. (2006). *Planning algorithms*. Cambridge university press.

Lawler, E. L., Lenstra, J. K., and Kan, A. R. (1982). Recent developments in deterministic sequencing and scheduling: a survey. In M. A. H. Dempster, J. K. Lenstra, A. H. G. R. K., editor, *Deterministic and stochastic scheduling*, pages 35–73. Springer.

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, 6(Nov):1783–1816.

Lawrence, N. D. (2012). A unifying probabilistic perspective for spectral dimensionality reduction: Insights and new models. *Journal of Machine Learning Research*, 13(May):1609–1638.

Lee, J. (2010). *Introduction to topological manifolds*, volume 940. Springer Science & Business Media.

Lee, J., Verleysen, M., et al. (2014). Two key properties of dimensionality reduction methods. In *Computational Intelligence and Data Mining (CIDM), 2014 IEEE Symposium on*, pages 163–170. IEEE.

Lee, J. A. and Verleysen, M. (2007). *Nonlinear Dimensionality Reduction*. Springer, New York.

Lee, J. A. and Verleysen, M. (2009). Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72(7):1431–1443.

Lee, J. A. and Verleysen, M. (2010). Scale-independent quality criteria for dimensionality reduction. *Pattern Recognition Letters*, 31(14):2248–2257.

Li, H., Teng, L., Chen, W., and Shen, I.-F. (2005). Supervised learning on local tangent space. In *Advances in Neural Networks–ISNN 2005*, pages 546–551. Springer.

Liang, Y.-C., Kulturel-Konak, S., and Smith, A. E. (2002). Meta heuristics for the orienteering problem. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 1, pages 384–389. IEEE.

Ludu, A. (2016). *Boundaries of a complex world*. Springer.

Luenberger, D. G. (1972). The gradient projection method along geodesics. *Management Science*, 18(11):620–631.

Luenberger, D. G. (1973). *Introduction to linear and nonlinear programming*, volume 28. Addison-Wesley Reading, MA.

Ma, Y., Košecká, J., and Sastry, S. (2001). Optimization criteria and geometric algorithms for motion and structure estimation. *International Journal of Computer Vision*, 44(3):219–249.

Mahony, R. (1994). *Optimization algorithms on homogeneous spaces*. PhD thesis, Australian National University.

Manton, J. H. (2002). Optimization algorithms exploiting unitary constraints. *IEEE Transactions on Signal Processing*, 50(3):635–650.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979). *Multivariate Analysis*. Academic Press, London.

Mehrotra, S. (1992). On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601.

Meng, D., Leung, Y., and Xu, Z. (2011). A new quality assessment criterion for nonlinear dimensionality reduction. *Neurocomputing*, 74(6):941–948.

Mitra, D., Morrison, J. A., and Ramakrishnan, K. G. (2001). Method for resource allocation and routing in multi-service virtual private networks. US Patent 6,331,986.

Mokbel, B., Lueks, W., Gisbrecht, A., and Hammer, B. (2013). Visualizing the quality of dimensionality reduction. *Neurocomputing*, 112:109–123.

Mokotoff, E. (2001). Parallel machine scheduling problems: a survey. *Asia-Pacific Journal of Operational Research*, 18(2):193.

Mulla, A., Raravi, G., Rajasubramaniam, T., Bose, R. P. J. C., and Dasgupta, K. (2016). Efficient task allocation in services delivery organizations. In *2016 IEEE International Conference on Services Computing (SCC)*, pages 555–562.

Naimpally, S. A. and Peters, J. F. (2013). *Topology with applications*. World Scientific Publishing.

Nesterov, Y. (2013). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.

Nesterov, Y. and Nemirovski, A. (2008). Primal central paths and riemannian distances for convex sets. *Foundations of Computational Mathematics*, 8(5):533–560.

Nishimori, Y. and Akaho, S. (2005). Learning algorithms utilizing quasi-geodesic flows on the stiefel manifold. *Neurocomputing*, 67:106–135.

Niyogi, P., Smale, S., and Weinberger, S. (2008). Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1):419–441.

Otter, N., Porter, M. A., Tillmann, U., Grindrod, P., and Harrington, H. A. (2017). A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1):17.

Paul, R., Bose, R. P. J. C., Chalup, S. K., and Rarav, G. (2017). Improving operational performance in service delivery organizations through a metaheuristic task allocation algorithm. In *BPM2017, 15th International Conference on Business Process Management*.

Paul, R. and Chalup, S. K. (2017). A study on validating non-linear dimensionality reduction using persistent homology. *Pattern Recognition Letters*, 100:160–166.

Polak, E. (2012). *Optimization: algorithms and consistent approximations*, volume 124. Springer Science & Business Media.

Powell, G. E., Lane, M. T., and Indseth, R. (1999). Method and system for allocating personnel and resources to efficiently complete diverse work assignments. US Patent App. 09/294,251.

Price, C. C. (1982). Task allocation in distributed systems: A survey of practical strategies. In *Proceedings of the ACM'82 conference*, pages 176–181. ACM.

Prot, D., Lapegue, T., and Bellenguez-Morineau, O. (2015). A two-phase method for the shift design and personnel task scheduling problem with equity objective. *International Journal of Production Research*, 0:1–13.

Rieck, B. and Leitte, H. (2015). Persistent homology for the evaluation of dimensionality reduction schemes. In *Computer Graphics Forum*, volume 34, pages 431–440. Wiley Online Library.

Ring, W. and Wirth, B. (2012). Optimization methods on riemannian manifolds and their application to shape space. *SIAM Journal on Optimization*, 22(2):596–627.

Roweis, S. T. (1998). Em algorithms for pca and spca. In *Advances in Neural Information Processing Systems*, pages 626–632.

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

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

Saul, L. K., Weinberger, K. Q., Ham, J. H., Sha, F., and Lee, D. D. (2006). Spectral methods for dimensionality reduction. In Chapelle, O., Schölkopf, B., and Zien, A., editors, *Semi-Supervised learning*, pages 293–308. MIT Press.

Smet, P., Wauters, T., and Vanden Berghe, G. (2013). Hardness analysis and a new approach for the shift minimisation personnel task scheduling problem. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 1. the 27th Annual Conference of the Belgian Operations Research Society (ORBEL).

Smith, S. T. (1994). Optimization techniques on riemannian manifolds. *Fields Institute Communications*, 3(3):113–135.

Smith, S. T. (2013). Geometric optimization methods for adaptive filtering. *arXiv preprint arXiv:1305.1886*.

Staal, J., Abràmoff, M. D., Niemeijer, M., Viergever, M. A., and Van Ginneken, B. (2004). Ridge-based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 23(4):501–509.

Su, T. (2002). Four-dimensional resource allocation system. US Patent App. 10/065,341.

Taherishargh, M., Belova, I., Murch, G., and Fiedler, T. (2017). The effect of particle shape on mechanical properties of perlite/metal syntactic foam. *Journal of Alloys and Compounds*, 693:55–60.

Tausz, A., Vejdemo-Johansson, M., and Adams, H. (2014a). JavaPlex: A research software package for persistent (co)homology. In Hong, H. and Yap, C., editors, *Proceedings of ICMS 2014*, Lecture Notes in Computer Science 8592, pages 129–136.

Tausz, A., Vejdemo-Johansson, M., and Adams, H. (2014b). JavaPlex: A research software package for persistent (co)homology. In Hong, H. and Yap, C., editors, *Proceedings of ICMS 2014*, Lecture Notes in Computer Science 8592, pages 129–136.

Tausz, A. P. (2012). *Extensions and Applications of Persistence Based Algorithms in Computational Topology*. PhD thesis, Stanford University.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Udriste, C. (1994). *Convex functions and optimization methods on Riemannian manifolds*, volume 297. Springer Science & Business Media.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85.

van der Maaten, L. J. P., Postma, E. O., and van den Herik, H. J. (2009). Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10(1-41):66–71.

Vazirani, V. V. (2013). *Approximation algorithms*. Springer Science & Business Media.

Venna, J., Peltonen, J., Nybo, K., Aidos, H., and Kaski, S. (2010). Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *The Journal of Machine Learning Research*, 11:451–490.

Ventura, C., Pont-Tuset, J., Caelles, S., Maninis, K.-K., and Van Gool, L. (2017). Iterative deep learning for network topology extraction. *arXiv preprint arXiv:1712.01217*.

Weinberger, K. Q., Sha, F., and Saul, L. K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the Twenty-first International Conference on Machine Learning*, page 106. ACM.

Whitney, H. (1936). Differentiable manifolds. *Annals of Mathematics*, pages 645–680.

Wiese, A., Bonifaci, V., and Baruah, S. (2013). Partitioned edf scheduling on a few types of unrelated multiprocessors. *Real-Time Systems*, 49(2):219–238.

Zhang, P., Ren, Y., and Zhang, B. (2012). A new embedding quality assessment method for manifold learning. *Neurocomputing*, 97:251–266.

Zhao, G. (2010). Representing the space of linear programs as the grassmann manifold. *Mathematical Programming*, 121(2):353–386.

Zhou, H., Ji, Y., and Zhao, B. (2015). Tabu-search-based metaheuristic resource-allocation algorithm for svc multicast over wireless relay networks. *IEEE Transactions on Vehicular Technology*, 64(1):236–247.