Reconstruction and Denoising of Low Dimensional Manifolds from High Dimensional Scattered Data

by

Alexandra (Shira) Golovin

A thesis submitted for the degree of Doctor of Philosophy

April 29, 2021

School of Mathematical Sciences

Tel Aviv University

Israel

Contents

Abstract

The dissemination of high-end technologies paved the way for fast and effortless data acquisition in high-dimension. For example, the fields which produce such data are color, hyperspectral, and medical imaging, different biological applications, as well as softer sciences (e.g., Humanities) and lately, with the rise of language processing, also in textual data, are the just the tip of the iceberg.

In real-life applications, data usually lie in high-dimensional space and contain noise and outliers. Real-life scenarios are challenging because frequently the geometry of the data is unknown, samples may be missing, and a noise model is also missing. As a result, mining raw data, is a challenging task, and the denoising and reconstruction steps are crucial prior to further analysis.

In many cases, the high-dimensional dataset to be processed resides near a low-dimensional manifold, and this information can be exploited to improve data processing and analyses. A common practice of dealing with high-dimensional data is to use dimensionality reduction. The motivation often stems from the need to analyze, process, and visualize high-dimensional data. However, there are several fundamental challenges that may hamper the results of the analysis performed (e.g., the geometry of the data and the intrinsic dimension are usually unknown).

In this thesis, we propose a framework that addresses various approximation tasks related to noisy high-dimensional data without having to project the data to a lower-dimensional space. Our work sheds light on several important problems which arise while dealing with noisy high-dimensional data with outliers. Specifically, we deal with the following questions:

1. Denoising and reconstructing a low-dimensional manifold in a high-dimensional space [57].
2. Recovering missing information by manifold repairing [58].
3. Approximating functions on a high-dimensional manifold [55].

4. Introducing the research topic of Manifold Compressed Sensing, and presenting an efficient solution [56].

We demonstrate the effectiveness of our approaches by considering different manifold topologies and various amounts of noise, including a case of a "manifold" of different dimensions at different locations.

We begin by addressing the question of denoising and reconstructing low-dimensional manifolds in high-dimensional spaces. We suggest a multidimensional extension of the Locally Optimal Projection algorithm that was introduced by Lipman et al. in 2007 [90] for surface reconstruction in 3D. The method bypasses the curse of dimensionality and avoids the need for dimension reduction. It is based on a non-convex optimization problem, which leverages a generalization of the outlier-robust L_1-median to higher dimensions while generating noise-free quasi-uniformly distributed points reconstructing the unknown low-dimensional manifold. We develop a new algorithm, called Manifold Locally Optimal Projection (MLOP). Thus, given a noisy point-set $P = \{p_j\}_{j=1}^{J} \subset \mathbb{R}^n$ situated near a manifold $\mathcal{M}$ in $\mathbb{R}^n$, of unknown intrinsic dimension $d \ll n$, we look for a new point-set $Q = \{q_i\}_{i=1}^{I} \subset \mathbb{R}^n$ which will serve as a noise-free approximation of $\mathcal{M}$. Under some mild assumptions, we prove that the proposed algorithm converges to a local stationary solution with a bounded linear rate of convergence in case the starting point is close enough to the local minimum. In addition, we show that its order of approximation is $O(h^2)$, where h is the representative distance between P-points.

Next, we turn to the problem of recovering missing information in the presence of holes in the data. The problem of hole filling can be formulated as follows: Given noisy data sampled from a manifold with holes, the task is to generate new points that will reconstruct the missing information within the holes. While in low-dimensions the problem was studied extensively with, in high-dimension manifold repairing is still an open problem. We introduce a new approach, based on the MLOP method, to cope with manifold repairing in low and high-dimensional cases. We prove the validity of the proposed method.

Subsequently, we consider the problem of approximation of functions on a manifold in high dimension. The proposed method leverages the advantages of MLOP and the strengths of

the Radial Basis Function (RBF) [47]. We show that its approximation order is $O(C_1 h^2 + C_2 h_2^k)$, were C_1 and C_2 are constants, h is the representative distance between points, and the approximated function has k derivatives with finite L^p norm.

We conclude our work on high-dimensional data by extrapolating the definition of Compressed Sensing, from a single signal to a signal which is a manifold. We consider the fundamental problem of sensing data from a manifold and using it as a basis for recovering the manifold from a limited set of measurements. In this setting, we recover a manifold with a given density from sparse observations, sensed from the manifold with noise.

Mathematics Subject Classification: 65D99, 53B20

Acknowledgments

Undertaking this Ph.D. has been a truly scientific journey for me, a journey that would not have been possible it without the support, guidance, and collaboration of many people that I was lucky to experience. Each and every person I have in mind and heart influenced the way my research skills matured during these last years.

First and formost, I would like to express my sincere gratitude to Prof. David Levin who guided me during the Ph.D. process. I was lucky to do my M.Sc. as well as my Ph.D. under the supervision of Prof. Levin, who not only performs innovative, interesting, and unique research, but also is the most modest person I know. It has been a privilege to work with him. I would also thank him for giving me the freedom to deal with various research questions, as well as for his dedicated support and guidance and especially for his confidence in me. Prof. Levin continuously provided encouragement and was always willing to assist in any way he could throughout the research project.

My deep gratitude goes also to my second advisor, Prof. Yoel Shkolnisky, for his innovative ideas, scientific advice, insightful discussions, and overall support during my Ph.D. process.

I will forever be thankful to Prof. Israel Finkelstein, and Prof. Eliezer Piasetzky for the scientific journey in the fascinating world of First-Temple Period writings. These have been ten years of a daily roller coaster ride of playing an Indiana Jones role, where you know how each day would start, but you never know what will you discover at its end. I would like to thank Prof. Finkelstein for his patience during the research endeavor, and for his multi-faceted role model, showing how to be passionate about your work along many years of research, and how quality and quantity goes together in research. I would also like to thank Prof. Piasetzky

for the valuable discussions and for his creative and unusual ideas, which resulted in valuable and unexpected outcomes, and for teaching me to be not afraid to ask questions, and always search for answers. I am also grateful to Prof. Eli Turkel for his advice and help during this fascinating journey.

I am also pleased to thank my dearest friends who shared with me this roller coaster ride, Dr. Barak Sober, and Dr. Arie Shaus. Thank you for pursuing the dream with me, not compromising, and being there in the hard moments (and yes, for the tea breaks, too). I would also like to express my sincere thanks to Dr. Eythan Levy and Dr. Anat Mendel-Geberovich for the interesting collaboration work, and to Michael Cordonsky for assisting with the multispectral photography of ancient texts. A special thank you goes to Dr. Yariv Aizenbud for stimulating discussions and encouragement.

I would like to thank the Israel Science Foundation Grant no. 1457/13 and 2062/18 for supporting my interdisciplinary research. I am also grateful for the generous support from Jacques Chahine (made through the French Friends of Tel Aviv University).

In the end, I am grateful to my family who supported me during this long journey. Special thanks go to my grandmother, Maria, who supported me and believed in me from day one during this journey. I also thank my parents, Luba and Misha for their endless love, support, and encouragement. I would also like to express my sincere thanks to my sister Dr. Raya Romm for moral support, encouragement, and motivation to accomplish any personal goals. This thesis work is dedicated to my husband, Alex, provided support and encouragement during the challenges of a Ph.D. I am truly thankful for having him in my life. Last but by far not least, this work is also dedicated to my son Reuven Yakov, who taught me that there is amaizing life beyond scatter data and the null hypothesis.

Chapter 1

Introduction

The technological advances of the last era paved the way towards effortless and fast, high-dimensional data acquisition. Among the fields in which high-dimensional data are daily generated are, biology, remote sensing, medicine, audio, texts, and an endless array of image processing applications. In addition, with computerization becoming more widespread, many fields from both soft and hard sciences became more and more open to integrating tools which will aid in solving big questions as well as helping with the daily life routine.

The dimensionality of the data is unfortunately a curse rather than a blessing since usually, the number of samples is not sufficiently large with respect to the data dimension. A common way of dealing with high-dimensional data is to use dimensionality reduction. The motivation often stems from the need to analyze, process, and visualize high-dimensional data. Over the years many dimensionality reduction techniques were developed (for a discussion of methods for dimensionality reduction, see Section 3.1). However, dare should be exercised when performing dimensionality reduction, since meaningful information can be lost in the process.

One cannot discuss high-dimensional data without addressing the eighth wonder of the world — Machine Learning and especially Neural Networks (NN). The valuable contribution of NN to addressing problems that were hard to solve analytically or algorithmically until now, can hardly be estimated, and many new NN applications are being developed. However, there are still many unsolved challenges to be settled, on the way to the development and application

of efficient NN methods, such as efficient data collection, training, model explainability, as well as model memory requirements. In addition, despite the enormous effort to close the theoretical gaps of NN with works like [4, 70, 80, 102, 106], there are still many open questions related to this prominent technology. As will be discussed in this thesis, dealing with high-dimensional data is not limited to NN algorithms.

The present thesis deals with various challenges posed by noisy high-dimensional data. The enormous amounts of real-life data, which are collected nowadays thanks to technological advances, pose many challenges due to the data dimension as well as to the presence of noise. Thus, a crucial step prior to mining the data is denoising it. Cleaning and reconstructing the data is a challenging task since usually hardly any information is available about the noise model or about its magnitude. In addition, frequently the data are sampled non-uniformly, parts of the data may be missing and outliers may be present, and also prior knowledge regarding the geometry of the data may be not available.

The dimension of the data challenges classical approximation tools, initially developed for surfaces, to adapt to high-dimensional data. For instance, given a uniform sampling in $\mathbb{R}^n$ on a grid with spacing $h = 1/L$ requires L^n samples, which is already challenging for $n > 10$. Moreover, classical approximation methods assume smoothness of order s, which is closely related to the order of approximation error. As a result, in the high-dimensional case, the reconstruction of the data still requires devoting special attention to the problems of denoising and reconstructing the manifold carrying the data.

The common assumption that the dataset resides on a low-dimensional manifold can be exploited in order to improve the approximation. Although we can attempt to deal with the curse of dimensionality by dimension reduction, this may result in information loss. In this thesis, we propose a framework that will tackle various approximation tasks concerning noisy high-dimensional data without having to reduce dimension by passing to a lower-dimensional space. We suggest a multidimensional extension of the Locally Optimal Projection algorithm, originally introduced by Lipman et al. in [90] for surface reconstruction in 3D. The Manifold Locally Optimal Projection (MLOP) method presented in this thesis is parametrization free, it assumes no knowledge about the Intrinsic Dimension (ID) of the manifold, and it can handle

high levels of noise as well as outliers (see Chapter 3). The method allows denoising the data and reconstructing the manifold in a high-dimensional space.

Laid in mathematical terms, given a noisy point-set $P = \{p_j\}_{j=1}^{J} \subset \mathbb{R}^n$ situated near a manifold $\mathcal{M}$ in $\mathbb{R}^n$, of unknown intrinsic dimension $d \ll n$, we wish to find a new point-set $Q = \{q_i\}_{i=1}^{I} \subset \mathbb{R}^n$ that will serve as a noise-free approximation of $\mathcal{M}$. Our solution is based on a non-convex optimization problem, which leverages a generalization of the L_1-median to higher dimensions while generating quasi-uniformly distributed points in the manifold reconstruction. We prove that the method converges to a local stationary solution with a bounded linear rate of convergence provided that the starting point is close enough to the local minimum. We further prove that the approximation order is $O(h^2)$, where h is closely related to the fill-distance of the data points.

Next, we turn to the problem of recovering missing information in the presence of holes in the data. Suppose that we observe a point-cloud sampled non-uniformly from a manifold, with noise, and there are 'holes' in the data. Can we recover missing information inside the holes? In Chapter 4 we introduce a new approach, Repairing Manifold Locally Optimal Projection (R-MLOP), which expands the MLOP method to cope with manifold repairing in low and high-dimensional cases. We prove the validity of the proposed method.

Subsequently, in Chapter 5 we consider the fundamental problem of approximation of function on a low-dimensional manifold embedded in high-dimensional space, with noise present in both the data and in the function values. Due to the curse of dimensionality as well as the presence of noise, the classical approximation methods applicable in low-dimensional have a hard time tackling the high-dimensional case. We propose a new method that leverages the advantages of MLOP and the strengths of the Radial Basis Function [47] for the approximation task. The method is parametrization free, does not require any knowledge of the manifold's intrinsic dimension, it can handle noise present both in the values of the function as well as noise in the location data, and is applied directly in the high-dimensional space. We show that the complexity of the method is linear in the manifold dimension, and square logarithmic in the function codomain. We show that its approximation order is $O(h^2)$ where h is related to the fill-distance of the initial data set.

We conclude our work on high-dimensional data by extrapolating the definition of Compressed Sensing from a single signal to a signal that is represented by a manifold. In the case of high-dimensional data, a common assumption is that the data lie on a low-dimensional manifold. Depending on the intendent application, the amount of data can be a challenge. The amount of available data can be a blessing or a curse: a vast amount of data may result in computational and memory load, while limited data may only partly represent the manifold. In Chapter 6, we extrapolate the problem of compressed sensing of a single signal to manifold representation. While compressed sensing of one signal received a lot of attention, manifold compressed sensing is still sn open and non-trivial problem in high-dimensional space. In this chapter, we consider the fundamental problem of sensing data from a manifold and using the result it as a basis for recovering the manifold from a limited set of measurements. In this setting, we propose a method that recovers the manifold with the desired density from sparsely sensed noisy observations.

Throughout this thesis, we use the term noise-free reconstruction to refer to an approximation of the manifold without any noise as close as possible. The noise models that we use in the numerical experiments are with uniform noise with various magnitudes, however other models of noise exist (e.g. Gaussian).

To summarize, the main contribution of this thesis is the establishment of the MLOP framework in order to address high-dimensional, noisy data. This methodology is the pillar stone for other solutions related to high-dimensional data such as Manifold Repairing (Chapter 4), Approximation of Functions (Chapter 5), and Manifold Compressed Sensing (Chapter 6).

During my Ph.D., I had the privilege, in addition to dealing with the theoretical aspects of high-dimensional data, to encounter a real-life application of such data. My Ph.D. research can be roughly divided into two tracks. The first track addressed the problem of denoising and reconstructing a manifold in high-dimensional space (this will be extensively described in this thesis). The second track focused on high-dimensional data stemming from the First-Temple period inscriptions. The challenge in this track was to develop and implement methodologies which can tackle real-life questions, specifically, research questions posed by archaeology experts. This research taught me how to tackle such a real-life questions, starting with expressing

the research question in mathematical terms, continuing with confronting new and unfamiliar fields in applied mathematics and statistics, and ending with developing efficient tools that can answer these historical questions. My research devoted to ancient writing is summarized in 18 published papers [50–53, 59–64, 67, 94, 95, 110–113, 117]. This research work opened my eyes to many other research directions, such as image processing, pattern recognition, and statistical analysis. In addition, I had a unique opportunity to work on high-dimensional data stemming from a remote sensing laboratory, which resulted in the valuable publication [136].

Chapter 2

Surface Reconstruction — Preliminaries

2.1 Introduction

Prior to the Big Data era, classical approximation mainly addressed surface reconstruction. However, a smaller amount of data in low-dimensions does not necessarily imply that the problem of surface reconstruction is easy. The problem of surface reconstruction can be viewed as follows: Given a set of points sampled from a surface, the task is then to construct a surface representation faithful to the unordered collection of points. There is a wide range of applications to surface reconstruction starting with CAD design (e.g., in industrial applications, like vehicle design), continuing with computer graphics and animation movies (e.g., Pixar), and even computerized teeth reconstructions.

Surface reconstruction has been active field in computational geometry since the 1980s [19], although the analytic framework is newer. The problem of low-dimensional reconstruction was thoroughly studied over the years [3, 13, 35, 84, 90]. However, there are still many challenges which modern applications pose, among them the need for preservation of sharp features preservation, reconstruction from uniform/non-uniform data sampling, undersampling, fast-changing curvature [54], adjacent surface segments, and the presence of noise. For example, usually reconstruction methods smooth out the existing corners and sharp features that

exist on the surface. The sharp features preservation challenge has been dealt with extensively (e.g., [77, 135]).

The available methods commonly assume almost noise-free data and rely on normal estimation. Unfortunately, in real-life cases, noise is often present, and normal estimation may not be robust enough (despite various processes for cleaning the normals). One method which offers a solution to handle high levels of noise is the parameterization-free projection (LOP) method proposed in [90]. This method does not require knowledge of a surface parameterization, avoids using local surface approximation, and normal estimation is cheap and can be parallelized due to its local support. In [90], it was demonstrated by various examples that the method is stable with respect to outliers, the different density of sampling, and varying topology. In the next section, we provide a detailed discussion of the method.

2.2 Locally Optimal Projection

In this section we will review the LOP method introduced in [90]. Let S be a surface in $\mathbb{R}^3$. Given the data point-set $P = \{p_j\}_{j\in J} \subset \mathbb{R}^3$ sampled from S with noise and outliers, LOP projects an arbitrary point-set $X(0) = \{x_i{}^{(0)}\}_{i\in I} \subset \mathbb{R}^3$ back onto the set P, in such a way that it approximates the original surface S. The set of projected points $Q = \{q_i\}_{i\in I} \subset \mathbb{R}^3$ is required to minimize the sum of the weighted distances to the points of P, with respect to radial weights centered at the same set of points Q. Furthermore, the points in Q should not be too close to each other. These requirements lead us to defining the desired set of point Q as the fixed-point solution of the equation

$$Q = G(Q), \tag{2.1}$$

where

$$G(C) = \text{argmin}_{X=\{x_i\}_{i\in I}}\{E_1(X, P, C) + \Lambda E_2(X, C)\}, \tag{2.2}$$

$$E_1(X, P, C) = \sum_{i\in I}\sum_{j\in J} \|x_i - p_j\|\theta(\|c_i - p_j\|) \tag{2.3}$$

and

$$E_2(X, C) = \sum_{i' \in I} \lambda_{i'} \sum_{i \in I \setminus \{i'\}} \eta(\|x_{i'} - c_i\|)\theta(\|c_{i'} - c_i\|\,. \tag{2.4}$$

Here θ is a rapidly-decreasing smooth weight function with compact support radius h defining the size of the influence radius, η is another decreasing function penalizing the points $x_{i'}$ that get too close to other points, and $\{\lambda_i\}_{i \in I}$ are balancing terms, the set of which we denote by Λ. The original LOP paper [90], used $\theta(r) = e^{-\frac{r^2}{h^2}}$ and $\eta(r) = \frac{1}{3r^3}$.

The motivation behind the two terms E_1 and E_2 terms appearing in (2.2) is as follows: E_1 drives the projected points Q to approximate the geometry of P, while E_2 strives at keeping the distribution of the points Q uniform. In particular, the term E_1 was inspired by the L_1-median, where, given a point-set $P = \{p_j\}_{j \in J}$, the L_1-median is defined as the point q which minimize the sum of the Euclidean distances,

$$q = \text{argmin}_x \{\sum_{j \in J} \|p_i - x\|\}\,. \tag{2.5}$$

The L_1-median was introduced as a solution to the challenge of service center location, considered by Weber in 1909 [128]. The industrial problem is to find the appropriate location for a warehouse that will service J customers represented by J points $p_1, ..., p_J$ in the plane. We idealize the situation by supposing that the transportation costs for deliveries from the warehouse to customers are proportional to the Euclidean distance. The proposed solution to the location problem suggested by Weber is to locate the warehouse so as to minimize the sum of the transportation costs to all customers. In statistics, the problem is known as the L_1-median [20, 115, 129].

An important advantage of the L_1-median is its breakdown point. The breakdown of an estimator is useful in understanding its robustness, especially in real-life scenarios where noise is present. Therefore, it is important to know what proportion of the data can be contaminated without affecting considerably the estimator. The L_1-median is a resistant statistic and its breakdown point is 0.5 [91], i.e., it fails only if more than 50% of the data points are contaminated, and this is one of the main advantage of it over averaging. If the data points are not collinear, the solution to problem (2.5) is unique [96,129] (the proof relays on the convexity

property of the L_1-median).

Accordingly, E_1 term in (2.2) is a generalization of the L_1-median to the local k-L_1-medians. Specifically, we look for the location of k service centers that will minimize the sum of samples belonging to the support of x_i as $\sum_{i\in I}\sum_{j\in J}\|x_i - p_j\|\theta(\|c_i - p_j\|)$, where the locality property is defined by h, the support of θ. As noted before, the E_2 term is responsible for the dissemination of the sample points on the manifold, with the points that are too close being penalized by the η function.

In contrast to the L_1-median, which is a convex function, the LOP function G is non-convex. The reason is that the second derivative of the $f(r) = re^{-r^2}$, where $r \geq 0$, is $f''(r) = -re^{-r^2}(6+4r^2) \leq 0$, and it vanishes only when $r = 0$, i.e., when the distance between two points of which r is their distance equal. As we will see in the following section, the non-convexity of G will introduce many difficulties in establishing the convergence, rate of convergence, and uniqueness of the solution.

The solution to the fixed point problem (2.2) is found by an iterative process. Let $\mu \in [0, 1/2]$ be the coefficient of the balancing term $\lambda = \mu\frac{\sum_{j\in J}\alpha_j^{i'}}{\sum_{i\in I\setminus\{i'\}}\beta_i^{i'}}$ in (2.4), and let $X^{(1)} = \{x_i^{(1)}\}_{i\in I}$ be defined as

$$x_{i'}^{(1)} = \frac{\sum_{j\in J} p_j\theta(\|p_j - x_{i'}^{(0)}\|)}{\sum_{j\in J}\theta(\|p_j - x_{i'}^{(0)}\|)}, \quad i' \in I\,. \tag{2.6}$$

The fixed point iterations are defined by

$$x_{i'}^{(k+1)} = \sum_{j\in J} p_j\frac{\alpha_j^{i'}}{\sum_{j\in J}\alpha_j^{i'}} + \mu\sum_{i\in I\setminus\{i'\}}(x_{i'}^{(k)} - x_i^{(k)})\frac{\beta_i^{i'}}{\sum_{i\in I\setminus\{i'\}}\beta_i^{i'}}, \quad i' \in I\,, \tag{2.7}$$

where $\alpha_j^{i'} = \dfrac{\theta(\|p_j - x_{i'}^{(k)}\|}{\|p_j - x_{i'}^{(k)}\|}$, $\quad \beta_i^{i'} = \dfrac{\theta(\|x_{i'}^{(k)} - x_i^{(k)}\|}{\|x_{i'}^{(k)} - x_i^{(k)}\|}\left|\dfrac{\partial\eta}{\partial r}(\|x_{i'}^{(k)} - x_i^{(k)}\|)\right|$.

For the sake of completeness, we mention here an important result regarding the order of approximation of the LOP approximation of a surface (see [90]). In Chapter 3 we will prove a generalization of this theorem to the case of a d-dimensional Riemannian manifold residing in $\mathbb{R}^n$.

Theorem 2.1. *Let $P = \{p_j\}_{j\in J}$ be data point sampled from a surface in $\mathbb{R}^3$. If the surface, S, is a C^2-smooth surface, the then the* LOP *operator has an order of approximation $O(h^2)$ to S, provided that Λ is carefully chosen.*

We illustrate the process pf surface reconstruction by LOP by presenting two examples. The first example demonstrates the robustness of the LOP to noise and shows its effectiveness in the case of close but separated surfaces. One can see in Figure 2.1 that the reconstruction maintains the distance between the cylinders and successfully deals with noise. The other example illustrates the role that E_1 term plays in the reconstruction, where the reconstruction maintains its proximity to the original sampled data and does not repair the holes in the data (Figure 2.2).

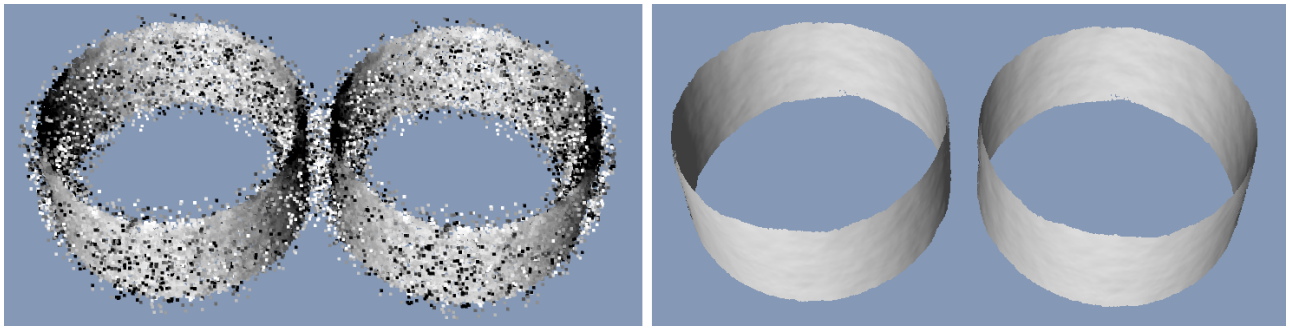


Figure 2.1: Nearly touching cylinders as considered in [90]. Left: input data. Right: LOP reconstruction.

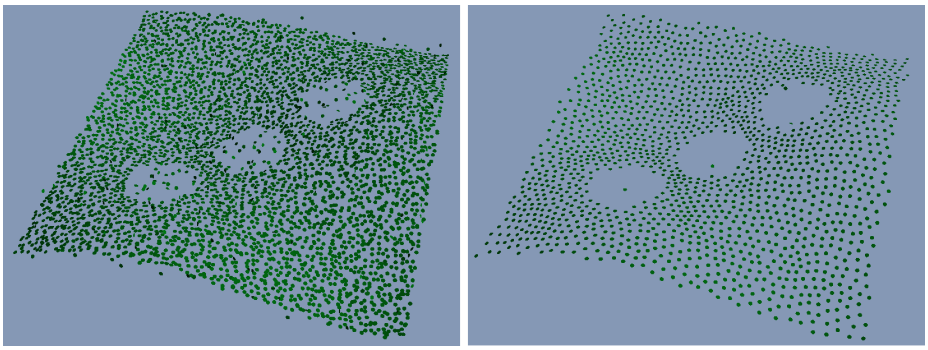


Figure 2.2: Reconstruction of data with holes as appeared in [90]. Left: a noisy point-cloud of a surface with three holes. Right: resulting LOP projections are shown in the right.

The main advantages of the LOP are apparent. Besides the fact that it can deal with high amounts of noise, the procedure does not require the estimation of local normals and planes, or

parametric representations, and can deal with non-uniform distributions. Due to its flexibility and satisfactory results, it has been extended to address other challenges related to surfaces [76, 77, 120]. In [76] the authors modified the LOP method to broaden its capability of dealing with non-uniform distributions, which are common in raw data, by incorporating locally adaptive density weights into LOP, and by using a different repulsion function η, resulting in the WLOP method. In [120] the authors propose a novel curvature-aware technique, by introducing a weight term related to surface variation at each point, and in addition, speed up the convergence of the method by introducing a new initialization process (denoted as ALOP). An additional very elegant enhancement deals with a very important challenge — the reconstruction of sharp features in surfaces. In [77] the authors alter the LOP operator and make it normal or edge-aware (EAR), allowing for resampling away from edges, with noticeable improved results.

In what follows we extend the vanilla LOP algorithm to the high-dimensional case. It is worth noting that the LOP enhancements, mentioned above, can be extended to the high-dimensional case. From the theoretical point of view, the paper [90] addresses the rate of approximation of the LOP for the surface case. However, no attention was paid to the convergence of the method and the rate of convergence, and bound was given for its complexity. In this thesis, we extend the LOP to the high-dimensional case and provide wide theoretical grounds to the high-dimensional methodology, were not available prior to this thesis. In Chapter 3 we present a formal description of the manifold reconstruction and show in Theorem 3.5 that the oder of approximation of LOP in high-dimensions is indeed $O(h^2)$. We also prove the convergence of the method to a stationary point, as well as boun the rate of convergence (Theorems 3.4, 3.7).

It is also worth mentioning that the most challenging part of the algorithm is estimating the size h of the support of the weight function θ. The support size plays a significant role in the algorithm and controls the amount of P points and Q points which will be served by a specific q_i. To some extend h is involved in balancing the proximity and the equal distribution of the reconstruction points Q. Furthermore, taking h too big can result in inaccurate results when one deals with surface/manifold with a small reach (for more details, see Remark 3.6). In Theorem 3.1 we estimate the value of h, discuss why in fact one should use two support

sizes (for P and Q, respectively). Later, we use this estimation in all the numerical examples.

Chapter 3

Manifold Reconstruction and Denoising

3.1 Introduction

High-dimensional data is increasingly available in many fields, and the problem of extracting valuable information from such data is of primal interest. Often, the data suffers from the presence of noise, outliers, and non-uniform sampling, which can influence the result of the mining task. We can address this problem by denoising a single sample, an approach extensively used in the last decades (the denoising method is often data-driven). However, it is still a challenge to produce a good noise-free result from a single sample with a large amount of noise present. Frequently, classical denoising algorithms lose the battle, since they denoise a single sample and overlook the intrinsic connections between different samples acquired from a chosen domain. As a result, obtaining a dataset of samples with certain properties can boost the denoising process. A common practice is to assume that the high-dimensional input data lies on an intrinsically low-dimensional Riemannian manifold.

For instance, with the development of image processing, the task of image denoising gained a lot of attention (see, e.g., [48, 93, 119]). Thus, given a single image, the task is to find its noise-free image. Now, let us consider a collection of noisy images depicting a single object, controlled by several parameters (such as a set of faces or written letters rotated in different

directions). This collection can be modeled by a manifold, and this representation can be utilized to produce a superior denoising result. A real-life case, which motivated the current research, is cryo-electron microscopy [114]. In this problem a single image is a projection of a three-dimensional macromolecule into a two-dimensional representation (Figure 3.1 (A)). Cryo-electron microscopy images are known to suffer from extremely low signal to noise ratio (Figure 3.1 (C)), and consequently classical denoising methods usually do not perform well on such samples. Nevertheless, using the fact that the images are sampled from a manifold (each corresponding to the molecule projected in a different direction) can facilitate the denoising task. Figure 3.1 (B) shows a collection of images, each depicting a projection of the simulated molecule in Figure 3.1 (A), captured in various directions. Thus, we transfer the problem from single image denoising to denoising the entire image set – which is treated as scattered data sampled from a manifold.

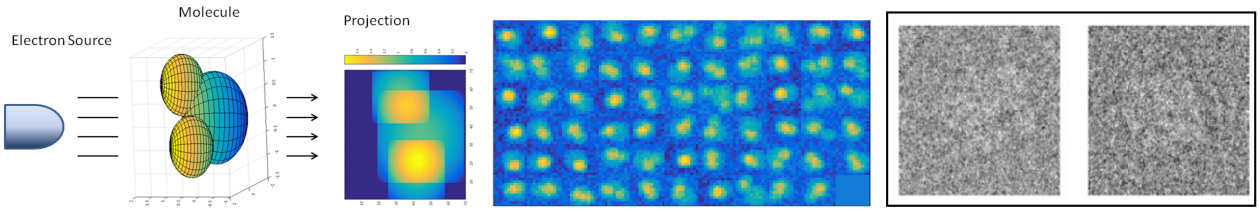


Figure 3.1: (A) Illustration of the cryo-electron microscopy projection process, where a 3D molecule is rotated and projected to 2D. (B) Collection of the artificial projections of the molecule with noise, where each image is the molecule rotated in a different direction. (C) Two real electron microscope images of the E. coli 50S ribosomal subunit (image is taken from [114]). These images demonstrate the denoising challenge of extremely low signal to noise ratio.

In this chapter, we address the problem of manifold denoising and reconstruction. Let $\mathcal{M}$ be a d-dimensional manifold in $\mathbb{R}^n$, where $d \ll n$. Suppose that the scattered data $P = \{p_j\}_{j=1}^J$ were sampled near $\mathcal{M}$ and contain noise and outliers. We wish to find a noise-free reconstruction of the geometry of $\mathcal{M}$ in $\mathbb{R}^n$.

As described in the previous section, there are various solutions that address a simpler, yet challenging problem of surface reconstruction. The application of classical approximation tools, developed for surfaces, to high-dimensional data, encounters various challenges, usually stemming from the high-dimension, and presence of noise. For instance, given a uniform

sampling in $\mathbb{R}^n$ on a grid with spacing $h = 1/L$ requires L^n samples and when $L \ll 10$ this is already challenging for $n \ll 10$. Moreover, classical approximation methods assume smoothness of order s, which is closely related to the approximation error. For example, for J sample points, the reconstruction accuracy can be of the order of $O(J^{-s/n})$, which implies that we need to increase the amount of data as the domain dimension increases [8]. As a result, in the high-dimensional case, the problem of manifold reconstruction still requires additional attention especially to the problem of denoising and reconstructing manifold.

A common way of dealing with high-dimensional data is to use dimensionality reduction. The motivation often stems from the need to analyze, process, and visualize high-dimensional data. Along the years many dimensionality reduction techniques were developed (PCA [103], Multidimensional Scaling [39], Linear Discriminant Analysis [68], Locality Preserving Projections [75], Locally Linear Embedding [107], ISOMAP [122], Diffusion Maps [36], and Neural Networks in their general form, [88], to mention just a few). However, one has to be careful when performing dimensionality reduction, since meaningful information can be lost due to the assumptions made. One fundamental challenge of dimensionality reduction is knowing or estimating the dimension of the data. In addition, since the geometry of the data is usually unknown, it is common to use an assumption regarding its geometrical structure (and use linear/non-linear algorithms accordingly). As a result, in the case of real-life data, it is still a challenge to address these issues, mainly because such assumptions have a direct influence on the usage of dimensionality reduction methods, and may, therefore, hamper the results of the analysis performed. For a comprehensive survey of manifold learning methods that rely on dimensionality reduction, see [88].

An alternative practice for handling high-dimensional data is manifold learning in high-dimensional space. Thus, instead of making assumptions on the geometry of the manifold, its intrinsic dimension and reducing the dimension of the data, the mining task is performed in a high-dimensional space. This approach has several advantages, and also disadvantages. On the one hand, there is no loss of information. On the other hand, the dimension of the data influences the efficiency and feasibility of the algorithms, and it is possible that one will not be able to see the forest for the tree. An additional important factor of high-dimensional

data is noise, which is usually present in real-life scenarios. In Table 4.1 we give a short survey of manifold reconstruction methods that avoid performing dimensionality reduction. Among the first papers that addressed the manifold reconstruction problem was [34]. The method presented therein relied on Delaunay triangulation, and as the authors themselves noted, it was impractical, mainly because it requires a very dense and noise-free sample, and also because it makes use of (weighted) Delaunay triangulation in higher dimensions. Next, in [99] it was proposed to use simplicial complexes. In that paper, the authors also address the challenge of noisy samples, under certain conditions. This work was followed by [18], which aimed at avoiding computing the Delaunay triangulation of the given set of points by using a Witness complex via an iterative process, and by [17] which addressed the problem using a Tangential Delaunay complex. Unfortunately, this method dealt only with noise-free samples. Next in [92], the authors proposed to learn a data-dependent dictionary from clean data in the chosen resolution level and use it for the manifold reconstruction of possibly noisy data. Later, in [66], it was suggested to use a covering of the manifold by discs to deal with a small amount of Gaussian noise. The recent paper [118] proposed to address manifold denoising under various noisy scenarios, when the intrinsic dimension of the manifold is known, by extending the Moving Least Squares method [84] to the high-dimensional case. Finally, the paper [1] address the problem manifold reconstruction and of tangent space and curvature estimation by using local polynomials.

Table 3.1: Survey of manifold reconstruction methods, that avoid performing dimensionality reduction

Authors	Algorithm key features	Assumptions on the data	Sampling	Handle noise	Error	Numerical exp.	Complexity, N is #points, d- is ID, n is the dim. of the ambient space
Cheng, et al. [34]	Weighted Delaunay triangulation	Compact manifold, smooth, no boundary	Sufficiently dense point sample	Noise-free sample	homeomorphic	N/A	$O(N\log(N))$
Niyogi, et al. [99]	Simplicial complex	Suffcient amount of points		Bounded/ specific models of noise	homeomorphic	N/A	N/A
Boissonnat, et al. [18]	Witness complex	Positive reach (i.e. C^1- continuous)	Not necessarily uniformly sampled, minimal local density	Low noise level	homeomorphic	N/A	$N^2d^{O(d^2)}$
Chazal, et al. [29]	Distance functions with probability distribution		Regularity of the input data	Bounded/ specific models of noise	homotopic	✓	N/A
Boissonnat, et al. [17]	Tangential Delaunay complex	Smooth manifold, positive reach	Sampling ratio, point sparsity, and the reach hold a condition	Noise-free sample	homeomorphic	N/A	$O(n)N^2 + n2^{O(d^2)}N$
Maggioni, et al. [92]	Dictionary	Smooth closed manifold, d is known	Homogeneous, reconstruct new noisy samples	Additive noise, dictionary is built from clean samples	✓	✓	$O(C^d(n+d^2)\epsilon^{-(1-\frac{d}{2})}log\frac{1}{\epsilon}+dn)$, where C is a constant, and ϵ is reconstruction error
Fefferman, et al. [66]	Disk stitching	Reach is bounded		Additive noise	✓	N/A	N/A
Sober, Levin [118]	Moving Least Squares	d is known, bounded reach		Additive noise	✓	✓	$O(d^3m + Nd^m + NI)$, I-#points in supp., m is the approx. degree
Aamari, Levrard [1]	Local Polynomials	d and order of regularity are known		Bounded/ specific models of noise	✓	N/A	N/A

The methods listed in the table provide a strong theoretical background, but most of them are not accompanied by numerical examples (except [92, 118]), which is an important aspect of evaluating the method execution. In addition, unfortunately, as can be seen from the table, handling noisy data, non-uniformly sampled, with no assumption on the data, is still a challenge in high-dimensional cases. In this thesis, we propose denoising and reconstructing

the manifold geometry in a high-dimensional space in the presence of high amounts of noise and outliers. We will tackle the manifold approximation question by extending the Locally Optimal Projection algorithm [90] to the high-dimensional case. The proposed algorithm is simple, fast and efficient, and does not require any additional assumption. Our theoretical analysis is accompanied by numerical examples of various manifolds with different amounts of noise.

3.2 High-Dimensional Denoising and Reconstruction

The Locally Optimal Projection(LOP) method, described in details in Section 2.2, was introduced in [90] to approximate two-dimensional surfaces in $\mathbb{R}^3$ from point-set data. The procedure does not rely on estimating local normal, planes, or on parametric representation. The main advantage of the method is that it performs well in the case of noisy samples.

Here we generalize the LOP mechanism to perform what we call *Manifold Locally Optimal Projection (MLOP)*. The vanilla LOP is not able to cope with high-dimensional data, mainly due to the sensitivity of the norm to noise and outliers (as will be discussed in details in subsection 3.3.1). In addition, other adaptations were needed due to practical reasons (as will be described in the end of this subsection).

First, we adapt the h-ρ condition defined for scattered-data approximation functions (in [83], defined for low-dimensional data), to handle finite discrete data on manifolds

Definition 3.1. *h-ρ **sets of fill-distance** h, **and density** $\leq \rho$ with respect to the manifold $\mathcal{M}$. Let $\mathcal{M}$ be a manifold in $\mathbb{R}^n$ and consider sets of data points sampled from $\mathcal{M}$. We say that $P = \{P_j\}_{i=1}^{J}$ is an h-ρ set if:*
1. h is the fill-distance, i.e., $h = median_{p_j \in P} \min_{p_j \in P \setminus \{p_i\}} \|p_i - p_j\|$.
2. $\#\{P \cap \bar{B}(y, kh)\} \leq \rho k^n, \quad k \geq 1, \quad y \in \mathbb{R}^n$.
Here $\#Y$ denotes the number of elements in a set Y and $\bar{B}(x, r)$ denotes the closed ball of radius r centered x.

Note that the last condition regarding the point separation δ defined in [83], which states that there $\exists \delta > 0$ such that $\|p_i - p_j\| \geq \delta, \quad 1 \leq i \leq j \leq J$, is redundant in the case of finite

data. We also note that the vanilla definition of the fill-distance uses the supremum *sup* in its expression (instead of the median). Here we use the median in order to deal with the presence of outliers.

The setting for the high-dimensional reconstruction problem is the following: Let $\mathcal{M}$ be a manifold in $\mathbb{R}^n$, of unknown intrinsic dimension $d \ll n$. One is given a noisy point-cloud $P = \{p_j\}_{j=1}^J \subset \mathbb{R}^n$ situated near the manifold $\mathcal{M}$, such that P is a h-ρ set. We wish to find a new point-set $Q = \{q_i\}_{i=1}^I \subset \mathbb{R}^n$ which will serve as a noise-free f $\mathcal{M}$. We seek a solution in the form of a new point-set Q, which will replace the given data P, provide a noise-free approximation of $\mathcal{M}$, and which is quasi-uniformly distributed. This is achieved by leveraging the well-studied weighted L_1-median [125] used in the LOP algorithm and requiring a quasi-uniform distribution of points $q_i \in Q$. These ideas are encoded by the cost function

$$G(Q) = E_1(P,Q) + \Lambda E_2(Q) = \sum_{q_i \in Q} \sum_{pj \in P} \|q_i - p_j\|_{H_\epsilon} w_{i,j} + \sum_{q_i \in Q} \lambda_i \sum_{q_{i'} \in Q \setminus \{q_i\}} \eta(\|q_i - q_i'\|) \hat{w}_{i,i'} \,, \quad (3.1)$$

where the weights $w_{i,j}$ are given by rapidly decreasing smooth functions. In our implementation we used $w_{i,j} = \exp\left\{ - \|q_i - p_j\|^2 / h_1^2 \right\}$ and $\widehat{w}_{i,i'} = \exp\left\{ - \|q_i - q_i'\|^2 / h_2^2 \right\}$. Here, we replace the L_1-norm used in [90] by the "norm" $\|\cdot\|_{H_\epsilon}$ introduced in [85] as $\|v\|_{H_\epsilon} = \sqrt{v^2 + \epsilon}$, where $\epsilon > 0$ is a fixed parameter (in our case we take $\epsilon = 0.1$). As shown in [85], using $\|\cdot\|_{H_\epsilon}$ instead of $\|\cdot\|_1$ has the advantage that one works with a smooth cost function and outliers can be removed. In addition, h_1 and h_2 are the support size parameters of $w_{i,j}$ and $\hat{w}_{i,i'}$ that guarantee a sufficient amount of P or Q points for the reconstruction. We provide additional details on how to estimate the support size, in Subsection 3.3.2. Also, $\eta(r)$ is a decreasing function such that $\eta(0) = \infty$; in our case we take $\eta(r) = \frac{1}{3r^3}$. Finally, $\{\lambda_i\}_{i=1}^I$ are constant balancing parameters.

Remark 3.1. *Note that the Huber loss can also be used in equation 3.1 for norm calculation* $\|v\|_{H_\epsilon}$.

We will now give some intuition about the definition of the cost function G. We can describe the cost function in (3.1) in terms borrowed from electromagnetism, where an electron generates an electric field that exerts an attractive force on a particle with a positive charge, such as the proton, and a repulsive force on a particle with a negative charge. In our scenario, we have

attraction forces between the Q-points and the original P-points, and repulsion forces between the Q-points to themselves in order to make them spread out in a quasi-uniform manner (Figure 3.2). An additional way of looking at the target function is to view the solution using a service center approach: placing a distribution of service centers $q_i \in Q$ to best serve the customers P, such that the service centers are spread uniformly. Thus, in case we have more points in P than in the reconstruction, each center $q_i \in Q$ will serve a certain amount of P-points in its neighborhood.

Remark 3.2. *We do not require that the amount of the points in the reconstruction (Q), and the size of the original sample set (P) be the same. This flexibility allows downsampling and upsampling in order to decode or encode manifold information.*

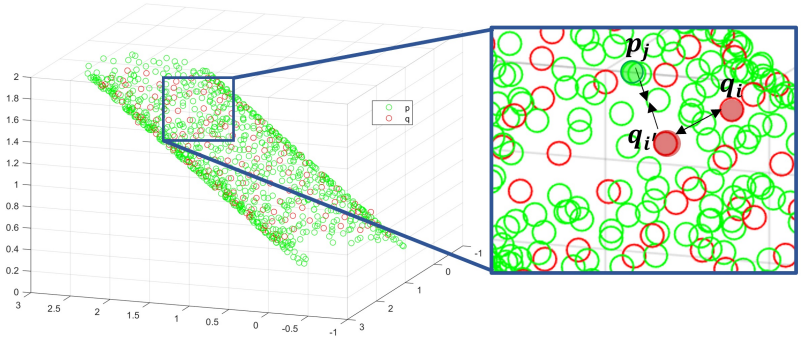


Figure 3.2: Illustration of the cost function during manifold reconstruction: each point from the reconstruction set Q (red points) is attracted to points in P (green dots), and repelled by other points in Q according to their distance.

In order to solve the problem with the cost function (3.1), we look for a point-set Q that minimizes $G(Q)$. The solution Q is found via the gradient descent iterations

$$q_{i'}^{(k+1)} = q_{i'}^{(k)} - \gamma_k \nabla G(q_{i'}^{(k)}), \qquad i' = 1, \dots, I\,, \tag{3.2}$$

where the initial guess $\{q_i^{(0)}\}_{i=1}^{I} = Q^{(0)}$ consists of points are sampled from P.

The gradient of G is given by

$$\nabla G(q_{i'}^{(k)}) = \sum_{j=1}^{J} \big(q_{i'}^{(k)} - p_j\big)\alpha_j^{i'} - \lambda_{i'} \sum_{\substack{i=1 \\ i\neq i'}}^{I} \big(q_{i'}^{(k)} - q_i^{(k)}\big)\beta_i^{i'}\,, \tag{3.3}$$

with the coefficients $\alpha_j^{i'}$ and $\beta_j^{i'}$ given by the formulas

$$\alpha_j^{i'} = \frac{w_{i,j}}{\|q_i - p_j\|_{H_\epsilon}} \left(1 - \frac{2}{h_1^2}\|q_i - p_j\|_{H_\epsilon}^2\right) \tag{3.4}$$

and

$$\beta_i^{i'} = \frac{\widehat{w}_{i,i'}}{\|q_i - q_{i'}\|} \left(\left|\frac{\partial \eta\left(\|q_i - q_{i'}\|\right)}{\partial r}\right| + \frac{2\eta\left(\|q_i - q_{i'}\|\right)}{h_2^2}\|q_i - q_{i'}\|\right), \tag{3.5}$$

for $i = 1, ..., I,\ i \neq i'$. In order to balance the two terms in $\nabla G(q_{i'}^{(k)})$, the factors $\lambda_{i'}$ are initialized in the first iteration as

$$\lambda_{i'} = -\frac{\left\| \sum_{j=1}^{J} \left(q_{i'}^{(k)} - p_j\right)\alpha_j^{i'} \right\|}{\left\| \sum_{i=1}^{I} \left(q_{i'}^{(k)} - q_i^{(k)}\right)\beta_i^{i'} \right\|}. \tag{3.6}$$

Balancing the contribution of the two terms is important in order to maintain equal influence of the attraction and repulsion forces in $G(Q)$. The step size in the direction of the gradient γ_k is calculated following the procedure suggested by Barzilai and Borwein in [11], as

$$\gamma_k = \frac{\langle \triangle q_{i'}^{(k)}, \triangle G_{i'}^{(k)} \rangle}{\langle \triangle G_{i'}^{(k)}, \triangle G_{i'}^{(k)} \rangle}, \tag{3.7}$$

where $\triangle q_{i'}^{(k)} = q_{i'}^{(k)} - q_{i'}^{(k-1)}$ and $\triangle G_{i'}^{(k)} = \nabla G_{i'}^{(k)} - \nabla G_{i'}^{(k-1)}$.

The main idea of Barzilai and Borwein's approach is to use the information of the previous iteration to decide the step-size in the current iteration. In [11] it is proven that for a convex function the gradient descent with this step size given by 3.7 converges R-superlinearly and R-order is $\sqrt{2}$. The intuition behind definition 3.7 is that the step size is chosen in such a way, that $\gamma_k I$ mimics the inverse Hessian of $\nabla^2 f(x)$. Specifically, using our notations of $\triangle q_{i'}^{(k)}$ and $\triangle G_{i'}^{(k)}$ we require that $\gamma_k \triangle q_{i'}^{(k)} \approx \triangle G_{i'}^{(k)}$ in the least-squares sense. Solving this problem results in equation 3.7 equation [132].

The reconstruction process is summarized in Algorithm 1 below:

Algorithm 1 MLOP: Iterative Manifold Reconstruction

Input: $P = \{p_j\}_{j=1}^{J} \subset \mathbb{R}^n$, $\epsilon > 0$

Output: $Q = \{q_i\}_{i=1}^{I} \subset \mathbb{R}^n$

Initialize $Q^{(0)}$ as a subsample of P

Estimate h_1 and h_2

repeat

  for each $q_{i'}^{(k)} \in Q^{(k)}$ **do**

    Calculate $\nabla G(q_{i'}^{(k)})$ by assessing $\alpha_j^{i'}$, $\beta_i^{i'}$

    $q_{i'}^{(k+1)} = q_{i'}^{(k)} - \gamma_k \nabla G(q_{i'}^{(k)})$

  end for

until $\|\nabla G(q_{i'}^{(k)})\| < \epsilon$

Naturally, several changes were made to the LOP algorithm when shifting from the low-dimension to high-dimensional case. The **main enhancements of the LOP algorithm which were introduced in MLOP for high-dimensional space** can be summarized in the following list:

1. The problem is reformulated in terms of looking for a new set Q which will maintain the conditions in (3.1). This change is taken into account when taking the derivatives.
2. The L_1 norm used in E_1 is replaced with the H_ϵ, defined in [85] as $\|v\|_{H_\epsilon} = \sqrt{v^2 + \epsilon}$, where $\epsilon > 0$ is a fixed parameter. The motivation behind this is to have a "norm" which is less sensitive to outliers. Instead of squares of errors or the absolute values of the errors, we will use an error measure that behaves as squared error for small errors and as an absolute error if the error is large. Please note that we change the norm only in the first term in (3.1) to cope with the outliers in P.
3. The norm calculation is modified to cope with high-dimensional data with noise, by using the sketching technique. For more details see Section (3.3.1).
4. From practical reasons, we replace the fixed point iterations used in [90], with a gradient descent. The motivation behind it was to use a methodology that will allow easier theoretical analysis of the already challenging non-convex function G.
5. A new definition for the balancing terms λ_i is suggested, such that the λ_i does not change

along the iterations (and there is no need to take the their derivatives).

6. Different support sizes are used when looking at the support of a given point q_i with respect to P and with respect to Q. This is natural when the number of points in P and Q differ. For additional details how to estimate these parameters see Section (3.3.2).

3.3 Practical Details

In Section 3.2 we introduced the method for high-dimensional denoising and reconstruction, by optimizing a cost function that leverages the proximity to the original data and asks for quasi-uniform reconstruction. In the following two subs-sections, we will discuss several practical aspects related to robust high-dimensional distance calculation, as well as the optimal selection of the support of the weight function $w_{i,j}$.

3.3.1 Robust Distance Calculation in High Dimensions

The reasoning in terms of Euclidean distances, which is the cornerstone of Algorithm 1, works well in low dimensions, e.g., for the reconstruction of surfaces in 3D, but breaks down in high dimensions once noise is present. For example, consider three points A, B and C in $\mathbb{R}^2$ (Figure 3.3 (A)), where the points A and B are close, whereas the point C is far. Next, we embed these points in to $\mathbb{R}^{60}$ with a uniformly additive noise distribution $U(-0.2, 0.2)$ (for example in Figure 3.3 (D) we plot one of the points in $\mathbb{R}^{60}$). Unfortunately, the noise completely wipes out the signal and as a result far points cannot be distinguished from adjacent ones, see Figure 3.3 (B) (see [2, 44]).

To deal with this issue, we perform dimension reduction via random linear sketching [131]. It should be emphasized that the dimension reduction procedure is utilized solely for the calculation of norms, and the manifold reconstruction is performed in the high-dimensional space. Given a point $x \in \mathbb{R}^n$, we project it to a lower dimension $m \ll n$ using a random matrix, S, with certain properties (its construction is described in detail in Algorithm 2). Subsequently, the norm of $\|S^t x\|$ will approximate $\|x\|$. Figure 3.3 (C) shows that calculating the distance in lower-dimensional space solves the distance conflicts.

In Algorithm 2 we present the details of finding the matrix $S \in \mathbb{R}^{n\times m}$. For given scattered data points $P = \{p_j\}_{j=1}^{J} \subset \mathbb{R}^n$ we construct matrix S only once during the initialization process of Algorithm 1. Next, given a new point $x \in \mathbb{R}^n$, its norm is approximated as $\|S^t x\|$ and utilized only for the gradient calculations in (3.3). In this paper, we choose to perform a global linear projection. However, for additional accuracy, it is possible to find a local transformation for each neighborhood.

Algorithm 2 Robust Distance Calculation in High Dimensions

1: **Input:** $P = \{p_j\}_{j=1}^{J} \subset \mathbb{R}^n$, m

2: **Output:** S - an $n \times m$ matrix

3: Sample $G \in \mathbb{R}^{J\times m}$ with $G \sim N(0, 1)$.

4: Compute $B \in \mathbb{R}^{n\times m}$ as $B := P^{\mathrm{t}}G$.

5: Calculate the QR decomposition of B as $B = SR$, where $S \in \mathbb{R}^{n\times m}$ has orthonormal columns and $R \in \mathbb{R}^{m\times m}$ is upper triangular.

Remark 3.3. *How should we choose the dimension* m *of the space on which we project the data? First, if the dimension of the manifold $\mathcal{M}$ is known, this information can be utilized for setting* m*. Alternatively, one can calculate a rough estimate, or apply a local* PCA*, and use the number of the dominant eigenvalues. In our examples, the typical size of* m *was set to* 10*.*

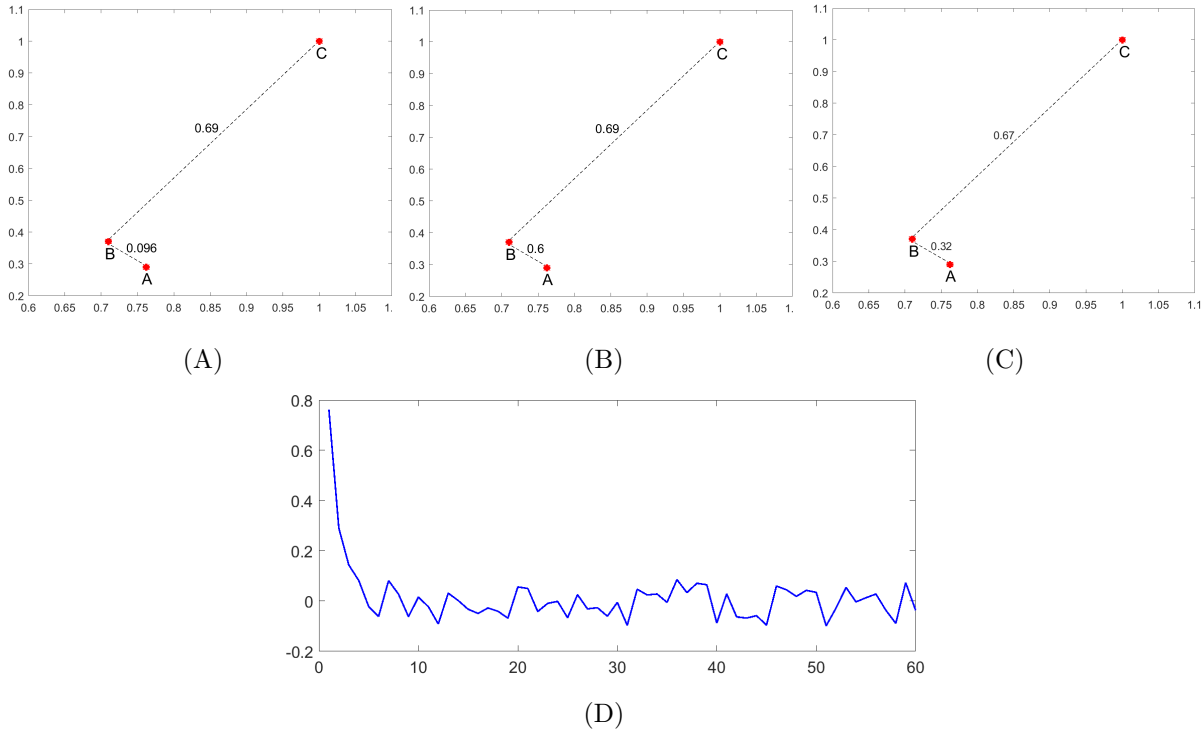


Figure 3.3: Calculating distances in low-and high-dimensional space. (A) Distance calculation of 2D points. (B) Distance calculation of 2D points embedded into 60D + noise $U(-0.2; 0.2)$; (C) Distance calculation of 2D points embedded into 60D + noise: after sketching, (D) Point A embedded into 60D + noise.

3.3.2 Optimal Neighborhood Selection

In this subsection we consider the support size of the locally supported weight functions $w_{i,j}$ utilized in (3.1) for manifold reconstruction. Specifically, given a point-set $X = \{x_k\}_{k=1}^{K}$, we address the problem of choosing a support size h that will guarantee a sufficient amount of points from X in the neighborhood of a point q_i during the MLOP approximation. As described in Section 2.2, the LOP technique has gained much popularity, and many extensions were suggested. However, the proper choice of neighboring points to be used in the reconstruction still remains an important open problem. From the one side, taking points far from the tested point can be influenced by the changing geometry of the manifold, from the other side if the neighborhood size is too small we can lose the robustness to noise property. As a result, support size selection is a critical point when dealing with a fast decaying weight function, and it is important to find an estimate to it (e.g., see the analysis for the MLS case in [89]).

There is a high degree of freedom in choosing the points participating in the approximation since the number of data points is usually very large. Naturally, one would like to make use

of these large degrees of freedom to achieve the "best" reconstruction. In what follows, we use the service centers considerations in order to approximate h as a radius of the ball containing the K-nearest neighbors. It should be noted that naturally, we look for two parameters h_1, and h_2, defined as the support sizes of q_i with respect to P and Q, respectively. The reason for having different supports is due to the fact that the number of points in P and Q can differ, and this should be reflected in the choice of their support size. As will be demonstrated in the numerical examples section 3.5, our approach outperforms the heuristic choice of support size in approximation quality and stability.

The support sizes h_1, and h_2 are closely related to the fill-distance of the P points and the Q points. Let J and I be the sizes of the sets P, and Q respectively. In case $I \leq J$, each q_i can be viewed as a service center that serves approximately $\nu = \left\lfloor \frac{J}{I} \right\rfloor$ points from the p_j's. We use this observation to calculate the fill-distance of P, then estimate the support that guarantees at least ν points in the neighborhood of p_j, as well as the practical support size of the Gaussian $w_{i,j}$ (see the illustration in Figure 3.4).

Unlike the standard definition of fill-distance in scattered data function approximation [83], we introduce

Definition 3.2. *The fill-distance of the set P is*

$$h_0 = median_{p_i \in P} \min_{p_j \in P \setminus \{p_i\}} \|p_i - p_j\| \,. \tag{3.8}$$

It should be also noted that the vanilla definition of fill-distance uses the sup *in the definition (instead of the* median*). However, as mentioned above, in our case we replaced the sup with the median so as to deal with the presence of outliers.*

Definition 3.3. *Given two point-clouds $P = \{p_j\}_{j=1}^{J} \subset \mathbb{R}^n$ and $Q = \{q_i\}_{i=1}^{I} \subset \mathbb{R}^n$, situated near a manifold $\mathcal{M}$ in $\mathbb{R}^n$, such that their sizes obey the constraint $I \leq J$, denote $\nu = \left\lfloor \frac{J}{I} \right\rfloor$. Then we say that the radius that guarantees approximately ν points from P in the support of each point q_i is $\hat{h}_0 = c_1 h_0$, with c_1 given by*

$$c_1 = argmin\{c : \#(\bar{B}_{ch_0}(q_i) \cap P) \geq \nu, \, \forall q_i \in Q\} \,. \tag{3.9}$$

where $\#(B_r(x) \cap P)$ *is the number of points in a ball* $B_r(x)$ *of radius* r *centered at the point* x.

Remark 3.4. *Let* σ *be the variance of a Gaussian* $w(r) = e^{-\frac{r^2}{\sigma^2}}$. *For the normal distribution, four standard deviations away from the mean account for* 99.99% *of the set. In our case, by the definition of* $w_{i,k}$, *since* h *is the square root of the variance,* $4\sigma = 4\frac{h}{\sqrt{2}} = 2\sqrt{2}h_1$ *covers* 99.99% *of the support size of* $w_{i,k}$.

The following theorem indicates how the parameters h_1 and h_2 should be selected.

Theorem 3.1. *Let* $\mathcal{M}$ *be a* d*-dimensional manifold in* $\mathbb{R}^n$. *Suppose given two point-clouds* $P = \{p_j\}_{j=1}^J \subset \mathbb{R}^n$ *and* $Q = \{q_i\}_{i=1}^I \subset \mathbb{R}^n$ *situated near a manifold* $\mathcal{M}$ *in* $\mathbb{R}^n$, *such that their sizes obey the constraint* $I \leq J$, *and let* $\nu = \left\lfloor \frac{J}{I} \right\rfloor$. *Let* $w_{i,j}$ *be the locally supported weight function given by* $w_{i,j} = \exp\left\{ - \|q_i - p_j\|^2/h^2 \right\}$. *Then a neighborhood size of* $h = 2\sqrt{2}\hat{h}_0$ *guarantees* $2^{1.5d}\nu$ *points in the support of* $w_{i,j}$, *where* $\widehat{h}_0 = c_1 h_0$, *with* c_1 *given by* (3.9).

Proof. Given a point q_i we look for the amount of points from P in the support of $w_{i,j}$. Using Remark 3.4 we can estimate the support size of $w_{i,j}$ as 4σ, where $4\sigma = 2\sqrt{2}h_1$. We denote the amount of points from P in the support of q_i by $S_{4\sigma}$. In what follows we assume that the proportion of the number of points in a support does not change with radius changes. Thus, $S_{4\sigma}$ can be determined from the ratio of the volume to the amount of served points: $\frac{V_1}{V_2} = \frac{S_\sigma}{S_{4\sigma}}$, where the volume of a ball with radius $\hat{h}_0$ in $\mathbb{R}^d$ is $V_1 = \pi^{d/2}\hat{h}_0^d/c(d)$, and the volume of a ball with radius 4σ is $V_2 = \pi^{d/2}(4\sigma)^d/c(d) = 2^{1.5d}\pi^{d/2}\hat{h}_0^d/c(d)$ (where c is Euler's gamma function). Thus, $S_{4\sigma} = \nu\frac{V_2}{V_1} = 2^{1.5d}\nu$. □

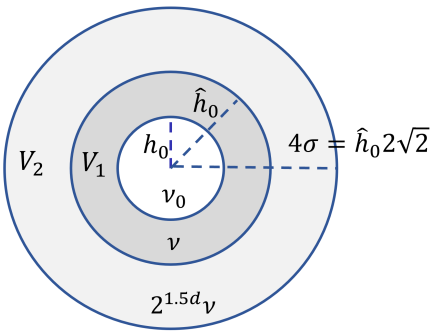


Figure 3.4: Scheme of the fill-distance and the size of the support of the weight function. h_0 is the radius that guarantees at least one point p_j in the support of q_i, $\hat{h}_0$ guarantees ν points, while the real number of points in the support is $2^{1.5d}\nu$.

Corollary 3.1. *Let* P *and* Q *be as defined in Theorem* 3.1, *and assume* $J < I$, *then the number*

of Q points in the support of each $p_j \in P$ is $2^{1.5d}\nu$.

Proof. Each p_j can be viewed as a service center that serves approximately $\nu = I/J$ points q_i from Q. All the preceding definitions remain valid, except that the roles of P and Q are switched. Namely, h_0 is the fill-distance of the set P within the set Q, $\hat{h}_0$ guarantees ν points from Q near each point from P, and the actual number of Q points in the support of P is $2^{1.5d}\nu$. □

Remark 3.5. ***Practical considerations for the support size calculations.*** *As mentioned above, given a point q_i we estimate two different support sizes h_1 and h_2 with respect to the sets P and Q to be used in 3.1. Assume $I < J$, then h_1 is set to be $\hat{h}_0$, which is calculated using definition 3.3. Since we don't have any knowledge about the uniformity of distribution of the Q points over $\mathcal{M}$, we estimate h_2 as follows. We sample I points uniformly from P, and denote this set by Q^{rand}. Next, we estimate h_2 as $\hat{h}_0$ using definition 3.3, when substituting both of the sets P and Q to be Q^{rand}. This gives a rough estimation of h_2 in the scenario when the Q points are equality distributed over $\mathcal{M}$.*

Remark 3.6. *The* reach τ_M *of $\mathcal{M} \subset \mathbb{R}^n$ is defined as the largest number such that any point at distance less than τ_M from $\mathcal{M}$ has a unique nearest point on $\mathcal{M}$ [65]. We note that h should be smaller than the reach τ_M of the manifold $\mathcal{M}$. The reason for this is to prevent a situation where the weighted summations used in the cost function* (3.1) *may be influenced by points in another branch of $\mathcal{M}$ if this constraint is violated.*

3.4 Theoretical Analysis of the Method

As mentioned at the end of Section 2.2, although LOP became popular for surface reconstruction, very important theoretical aspects of the methodology didn't gain attention. The main goal of the analysis presented in this section is to complete the missing parts of the puzzle for the high-dimensional case. We will prove the convergence of the MLOP method, order of approximation, convergence rate as well as its complexity (presented in Theorem 3.4, Theorem 3.5 and Theorem 3.7, respectively). In addition, we will discuss the uniqueness of the MLOP solution (see Subsection 3.4.4).

3.4.1 Convergence to a Stationary Point

We are now ready to state our main convergence theorem. The fact that the cost function is non-convex poses a challenge for the proof of the convergence of the proposed method. First, we define h as described in Section 3.3.2 and assume that the h-ρ condition, defined above, is satisfied. Next, we utilize the following general non-convex convergence theorem presented in [82] to prove the convergence of our method.

Theorem 3.2. *Let $f : \mathbb{R}^d \to \mathbb{R}$, not necessarily convex, be twice continuously differentiable and has Lipschitz gradient, with constant L, i.e., $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$. Let its the gradient descent of f be $x^k = x^{k-1} - \alpha\nabla f(x^{k-1})$, with bounded step size $0 < \alpha < 1/L$. Suppose, all saddle points of the function f are strict-saddle (i.e., for all critical points x^* of f, $\lambda_{\min}\nabla^2(f(x^*)) < 0$). Then the gradient descent with random initialization and sufficiently small constant step size converges almost surely to a local minimizer or to minus infinity. i.e., if x^* is a strict saddle then* $\Pr(\lim x_k = x^*) = 0$.

We also recall the following theorem on eigenvalue bounds, due to Iyengar et al. [78].

Theorem 3.3. *Let X be a self-adjoint matrix, with entries $x_{i,j}$. Then its highest and lowest eigenvalues lie in the range*

$$\lambda_{\min}, \lambda_{\max} \in [l, u]\,,$$

where

$$l = \min_{i \in I}\left(x_{i,i} - \sum_j |x_{i,j}|\right) \text{ and } \quad u = \max_{i \in I}\left(x_{i,i} + \sum_j |x_{i,j}|\right)\,.$$

Theorem 3.4 (Convergence to a stationary point)**.** *Let $\mathcal{M}$ be a d-dimensional manifold in $\mathbb{R}^n$, where d is an unknown intrinsic dimension. Suppose that the scattered data points $P = \{p_j\}_{j=1}^J$ were sampled near the manifold $\mathcal{M}$, h_1 and h_2 are set as defined in Section* 3.3.2*, and the h-ρ set condition is satisfied with respect to $\mathcal{M}$. Let the points $Q^{(0)} = \{q_i^{(0)}\}_{i=1}^I$ be sampled from P. Then the gradient descent iterations* (3.1) *converge almost surely to a local minimizer Q^*.*

Proof. We proceed by verifying that the conditions of Theorem 3.2 hold. At a high level, our proof consists of the following steps:

1. Calculate the Hessian of the cost function (3.1).
2. Bound the eigenvalues of the Hessian.
3. Show that the minimal eigenvalue is negative.
4. Bound the norm of the Hessian.

We rephrase the minimization problem from (3.1) by writing E_1 and E_2 in a matrix form as

$$E_1 = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}^t \begin{pmatrix} \|q_1 - p_1\| w_{1,1} & \dots & \|q_1 - p_J\| w_{1,J} \\ \vdots & \vdots & \vdots \\ \|q_I - p_1\| w_{I,1} & \dots & \|q_I - p_J\| w_{I,J} \end{pmatrix} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix},$$

$$E_2 = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_I \end{pmatrix}^t \begin{pmatrix} 0 & \eta(\|q_1 - q_2\|) w_{1,2} & \dots & \eta(\|q_1 - q_I\|) w_{1,I} \\ \eta(\|q_2 - q_1\|) w_{2,1} & 0 & \dots & \eta(\|q_2 - q_I\|) w_{2,I} \\ \vdots & \vdots & \vdots & \vdots \\ \eta(\|q_I - q_1\|) w_{I,1} & \dots & \eta(\|q_I - q_{I-1}\|) w_{I,I-1} & 0 \end{pmatrix} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}.$$

The cost function is rewritten as

$$G(Q) = \vec{1}^t \Phi \vec{1} + \vec{\Lambda}^t \Psi \vec{1},$$

where $\phi_{i,j} = \|q_i - p_j\| w_{i,j}$ are the entries of Φ, $\psi_{i,j} = \eta(\|q_i - q_{i'}\|) \hat{w}_{i,i'}$ are the entries of Ψ, and the vector of balancing parameters $\vec{\Lambda} = (\lambda_1, \dots, \lambda_I)$ is defined in (3.6).

The proof relies on the fact that the weights $w_{i,j}$ are defined by rapidly decreasing functions with bounded support. Although the weight function $w_{i,j}$ in definition (3.1) does not have compact support, for practical reasons it can be assumed that the Gaussian with 4σ covers 99% of the support size. As a result, the matrices Φ and Ψ are sparse, and the number of their non-zero entries depend on the support size of $w_{i,j}$. Following Definition 3.4, we estimate the number of non-zero entries in each row of the matrices Φ and Ψ, in the kth iteration of our

algorithm, as

$$\Phi_{q_i^{(k)}} = \#\{B_h(q_i^{(k)}) \cap P\}\,,$$
$$\Phi_{p_j^{(k)}} = \#\{B_h(p_j^{(k)}) \cap Q^{(k)}\}\,,$$
$$\Psi_{q_i^{(k)}} = \#\{B_h(q_i^{(k)}) \cap Q^{(k)}\}\,,$$

where $B_h(x)$ is a ball centered at x with radius h.

Using these definitions, we calculate the Hessian and its eigenvalues for our cost function in (3.1),

$$H = \nabla^2 G(Q) = \nabla^2 E_1 + \Lambda \nabla^2 E_2\,.$$

For simplicity, we denote $r_{i,j} = q_i - p_j$; then with $w_{i,j} = \exp\big\{-\|q_i - p_j\|^2/h_1^2\big\}$, $\frac{\partial E_1}{\partial q_i}$ can be rewritten as

$$\frac{\partial E_1}{\partial q_i} = \sum_{j=1}^{J} \frac{r}{\|r_{i,j}\|}\left(1 - \frac{2}{h_1^2}\|r_{i,j}\|^2\right) w_{i,j}\,.$$

We notice that, by definition, $\frac{\partial^2 E_1}{\partial q_i \partial q_i'} = 0$, and by the chain rule we have

$$\frac{\partial^2 E_1}{\partial q_i^2} = \sum_{j=1}^{J} a(r_{i,j}) w_{i,j}\,,$$

where $a(r) = -\frac{2}{h_1^2}\|r\|\left(1 + \frac{2}{h_1^2}\|r\|^2\right) < 0$.
For the second term in expression (3.1), we denote $\hat{r}_{i,i'} = q_i - q_i'$, and recall that $\eta(r) = \frac{1}{r^3}$. Then the first derivative of E_2 is

$$\frac{\partial E_2}{\partial q_i} = \sum_{i'=1}^{I}\left(-\frac{\hat{r}_{i,i'}}{\|\hat{r}_{i,i'}\|^5} - \frac{2\hat{r}_{i,i'}}{3h_2^2\|\hat{r}_{i,i'}\|^3}\right)\hat{w}_{i,i'}\,.$$

The second derivatives can be expressed as

$$\frac{\partial^2 E_2}{\partial q_i \partial q_i'} = -b(\hat{r}_{i,i'})\hat{w}_{i,i'}\,,$$

where $b(\hat{r}) = \frac{4}{\|\hat{r}\|^5} + \frac{3\frac{1}{3}}{h_2^2\|\hat{r}\|^3} + \frac{4}{3h_2^4\|\hat{r}\|} > 0$, and

$$\frac{\partial^2 E_2}{\partial q_i^2} = \sum_{i'=1}^{J} b(\hat{r}_{i,i'})\hat{w}_{i,i'} \,.$$

Thus,

$$H = \begin{pmatrix} \sum\limits_{j=1}^{J} a(r_{1,j})w_{1,j} + \lambda_1 \sum\limits_{i'=1}^{I} b(\hat{r}_{1,i})w_{1,i'}; & -\lambda_1 b(\hat{r}_{1,2]})w_{1,2}; & \dots & -\lambda_1 b(\hat{r}_{1,I})w_{1,I} \\ \vdots & \vdots & \vdots & \vdots \\ -\lambda_I b(\hat{r}_{I,1})w_{I,1}; & \dots & -\lambda_I b(\hat{r}_{I,I-1})w_{I,I-1}; & \sum\limits_{j=1}^{J} a(r_{I,j})w_{I,j} + \lambda_I \sum\limits_{i'=1}^{I} b(\hat{r}_{I,i'})\hat{w}_{i,i'} \end{pmatrix}.$$

Let us check that the eigenvalues $\lambda_{\min}$, and $\lambda_{\max}$ of the MLOP Hessian $H \in \mathbb{R}^{I\times I}$ are bounded and negative. By Theorem 3.3, the eigenvalues of H belong to the range $\lambda_{\min}, \lambda_{\max} \in [l, u]$, where in our case

$$l = \min_{i\in I} \left(\sum_{j=1}^{J} a(r_{i,j})w_{i,j} + \lambda_i \sum_{i'=1}^{I} b(\hat{r}_{i,i})\hat{w}_{i,i'} - \sum_{i'=1}^{I} |\lambda_i b(\hat{r}_{i,i'])}\hat{w}_{i,i'}| \right) .$$

Let $h = min(h_1, h_2)$. Using the expressions for $a(r)$ and $b(r)$, and the fact that from Definition 3.4 $\|r\| = 4\sigma = \frac{4h}{\sqrt{2}}$, it can be verified that $0 < \min(b(r)) \leq \frac{c_1}{h^5}$, $\min(a(r)) \leq \frac{-c_2}{h\sqrt{2}}$, $\max(a(r)) \leq 0$, where c_1, c_2 are constants and $c_1, c_2 > 0$. Thus, since $\lambda_i < 0$ from (3.6), and the number of points from P and Q in the support of q_i is bounded by $\Phi_{q_i^{(k)}}$ and $\Psi_{q_i^{(k)}}$, respectively, we have

$$u \leq -\frac{c_2}{h\sqrt{2}} \max_{i\in I}(\Phi_{q_i^{(k)}}) < 0 \,, \tag{3.10}$$

$$l \leq -\frac{c_2}{h\sqrt{2}} \max_{i\in I}(\Phi_{q_i^{(k)}}) - \frac{2c_1}{h^5} \max_{i\in I}(|\lambda_i|) \max_{i\in I}(\Psi_{q_i^{(k)}} - 1) < 0 \,. \tag{3.11}$$

Since the eigenvalues are negative, all saddle points of the MLOP target function are strict-saddle, and the second condition of Theorem 3.2 holds. Let us also check that the first condition in Theorem 3.2 is satisfied, i.e., that the norm of the Hessian is bounded: $\|H\| \leq L$, and find

L. Indeed,

$$\|H\|_2 = \lambda_{\max}(H'H) = \lambda_{\max}(H^2) = \max\{\lambda^2 \mid \lambda \text{ is an eigenvalue of } H\} = \max\{\lambda_{\max}^2, \lambda_{\min}^2\},$$

so the required bound holds with $L = \max\{\lambda_{\max}^2, \lambda_{\min}^2\} \leq \max\{u^2, l^2\} = l^2$.

To summarize, all the conditions of Theorem 3.2 are satisfied. It follows that the gradient descent with random initialization and a sufficiently small constant step size converges almost surely to a local minimizer or minus infinity. □

3.4.2 Order of Approximation

The support size of the locally supported function defining the weight function $w_{i,j}$ which is tightly related to the fill-distance of available sample data P, plays an important role in the order of approximation of the MLOP algorithm. The following theorem guarantees an $O(h^2)$ order of approximation, which is asymptotic as $h \to 0$. Here, $h = \max(h_1, h_2)$, where h_1 and h_2 are defined in Remark 3.5.

Theorem 3.5 (Order of approximation)**.** *Let $P = \{p_j\}_{j=1}^J$ be a set of points that are sampled (without noise) from a d–dimensional C^2 manifold $\mathcal{M}$, and satisfy the h-ρ condition. Then for a fixed ρ, and a finite support of size h of the weight functions $w_{i,j}$, the set of points Q defined by the* MLOP *algorithm has an order of approximation $O(h^2)$ to $\mathcal{M}$.*

Proof. We break the proof into the following steps.

1. **The MLOP cost function can be rewritten in matrix form as $AQ = R$.** We look for a solution Q that will minimize the cost function in (3.1), i.e., such that the gradient $\nabla G(Q) = 0$. Thus equation (3.3) can be recast as a system of equations

$$(1 - \tau_{i'})q_i + \tau_i \sum_{i' \in I \setminus \{i\}} q_{i'}^{(k)} \frac{\beta_i^{i'}}{\sum_{i' \in I \setminus \{i\}} \beta_i^{i'}} = \sum_{j=1}^{J} p_j \frac{\alpha_j^{i'}}{\sum_{j \in J} \alpha_j^{i'}}, \tag{3.12}$$

where we express $\lambda_{i'}$ in the form $\lambda_{i'} = \tau_{i'} \dfrac{\sum_{j \in J} \alpha_j^{i'}}{\sum_{i' \in I \setminus \{i\}} \beta_i^{i'}}$.

As a result, the problem can be written in matrix form as $AQ = R$, where both A, and R depend on Q. In the new notations, we need to show that the points $Q = A^{-1}R$ lie at a distance of $O(h^2)$ from $\mathcal{M}$.

2. **The R term has order of approximation of $O(h^2)$ to $\mathcal{M}$.** Let J_k be the indices of points from P which lie at the distance h from a given poin $q_{i'}$ (the set is not empty due to the optimal neighborhood selection in Subsection 3.3.2). Let t be the index of the closest point in $\{p_j\}_{j\in J_k}$ to the projection of $q_{i'}$ on the manifold $\mathcal{M}$ (Figure 3.5 left), and T be the tangent space to $\mathcal{M}$ at that point. Then the sum $\sum_{j=1}^{J} p_j \frac{\alpha_j^{i'}}{\sum_{j\in J}\alpha_j^{i'}}$, with bounded support weights $w_{i,j}$, is a local convex combination of points p_k, and thus it also lie in T. Since $\mathcal{M}$ is C^2, T approximates $\mathcal{M}$ in the order of $O(h^2)$, the r.h.s of (3.12) can be written as $F + O(h^2)$, where $F = \{f_i\}_{i\in I}$ are points on $\mathcal{M}$. Thus, $AQ = F + O(h^2)$.

3. **Then norm of the matrix A^{-1}, $\|A^{-1}\|_\infty$ and its entries $(A^{-1})_{l,m}$ are bounded**. For $\tau_i \in [0, 0.5)$, the matrix A is strictly diagonally dominant and therefore we can bound $\|A^{-1}\|_\infty \leq c_1(\tau_i)$, as well as $|(A^{-1})_{l,m}| < c_2(\tau_i)$ for two points q_l and q_m lying at a distance of at least h, where the influence of distant points decays exponentially with distance. We also note that since the rows of A sum up to one, so do the rows of A^{-1}.

4. **The MLOP reconstruction is of order $O(h^2)$ to the manifold**. The MLOP reconstruction can be written as $Q = A^{-1}F + O(h^2)$, where each element of $(A^{-1}F)_{i'}$ is the affine average of f_i over the manifold, with exponentially decaying weights $w_{i,j}$. Let T be the tangent space to the manifold $\mathcal{M}$ at the point $f_{i'}$, and let t_i be the projection of f_i on T (Figure 3.5 right). If we rewrite f_i using its projection as $f_i = t_i + r_i$, it follows that $(A^{-1}F)_{i'} = \sum_{i\in I} A^{-1}_{i',i}(t_i + r_i) = \sum_{i\in I} A^{-1}_{i',i}t_i + \sum_{i\in I} A^{-1}_{i',i}r_i$. We would first like to show that $\|\sum_{i\in I} A^{-1}_{i',i}t_i - f_{i'}\| = O(h)$, and since $\sum_{i\in I} A^{-1}_{i',i}t_i$ is on T, and T approximates the manifold with $O(h^2)$, it will follow that $\sum_{i\in I} A^{-1}_{i',i}t_i$ is of order $O(h^2)$ distance from $\mathcal{M}$. In addition, we show that $\sum_{i\in I} A^{-1}_{i',i}(r_i) = O(h^2)$.

   In more details:

   (a) For a given $q_{i'}$, we denote by I_k its q_i neighbors at the distance $\|q_i - q_{i'}\| \in [kh, (k+1)h]$. We use the fact that the sum of the rows of A^{-1} equals one, and rewrite and

estimate $\sum_{i\in I} A_{i',i}^{-1} t_i$ as

$$\|\sum_{i\in I} A_{i',i}^{-1} t_i - f_{i'}\| = \|\sum_{i\in I} A_{i',i}^{-1}(t_i - f_{i'})\| \leq \sum_{i\in I} c_2(\tau)\|t_i - f_{i'}\| = O(h)\,. \tag{3.13}$$

For the last step we note that $\|t_i - f_{i'}\| = \|t_i - f_i + f_i - f_{i'}\| \leq \|t_i - f_i\| + \|f_i - f_{i'}\| \leq O(h) + (k+1)h$, due to the local approximation property and the distance constraint on the point q_i. Thus, the sum $\sum_{i\in I} A_{i',i}^{-1} t_i$ is an affine combination of points t_i on T and therefore lies in T as well (in a distance $\leq O(h)$), therefore it will follow that it is an $O(h^2)$ from the manifold.

(b) Next, similar considerations show that $\|r_i\| \leq \|f_i - f_{i'}\|^2 \leq c_3((k+1)h + O(h))^2$.

To conclude, that based on items (a) and (b), the MLOP order of approximation to the manifold is $O(h^2)$.

□

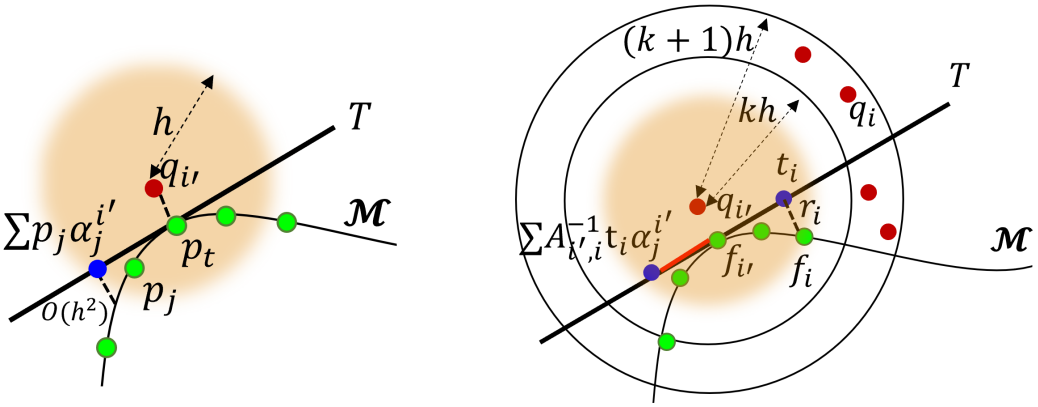


Figure 3.5: Illustration of the points participating in the estimate of the order of approximation. Left: demonstration why the affine combination of the p_j points, in the neighborhood of $q_{i'}$, is of order $O(h^2)$. Right: Illustration of the elements used in the estimation of the order of approximation. The P points are marked in green, the Q points in red, while the auxiliary points in the proof are marked in blue.

3.4.3 Rate of Convergence

First, let us consider the gradient-descent rate of convergence of a Lipschitz-continuous strongly convex function. This rate of convergence depends on the condition number of the Hessian of the cost function, and so on the ratio between the smallest and the largest eigenvalues of the

Hessian, i.e., $|1 - c\frac{\lambda_{\min}}{\lambda_{\max}}|$, with $0 < c < 2$. Therefore, if our cost function would be convex, the rate of convergence could be $O(1 - c/h^4)$. However, for non-convex optimization, the situation is much more complex. In our setting, where there is no convexity, one can analyze convergence to ϵ-first-order stationary points, as defined below.

Definition 3.4. *A differentiable function $f(\cdot)$ is called L-smooth if for any x_1, x_2*

$$\|\nabla f(x_1) - \nabla f(x_2)\| \leq L\|x_1 - x_2\| .$$

Definition 3.5. *If $f(\cdot)$ is a differentiable function, we say that x is an ϵ-first-order stationary point if $\|\nabla f(x)\| \leq \epsilon$.*

For the rate of convergence of our method, we will use the following theorem proved by Nestrove in [98].

Theorem 3.6. *Let $f(\cdot)$ be an L- smooth function that is bounded below. Then for any $\epsilon > 0$, for the gradient descent with stop criterion $\|\nabla f(x)\| \leq \epsilon$, the output will be an ϵ-first-order stationary point, which will be reached after $k = \frac{L}{w\epsilon^2}(f(x_0) – f^*) - 1$ iterations, where w is $0 < w < 2$. In case the starting point is close enough to the local minimum, the convergence is linear.*

It follows that in our case the rate of convergence is bounded.

Theorem 3.7 (Rate of convergence)**.** *Let the points-set $P = \{p_j\}_{j=1}^J$ be sampled near a d-dimensional manifold in $\mathbb{R}^n$ and let the assumptions in Theorem 3.4 be satisfied. Let the cost function G, defined as in (3.1), be an L-smooth function. For any $\epsilon > 0$, let Q^* be a local fixed-point solution of the gradient descent iterations. Set the termination condition as $\|\nabla G(Q)\| \leq \epsilon$. Then Q^* is an ϵ-first-order stationary point that will be reached after $k = \frac{L(G(Q^{(0)}) – G(Q^*))}{w\epsilon^2} - 1$ iterations, where w is $0 < w < 2$, and $L = l^2$ and l is given in* (3.11).

Proof. It is quite easy to verify that $G(Q)$ satisfies all the conditions of Theorem 3.6; in particular, the L-smoothness condition was proven above. □

Remark 3.7. *In our case, due to the bound on l in (3.11), we see that k is of order $\frac{1}{h^{10}}$. However, in practice, in our numerical examples 3.5.1 - 3.5.5, fewer iterations were needed to achieve convergence. In an example presented in the following section, with approximately* 800 *noisy points P and* 160 *points in Q (sampled in a certain area around a specific point), of a two-dimensional manifold embedded into a* 60*-dimensional space, the method converged in approximately* 500 *iterations which took around* 90 *seconds. When the initial set Q was randomly sampled from P, we observed convergence in* 50 *iterations which took 11 seconds.*

Remark 3.8. *It should be emphasized that the calculations of the gradient for each point are independent of one another, and in order to reduce the execution time, they can be run in multiple threads.*

3.4.4 Uniqueness

As shown in the previous section, convergence to a local minimum is guaranteed. However, since the cost function in (3.1) is non-convex, a unique global solution can not be ensured. In order to address the uniqueness question, we have to rephrase the notion of uniqueness for our case. We do not refer to the uniqueness of the set Q, since there may be many sets Q which satisfy the cost function (3.1), but to a common property of these optimal Q sets, the fill-distance of their points. For instance, given a solution, its linear transformation can still minimize (3.1). This scenario is illustrated in Figure 3.6. In this example, which will be explained in detail in the experimental section, the orthogonal matrices in $\mathbb{R}^2$, which are represented by their angle, form a manifold. Although the two sets in Figure 3.6 (left and right) differ, they can still be solutions to the problem.

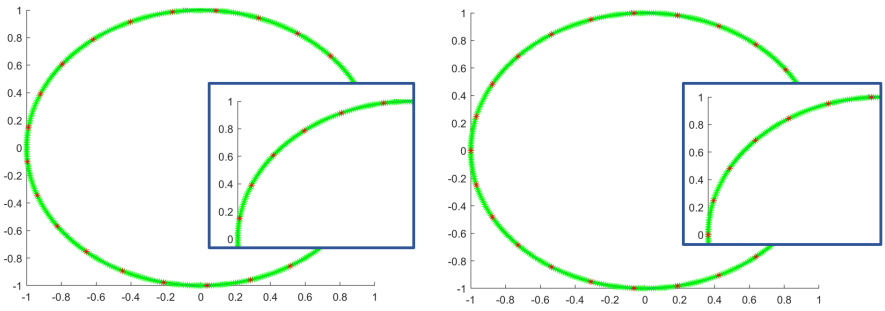


Figure 3.6: Manifold of orthogonal matrices: each matrix is represented by means of an angle (green), sampled with the same fill-distance, in two manners (red).

Thus the appropriate notation of uniqueness of the solution is as follows:

Definition 3.6. *Let Q_1 and Q_2 be two point-sets uniformly sampled from a manifold $\mathcal{M}$, with fill-distance h_2^1 and h_2^2, respectively. Then Q_1 and Q_2 are said to be "distribution equivalent" if their fill-distances coincide ($h_2^1 = h_2^2$). For a fixed fill-distance h_q, the corresponding class of distribution equivalent sets is denoted here by $[h_q]$.*

Remark 3.9. *Let Q^* be a solution of the optimization problem* (3.1)*, from points P. Then Q^* is unique up to the equivalence class $[h_q]$. This follows from the definition of h_q, which specifies the number of P points served by a single q_i, which uniquely define the equivalence class $[h_q]$ of the solution Q^*.*

3.4.5 Complexity of the MLOP Algorithm

The complexity of the MLOP algorithm described in Algorithm 1 is based on a pre-step and a gradient decent iterations. As described in Section 3.3.1, due to the curse of dimensionality and presence of noise all the norms are calculated in a lower dimension m. Thus, a pre-step to the MLOP algorithm is reducing the dimension of P from n to m (where $m \ll n$), and have the complexity nmJ. In addition in every gradient descent step, and for every q_i we reduce the dimension of current Q which results in the complexity of nmI. As a result, a single gradient descent step is $O(I(nmI+I+J))$. With efficient neighboring calculation, this can be reduced to $O(I(nm\hat{I}+\hat{J}))$, where $\hat{I}$ and $\hat{J}$ are the numbers of points in the support of the weight function with respect to the Q and P sets, respectively (for instance, in the numerical examples section 3.5 below $\hat{J}$ was around 30 points, instead of 900 points in P). These operations are repeated k times until convergence, where k is bounded as in Theorem 3.7. Thus, the overall complexity is $O(nmJ + kI(nm\hat{I} + \hat{J}))$.

Corollary 3.2. *Given a point-set $P = \{p_j\}_{j=1}^J$ sampled near a d-dimensional manifold $\mathcal{M} \in \mathbb{R}^n$, and let $Q = \{q_i\}_{i=1}^I$ be a set of points that will provide the desired manifold reconstruction. Then the complexity of the* MLOP *algorithm is $O(nmJ + kI(nm\widehat{I} + \widehat{J}))$, where the number of iterations k is bounded as in Theorem* 3.7*, $m \ll n$ is the smaller dimension to which we reduce the dimension of the data, and $\widehat{I}$ and $\widehat{J}$ are the numbers of points in the support of the weight functions $\hat{w}_{i,i'}$, $w_{i,j}$ with the Q-set and P-set, respectively. Thus, the approximation is linear*

in the ambient dimension n, and does not depend on the intrinsic dimension d.

3.5 Numerical Examples

In this section, we present some numerical examples which demonstrate the validity of our method, as well as its robustness under different scenarios, for example, diverse manifold topologies, different amounts of noise, and many intrinsic dimensions. In all the examples the input points P were sampled uniformly in the parameter space. Next, a uniform noise $U(-\sigma, \sigma)$ with magnitude σ was added. Then the set Q was initialized by sampling from the set P around a certain selected point. In what follows we illustrate the results of applying the MLOP algorithm.

3.5.1 One-Dimensional Orthogonal Matrices

Consider the case of the manifold $O(2)$ of orthogonal matrices, embedded into a 60-dimensional linear space by using the parameterization

$$\hat{p} = [\cos(\theta), -\sin(\theta), \sin(\theta), \cos(\theta), 0, \ldots, 0]\,,$$

where $\theta \in [-\pi, \pi]$. The input data $\hat{P}$ were constructed by sampling 500 equally distributed points in the parameter space. Next, we randomly sampled an orthogonal matrix $A \in \mathbb{R}^{60\times 60}$, and created a new point-set via non-trivial vector embedding

$$P = A\hat{P}\,. \tag{3.14}$$

Later we added a uniform noise $U(-0.2, 0.2)$, and initialized the set Q selecting 50 points around a certain point. Figure 3.7 left illustrates the first two coordinates of the points in our set (after a multiplication with A^{-1}). The noisy sampled points are shown in green, while the initial reconstruction points are shown in red. Figure 3.7 right shows the reconstructed and denoised manifold of orthogonal matrices, after 500 iterations of the MLOP algorithm (red).

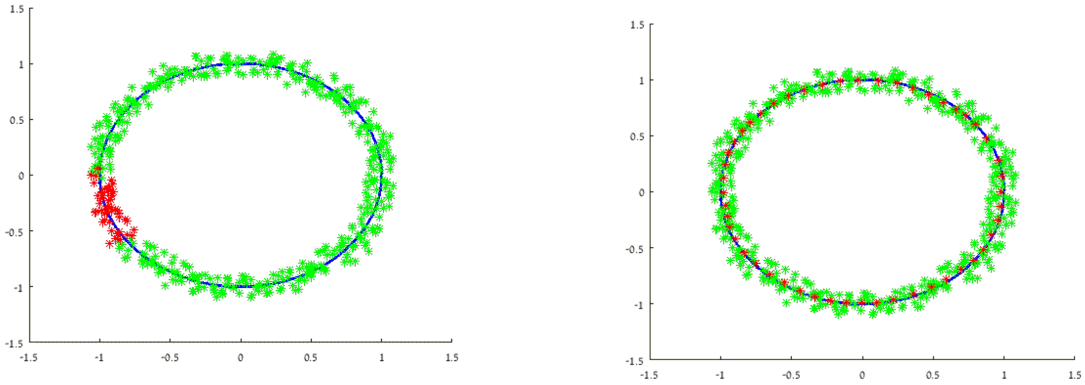


Figure 3.7: Manifold of orthogonal matrices embedded into a 60-dimensional space. Shown are the first two coordinates of the point-set (after multiplication with A^{-1}). Left: Scattered data with uniformly distributed noise $U(-0.2; 0.2)$ (green), and the initial point-set $Q^{(0)}$ (red) Right: The resulting point-set of MLOP algorithm after 500 iterations, $Q^{(500)}$ (red) overlaying the noisy samples (green).

3.5.2 Three-Dimensional Cone Structure

Next, we demonstrate the ability of the MLOP to cope with a geometric structure of different dimensions at different locations. Here we combined a 3-dimensional manifold, namely, a cone structure, with a one-dimensional manifold, namely, a line segment. This object was embedded into a 60-dimensional linear space. The cone's parameterization used was

$$p = tv_1 + \frac{e^{-R^2}}{\sqrt{2}}(\cos(u)v_2 + \sin(u)v_3)\,,$$

where $v_1 = [1, 1, 1, 1, 0, \ldots, 0], v_2 = [0, 1, -1, 0, 0, \ldots, 0], v_3 = [1, 0, 0, -1, 0, \ldots, 0]$, $(v_1, v_2, v_3) \in \mathbb{R}^{60}$, $t \in [0, 2]$, $R \in [0, 2.5]$, and $u \in [0.1\pi, 1.5\pi]$. We sampled 720 points from the structure with added uniformly distributed noise of magnitude 0.2. The initial set $Q^{(0)}$ of size 144 was selected (Figure 3.8 left), and 500 iterations of the MLOP were performed to reconstruct and denoise the geometrical structure (Figure 3.8 right).

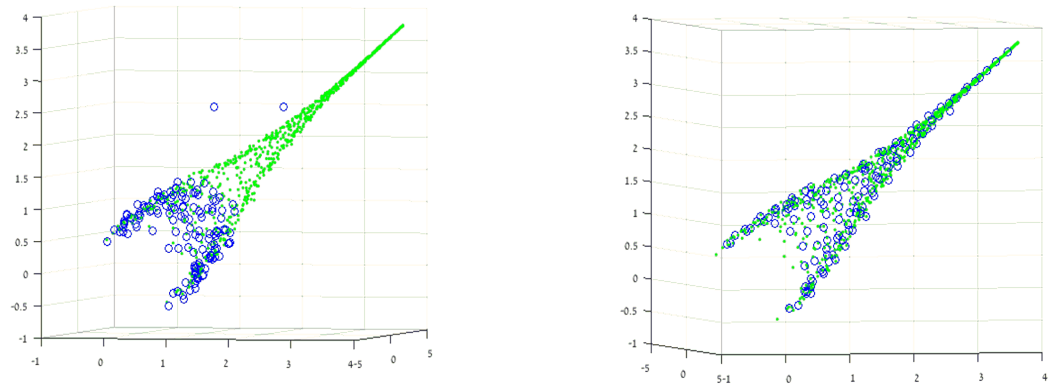


Figure 3.8: Geometrical structure of changing dimension. Combination of a cone and a line segment, embedded into a 60-dimensional space. The first three coordinates of the point-set are shown. Left: Scattered data with uniformly distributed noise $U(-0.2; 0.2)$ (green), and the initial point-set $Q^{(0)}$ (blue) Right: The point-set generated by the MLOP algorithm after 500 iterations, $Q^{(500)}$ (blue) overlaying the noisy samples (green).

3.5.3 Two-Dimensional Cylindrical Structure

In the next example, we embedded a two-dimensional cylindrical structure into a 60-dimensional linear space. We sampled the structure using the parameterization

$$p = tv_1 + \frac{R}{\sqrt{2}}(\cos(u)v_2 + \sin(u)v_3)\,,$$

where $v_1 = [1, 1, 1, 1, 1, \ldots, 1]$, $v_2 = [0, 1, -1, 0, 0, \ldots, 0], v_3 = [1, 0, 0, -1, 0, \ldots, 0]$, $(v_1, v_2, v_3 \in \mathbb{R}^{60})$, $t \in [0, 2]$ and $u \in [0.1\pi, 1.5\pi]$. Using this representation 816 equally distributed (in parameter space) points were sampled with uniformly distributed noise (i.e., $U(-0.1, 0.1)$). As can be seen in Figure 3.9 left, the initial set $Q^{(0)}$ of size 163 was selected very roughly, and 500 iterations of the MLOP were performed to reconstruct the cylindrical structure, shown in Figure 3.9 right.

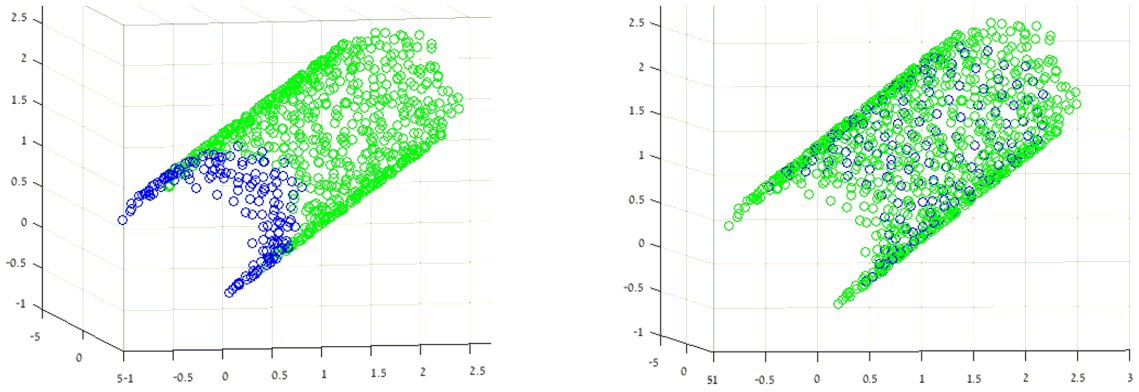


Figure 3.9: Cylindrical structure embedded into a 60-dimensional space. The first three coordinates of the point-set are shown. Left: Scattered data with uniformly distributed noise $U(-0.1; 0.1)$ (green), and the initial point-set $Q^{(0)}$ (blue) Right: The point-set generated by the MLOP algorithm after 500 iterations, $Q^{(500)}$ (blue) overlaying the noisy samples (green).

3.5.4 Robustness to Noise

The noise level has a direct influence on the accuracy of the reconstruction. Here we examine the robustness of the MLOP under various levels of noise. Our test was performed on the two-dimensional cylindrical structure embedded into 60-dimensions, with various amounts on noise magnitude (0, 0.1, 0.2, and 0.5). The accuracy was calculated as the relative error of the reconstruction Q, against a densely sampled noise-free cylindrical structure. The norm used for accuracy calculations was the one that is based on linear sketching, as defined in Section 3.3.1. As can be seen in Figure 3.10, even with a noise level of 0.5, the reconstruction quality is satisfactory (with a relative error of 0.15).

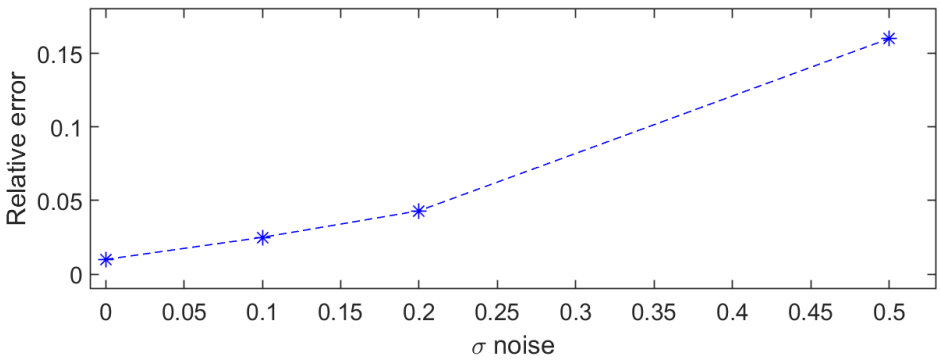


Figure 3.10: Effect of noise level on the reconstruction accuracy of a cylindrical structure embedded into a 60-dimensional space.

3.5.5 Six-dimensional cylindrical structure

Finally, we tested our method on manifolds of the higher dimension by utilizing an n-sphere to generate an $(n+1)$-dimensional cylinder (in the example of the two-dimensional cylinder, we used a circle to generate the structure). Here, we utilized a five-dimensional sphere to build a six-dimensional manifold, using the parameterization

$$x_1 = R\cos(u_1)\,, \quad x_2 = R\sin(u_1)\cos(u_2), \quad \ldots, \quad x_6 = R\sin(u_1)\sin(u_2)\cdots\sin(u_5)\sin(u_6)\,.$$

We then embedded the sampled data in a 60-dimensional space

$$p = tv_0 + R^2[x_1, x_2, x_3, x_4, x_5, x_6, 0, \ldots, 0]\,, \tag{3.15}$$

where $R = 1.5$, $t \in [0, 2]$, $u_i \in [0.1\pi, 0.6\pi]$, and $v_0 \in \mathbb{R}^{60}$ is a vector with 1's in positions $1, ..., d+1$ and 0 in the remaining positions. In this test, we sampled 1200 points from this manifold and added a noise $U(-0.1, 0.1)$. The initial reconstruction set was chosen to consist of randomly selected 460 points. The method converged after approximately 300 iterations. To avoid trying to visualize a six-dimensional manifold, we plot in Figure 3.11 the cross-section of the cylindrical structure in three-dimensions. We evaluate the efficiency of the denoising effect by calculating the maximum relative error, root mean square error, and variance of both the initial $Q^{(0)}$ points and the noise-free reconstruction set $Q^{(300)}$ with respect to the closest point in the clean reference data. As a result, the errors if $Q^{(0)}$ are 0.083, 0.32 ± 0.0007, and of the noise-free reconstruction are 0.058, 0.28 ± 0.0006. Thus, we see that in this scenario of non-trivial intrinsic dimension of the manifold the error decrease dramatically. In addition, the fill-distance of the initial random $Q^{(0)}$ set was 0.36, and 0.32 in the reconstruction. Thus, we also observe the effect of quasi-uniform sampling after applying the MLOP.

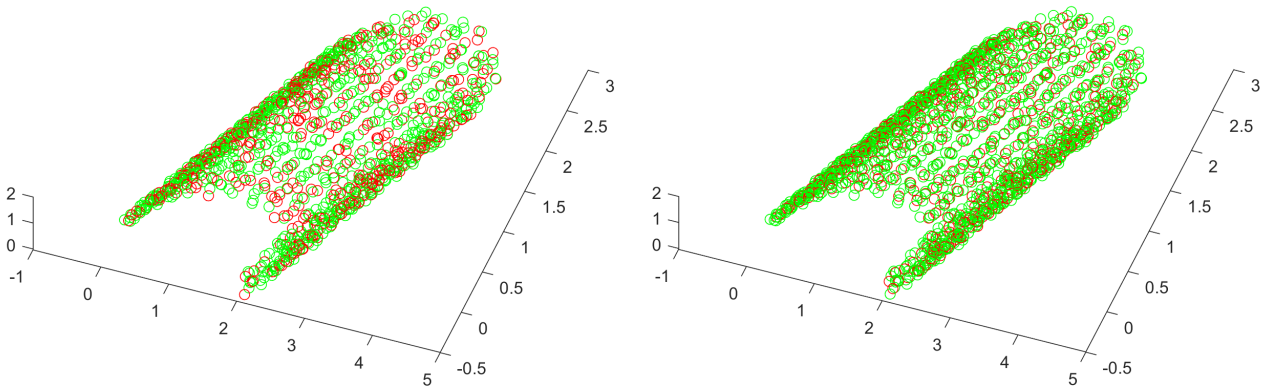


Figure 3.11: Six-dimensional cylindrical structure embedded in a 60-dimensional space. The cross-section of the six-dimensional cylindrical structure is plotted in three-dimensions. Left: Scattered data with uniformly distributed noise $U(-0.1; 0.1)$ (green), and the initial point-set $Q^{(0)}$ (red) Right: The point-set generated by the MLOP algorithm after 300 iterations, $Q^{(300)}$ (red) overlaying the noisy samples (green).

3.5.6 Applications to Image Processing

Manifold denoising and reconstruction methodology can be also applied to image processing problems. At the beginning of this paper, we described the cryo-EM (in Figure 3.1) which motivated our study. In this framework a manifold is created by acquiring images of a single object in various directions. As a preliminary example, before addressing the real case of cryo-EM, we simulated data that resemble the cryo-EM conditions. Specifically, we sampled 900 images of ellipses of size 20×20. The ellipses were centered and no rotations were used. Thus, we have 900 samples of a 2-dimensional submanifold embedded in $\mathbb{R}^{400}$. We added a Gaussian noise $N(0; 0.05)$ to each pixel. Figure 3.12 shows the sample of the manifold (with some zoom-in examples), along with a graph where the (x, y) - coordinates of each point are the ellipse radii. For the execution of the MLOP, we took 180 ellipses as the initial sample points (Figure 3.13 left). As can be seen in Figure 3.13 right, after 1000 iterations the samples were cleaned, while the radii distribution graph shows that the radii domain is fully sampled.

We evaluated the MLOP denoise performance on the ellipses samples Q. We measured the SNR as $SNR = \frac{\mu}{\sigma}$ on the background pixels of each ellipse image (where μ is the average signal value, while σ is the standard deviation). We observe that the median SNR of the set Q increased after applying the MLOP denoising, from 15.6 to 36.5. This gives us a quantitative measure of the denoising performed by the MLOP (as can also be seen in Figure 3.13 in the

zoomed-in areas).

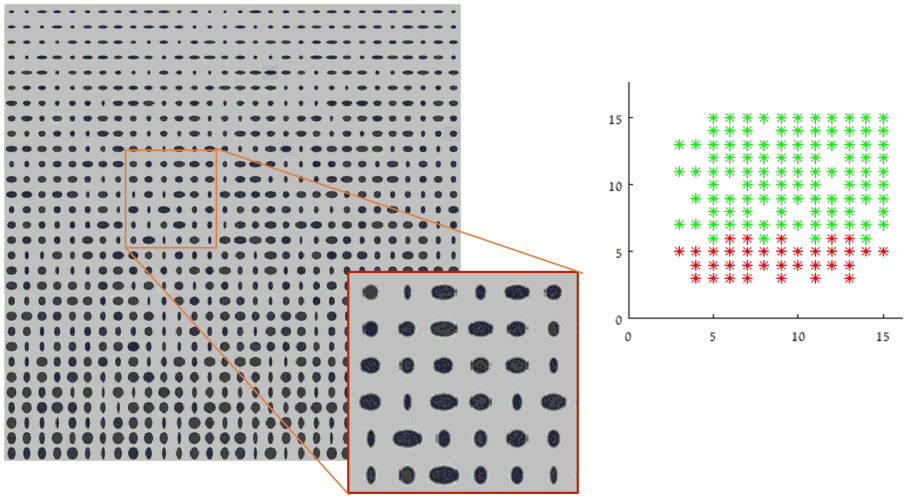


Figure 3.12: Left: Images of ellipses with varying radii that were sampled from a 2-dimensional manifold, prior to adding noise, which will form the P set. Right: a graph depicting the radii of the ellipses, with the coordinates of points given by these radii. The manifold samples are shown in green (P), while the initial set $Q^{(0)}$ is shown in red.

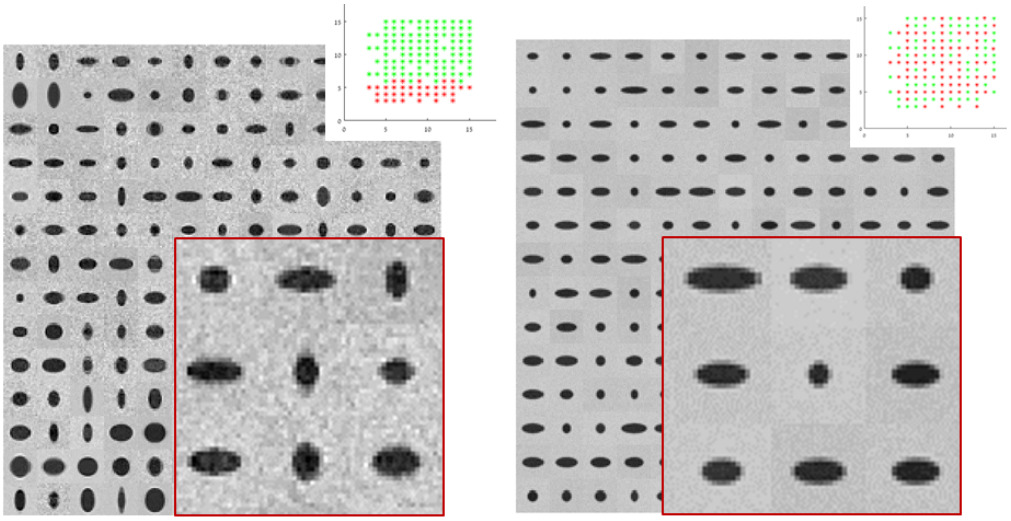


Figure 3.13: The samples that were used to reconstruct the manifold. Each side of the figure consists of an image of the samples, a zoomed-in area, and a graph of sample radii. The manifold samples are shown in green, while the initial set is shown in red. Left: the initial configuration of points sampled from the 2-dimensional manifold. Right: the manifold reconstruction configuration after 1000 iterations.

3.6 MLOP Denoise Benefits

The current section dealt both with manifold reconstruction and cleaning of high amounts of noise. The denoising property was induced by the first term in (3.1), which performs smoothing of p_j samples in the neighborhood of the examined point q_i . This term is inspired by the

L_1-median [125], and thus is robust to high amounts of noise. This fact was demonstrated in the "Robustness to Noise" subsection in [63], where the effect of various levels of noise on reconstruction accuracy was examined. The test demonstrated the robustness of the MLOP method to various amounts of noise magnitude (0, 0.1, 0.2, and 0.5), on a two-dimensional cylindrical structure embedded into 60-dimensions. The calculation of relative error of the reconstruction Q, against a densely sampled noise-free cylindrical structure, showed good results even at a noise level of 0.5 (with a relative error of 0.15). Thus, it is natural to use MLOP as a pre-processing step prior to performing mining tasks on the data.

In this section, we demonstrate the effectiveness of high-dimensional denoising in the case of local PCA. In our test we examine a set of points $X = \{x_i\}$, with a fill-distance h. We calculate PCA for each point x_i using its neighboring points x_j, which maintain the constraint $\|x_i - x_j\| < h$. Next, we extract the first eigenvector and evaluate its accuracy with respect to the first eigenvector of a PCA executed on clean reference data. Specifically, for each point x_i we find the closest point in the clean reference data and calculate the cosine distance between the corresponding PCA first eigenvectors (the error is given in degrees). Next, we determine the median of the errors stemming from all the points X. It is important to note that the error is tightly connected with the number of points in the set, with their fill-distance, and naturally with the noise levels. For example, on clean data with 160 points randomly sampled from a manifold, the error was 11.8, while with 7000 points, the error decrease to 0.2. This stems from the fact that taking a larger number of points in the neighbor of a point x_i leads to a more accurate eigenvector. This fact has to be taken into account in error analysis.

The numerical calculations were performed on the example of a two-dimensional cylindrical structure embedded into a 60-dimensional linear space. We sampled the structure using the parameterization

$$p = tv_1 + \frac{R}{\sqrt{2}}(\cos(u)v_2 + \sin(u)v_3)\,,$$

where $v_1 = [1, 1, 1, 1, 1, \ldots, 1]$, $v_2 = [0, 1, -1, 0, 0, \ldots, 0], v_3 = [1, 0, 0, -1, 0, \ldots, 0]$ ($v_1, v_2, v_3 \in \mathbb{R}^{60}$), $t \in [0, 2]$ and $u \in [0.1\pi, 1.5\pi]$. Using this representation, 816 uniformly distributed (in parameter space) points were sampled with uniformly distributed noise (i.e., $U(-0.2, 0.2)$). As

can be seen in Figure 3.14 left, after 500 iterations of the MLOP algorithm, the cylindrical structure was reconstructed with high accuracy (red points).

The experiments testing the efficiency of MLOP denoising were carried out on five data sets, all of size 160:

1. Noise-free data.
2. Noise data with additive noise of 0.1.
3. Data denoised by the MLOP from the data in item 2.
4. Noisy data with additive noise of 0.2.
5. Data denoised by the MLOP from the data in item 4.

The results for noise levels of 0.1 and 0.2 are presented in Figure 3.14 right. To achieve a robust error value, we performed ten bootstrap iterations for the "noise-free", as well as "noisy data" data-sets, where we randomly sampled the manifold, and calculated the median PCA error of the iterations. As expected, the effect of the MLOP denoising is to improve the accuracy of the local PCA calculations. One can see that the noise level has a small effect on the error (increasing it from 7.9 to 8.2, for the 0.1 and 0.2 noise level respectively). An additional benefit is that the accuracy of the denoised data is superior the one of the is noise-free data. The reason for this is the quasi-uniform manifold sampling which MLOP carries out accordingly due to the second term in (3.1), while the noise-free samples come from randomly sampled points (which not necessarily sample the manifold uniformly).

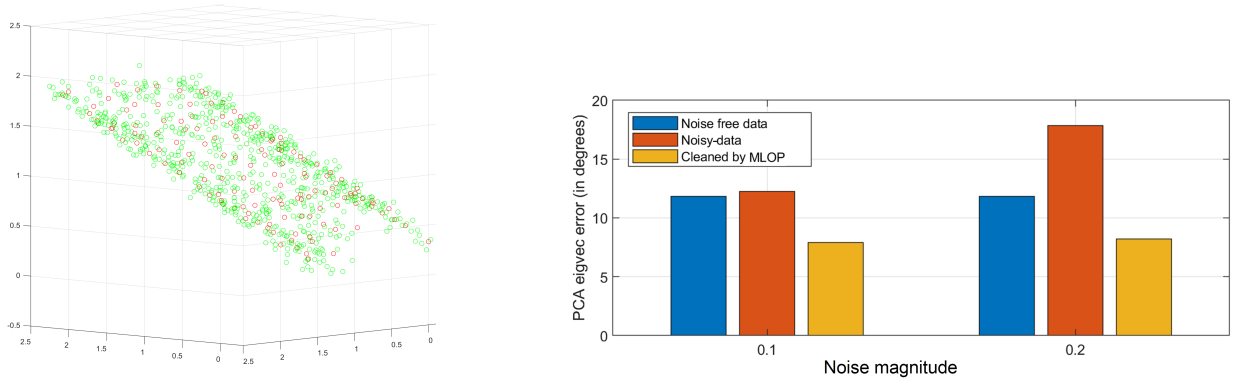


Figure 3.14: Left: Cylindrical structure, sampled with noise $U(-0.2, 0.2)$, and embedded in $\mathbb{R}^{60}$. The figure presents the first three coordinates of the points set. The point-set generated by the MLOP algorithm after 500 iterations, $Q^{(500)}$ (red) overlaying the noisy samples (green). Right: illustration of the MLOP denoising effect on the accuracy of PCA calculations. The graphs present the error of the first eigenvector of local PCA calculated on noise-free, noisy, and denoised data.

3.7 Discussion of the MLOP Method

In this chapter, we extended the LOP methodology to the high-dimensional setting. We modified the well-known surface reconstruction method [90] to deal with noisy data in high dimension. We provided theoretical grounds for the validity, accuracy, and efficiency of the method. In addition, we demonstrated the effectiveness of our approach on considering various examples with different manifold topologies with various amounts of noise. These examples show not only the data denoising effect but also the strength of the method in the task of manifold reconstruction without any prior knowledge about the parameterization of the manifold (nor of the noise model). In addition, as discussed above, having a quasi-uniform manifold sampling is a very important prerequisite for increasing the accuracy of various algorithms (as demonstrated for the local PCA case).

In the following chapters, we sketch several enhancements and applications of the MLOP method. Specifically, we will present and deal with the challenging problem of manifold repairing in high dimensions. In addition, discuss the problem of approximation of functions in high dimensions, and show how MLOP can shed new light on this topic. Subsequently, we will address the problem of manifold compressed sensing, which in contrast to the single signal compressed sensing gained little attention.

Chapter 4

Manifold Repairing in Low and High Dimensions

4.1 Introduction

Surface reconstruction gained a lot of attention in the recent years, with many methods proposed to address it (for detailed information see Section 2.1). As presented above, there are many challenges when dealing with surface reconstruction, one of them is missing information in the data, i.e., the presence of holes. During manifold acquisition, sampling the entire surface geometry is not always possible due to physical obstacles, occlusions, scanning angle issues, scanning costs, or to the fact that simply incomplete data is available. For example, in the case of acquiring oil and gas well data, where each drill is expensive, high resolution sampling can be extremely pricey. As a result, surface scanning can result in an incomplete surface with many holes, which can weaken the result of surface reconstruction. Therefore, a hole-free surface reconstruction is essential for both efficient data completion and minimizing data acquisition costs. In this chapter, we will address the problem of hole repairing in low and high-dimensions. We will start by presenting the problem and the available solutions in low-dimension, and later turn to the high-dimensional case.

The problem of hole filling can be formulated as followings: Given data sampled from a surface with holes, the task is to generate new points to reconstruct the hole regions. The

problem of filling holes and recovering the lost information in low dimension has been dealt by many scholars (for a concise survey, see [72]). The problem itself is usually treated as a two-step challenge, i.e., locating the hole, and then recovering the missing data. The solution for both steps can be roughly divided according to the way that the input data are given: as a point-cloud or as a mesh.

The hole identification problem is challenged by the real-life scenarios since usually no additional information regarding surface geometry is available. Subsequently, the task of recognizing the areas that need to be completed and the ones that should be omitted (since this is the surface geometry e.g., Figure 2.2), becomes ill-defined and non-trivial. If some information regarding the connectivity of the data is available, mesh representation comes in handy. Methods like these presented in [74, 79, 139] utilize this information for the hole identification by recognizing hollow edges, extract boundary via k-nearest neighbors search, or labeling boundaries based on the triangulation structure. On the other hand, if the data are given as a point-cloud, other methods are applicable. In [28] the authors suggested locating boundaries via sparse area heuristics, in [97] an enhanced k-nearest neighborhood is used, while in [12] a set of criteria are derived for automatic hole detection (the criteria rely on the symmetric k-ϵ neighborhood, their angle criterion and the deviation of a point and its neighborhood average, as well as the shape criterion, based on eigenvalues).

Once the hole is located, reconstructing the information inside the hole can be attained by various methods. In the case where the surface is represented as a mesh, the paper [139] uses a triangle growth procedure based on the boundary edge angle, in [74] the hole is filled under constraint on the boundary, in [79, 121] the holes are amended in a piecewise manner, i.e., by splitting holes, into smaller fragments and fixing latter, while in [7] a mesh repairing technique is proposed. When the surface is given as a point-cloud, the authors of [28] suggested fitting an algebraic surface patch to the neighborhood of the boundary; this avoids parametrization and allow to deal with holes with complicated topology. Other solutions rely on the minimal surface solution (Plateau's problem [73, 123]), diffusion [42], or an MLS-based point-cloud resampling technique [97, 126]. In addition, some methods address the surface repairing problem as the problem of reconstructing the surface from boundary conditions [101, 127].

The problem of completing missing information is presented not only in geometrical problem settings, but also in other fields. For example, in matrix computations [24], as well as in image processing. In the latter case, the problem is known as image inpainting, and various approaches were proposed for this case [14,40,105,134]. Of special interest is the elegant solution suggested in [14], where the hole edge information is propagated inside the hole using the gradient of the Laplacian in the direction of the edge. Due to the geometrical nature of the solution, it can be adopted to the surface case.

The existing solutions for surface reconstruction have a hard time of coping with challenges raised by real-life examples. First, they rely on some prior knowledge regarding the surface (e.g., a triangulation). In addition, these solutions do not deal with noise or outliers, which are usually present in the data. Moreover, since they were tailored for the surface repairing challenge, most of the known methods cannot be extended to the high-dimensional case (due to practical reasons connected with inefficiency of meshes in high dimensions and to curse of dimensionality). Due to these reasons, manifold repairing in high-dimension is still an open problem, and there is a need for a robust solution. In this paper we propose a simple, and efficient methodology, which is not based on deep learning algorithms (which at times are non transparent, and complicated to implement) in order to solve the challenging problem of manifold repairing.

In this chapter, we introduce a new approach, based on the MLOP method, to cope with manifold repairing in low and high-dimensional cases. The vanilla MLOP does not repair the manifold, since according to equation (3.1), it maintains the proximity to the original points (for example, see the Stanford bunny MLOP reconstruction in Figure 4.2 (A-B)). The attraction forces present in equation (3.1) prevent the MLOP from completing the missing data, since they compel the reconstruction to remain near the sample points. To overcome this, we suggest enhancing the MLOP method to address data repairing problems by adding another attraction force which will propagate the boundary points towards their convex hull. It should be noted that since hole identification is ill-posed, and the best way to fix the missing information is by the user manually identifying the holes that need to be amended. In what follows we first assume that the location of the hole is known, and later propose a method for

hole identification.

4.2 Manifold Repairing

The settings for the reconstruction and repairing problem are the following: We are given a noisy point-cloud $P = \{p_j\}_{j=1}^{J} \subset \mathbb{R}^n$ situated near a manifold $\mathcal{M}$ in $\mathbb{R}^n$, of unknown intrinsic dimension d and with incomplete data in a ball $B(c, r)$. We look for a solution in the form of a new point-set $Q = \{q_i\}_{i=1}^{I} \subset \mathbb{R}^n$, which will replace the given data P, be quasi-uniformly distributed, provide a noise-free approximation of $\mathcal{M}$, and will reconstruct the missing information in the given location. This is achieved by leveraging the well-studied weighted L_1-median [125], requiring a quasi-uniform distribution of points $q_i \in Q$ and propagating the boundary information inside the hole.

We propose *Repairing Manifold Locally Optimal Projection*(R-MLOP) method, which enhances the MLOP approach (introduced in Section 3) and is inspired by a local function approximation method. In the latter approach, the value of a function at a point x is estimated as a weighted average of the values of the function at nearby points x_i:

$$f(x) = \frac{\sum_{i \in I} w_i f(x_i)}{\sum_{i \in I} w_i},$$

where $w_i = e^{-\frac{\|x_i - x\|^2}{h^2}}$ and h is the fill-distance of the points $\{x_i\}_{i \in I}$.

The proposed R-MLOP method introduces a new term E_3 in equation (3.1), which plays the role of an attraction force, pulling points at the boundary of the hole towards their convex hull. We define this repairing E_3 term as

$$E_3 = \sum_{i' \in I} \bar{\tau}_{i'} \| q_{i'} \sum_{i \in I} \hat{w}_{i,i'} - \sum_{\substack{i \in I \\ i \neq i'}} \hat{w}_{i,i'} q_i \|^2, \tag{4.1}$$

where $\widehat{w}_{i,i'} = \exp\big\{ - \|q_i - q_i'\|^2 / h_2^2 \big\}$, $\bar{\tau}_i$ are balancing terms (see below), and the h_2 is the expected representative distance of the Q points, as defined in Subsection 3.3.2.

Let us note that in matrix notation the repairing term E_3 can be rewritten as the norm of the

graph Laplacian L:

$$E_3 = \sum_{i' \in I} \bar{\tau}_{i'} \|L(Q - Q_{i'})\|^2 \,, \tag{4.2}$$

where $L = D - W$, $D_{i'i'} = \sum_i \hat{w}_{i,i'}$, $\widehat{w}_{i,i'} = \exp\left\{ - \|q_i - q'_i\|^2 / h_2^2 \right\}$, Q is the matrix with the rows $\{q_i\}_{i \in I}$, while $Q_{i'}$ is the matrix with the rows $q_{i'}$.

We now define the *manifold reconstruction and repairing algorithm* as the minimization of the non-convex function

$$G(Q) = c_1 E_1(P, Q, T) + c_2 E_2(Q) + c_3 E_3(Q, T) \,, \tag{4.3}$$

where

$$E_1(P, Q, T) = \sum_{q_i \in Q} (1 - \bar{\tau}_i) \sum_{pj \in P} \|q_i - p_j\|_{H_\epsilon} w_{i,j} \tag{4.4}$$

and

$$E_2(Q) = \sum_{q_i \in Q} \sum_{q_{i'} \in Q \setminus \{q_i\}} \eta(\|q_i - q'_i\|) \hat{w}_{i,i'} \,, \tag{4.5}$$

and E_3 is given in (4.1).

The constants c_1, c_2 and c_3 replace the λ_i terms in (3.1) and balance the three terms in (4.3). They are calculated as $c_k = \underset{i \in I}{\text{median}}(\|\frac{\partial E_k}{\partial q_i}\|)$, for $k = 1, 2, 3$ after the first gradient descent iteration is completed. In addition, T is a vector of weights $\bar{\tau}_i \in [0, 1]$, each assigned to a q_i point, that balance between the attraction forces to P and the attraction towards the convex hull of the neighboring Q points. It is calculated only in the first iteration, and is based on the distance of the points q_i from the hole location. Specifically, near a hole the value of $\bar{\tau}_i$ is chosen to be close to 1 and thus E_3 gains more weight, while in areas where no repairing is required $\bar{\tau}_i$ is small and E_1 becomes dominant.

Given that the hole is located in a ball $B(c, r)$ (we will discuss how to approximate the parameters l and c later) the weight $\bar{\tau}_i \in [0, 1]$ associated to a point q_i is calculated by the rule

$$\bar{\tau}_i = \frac{\tau_i - \min(\tau_i)}{\max(\tau_i) - \min(\tau_i)}, \quad i \in I \tag{4.6}$$

where

$$\tau_i = e^{-\frac{\|q_i - c\|^2}{r^2}} . \tag{4.7}$$

The solution to the minimization problem (4.3) is found via the gradient descent algorithm:

$$q_{i'}^{(k+1)} = q_{i'}^{(k)} - \gamma_k \nabla G(q_{i'}^{(k)}) , \tag{4.8}$$

where the points $q_i{}^{(0)}$'s are randomly sampled from P, and the coefficients γ_k are given in equation (3.7).

The gradient of R-MLOP cost function G has the expression

$$\nabla G(q_{i'}^{(k)}) = (1 - \bar{\tau}_{i'}) c_1 \sum_{j=1}^{J} \left(q_{i'}^{(k)} - p_j \right) \alpha_j^{i'} - c_2 \sum_{\substack{i=1 \\ i \neq i'}}^{I} \left(q_{i'}^{(k)} - q_i^{(k)} \right) \beta_i^{i'} + \bar{\tau}_{i'} c_3 \frac{\partial E_3}{\partial q_{i'}} , \tag{4.9}$$

where $\alpha_j^{i'}$ and $\beta_i^{i'}$ are given in (3.4)-(3.5), and the gradient of E_3 is given by

$$\frac{\partial E_3}{\partial q_{i'}} = 2 \sum_{\substack{i \in I \\ i \neq i'}} (q_{i'} - q_i) \hat{w}_{i,i'} \sum_{\substack{i \in I \\ i \neq i'}} \left(1 - \frac{2}{h_2^2} \|q_{i'} - q_i\|^2 \right) \hat{w}_{i,i'} . \tag{4.10}$$

4.2.1 Multiple Hole Repair

In the case of multiple hole repair, it is necessary to modify the T weights in equation (4.3) so as to include all the information regarding the holes which need to be repaired. Let $B(c_k, r_k)$ be a set of balls, each with incomplete data which need to be amended. Then, the normalized weight $\bar{\tau}_i^k \in [0, 1]$ for a hole k are calculated as

$$\bar{\tau}_i^k = \frac{\tau_i^k - \min(\tau_i^k)}{\max(\tau_i^k) - \min(\tau_i^k)}, \quad i \in I \tag{4.11}$$

where

$$\tau_i^k = e^{-\frac{\|q_i - c_k\|^2}{r_k^2}} . \tag{4.12}$$

Subsequently, we redefine the normalized weights $\bar{\tau}_i \in [0, 1]$ which are used in (4.3) for multiple hole repairing, as

$$\bar{\tau}_i = \frac{\tau_i - \min(\tau_i)}{\max(\tau_i) - \min(\tau_i)} , \tag{4.13}$$

where

$$\tau_i = \prod_{k=1}^{K} \tau_i^k \,. \tag{4.14}$$

This definition of the weights which comprise T, incorporates the information from all the holes in a single coefficient attached to the points q_i.

4.3 Theoretical Analysis of the Method

In this section, we prove the validity of the proposed method. Specifically, given noisy data with a hole we study whether the proposed method amends the hole, and creates a noise-free reconstruction of the manifold which is of order $O(C_1h^2 + C_2r^2)$. In addition, we show that the order of complexity does not change with the new extension, compared to the vanilla MLOP. Last, we propose a method for identified holes that uses the intrinsic properties of the MLOP mechanism.

Definition 4.1. *Let $\epsilon > 0$ be a small number, and let $\mathcal{M}$ be a d-dimensional manifold with a hole H in a location $l \in \mathbb{R}^n$ and with the size bounded by r (where r is the smallest radius of a ball that contains the hole). Suppose that the scattered data points $P = \{p_j\}_{j=1}^{J} \subset \mathbb{R}^n$ were sampled near the manifold $\mathcal{M}$. Let $Q = \{q_i\}_{i=1}^{I}$ be the sought-for noise-free approximation of $\mathcal{M}$. Let $\bar{\tau}_i$ be defined as in* (4.6)*. Then, the* ϵ-neighborhood of the boundary of the hole *is the set of points q_i such that $\bar{\tau}_i > 1 - \epsilon$ (notice $\bar{\tau}_i \in [0, 1]$).*

4.3.1 Order of Approximation

Theorem 4.1 (Order of Approximation)**.** *Let $\mathcal{M}$ be a d-dimensional manifold in $\mathbb{R}^n$, where d is an unknown intrinsic dimension. Let H be a convex hole in $\mathcal{M}$ bounded by $B(c, r)$. Suppose that the scattered data points $\{p_j\}_{j=1}^{J}$ were sampled from the manifold $\mathcal{M}$ with noise, and h is defined using the definition from Section* 3.3.2*, and the h-ρ conditions are satisfied with respect to $\mathcal{M}$. Also let $Q^{(0)} = \{q_i^{(0)}\}_{i=1}^{I}$ be initial points set sampled from P. Then the order of approximation of* R-MLOP *to $\mathcal{M}$ is less than $C_1h^2 + C_2r^2$, where the constant C_1 depends on the curvature of the manifold outside the hole, and C_2 depends on the curvature inside the hole. It should be noted that although $r > h$, the dominant factor in the sum is the one which*

is dominant with respect to the C_i.

Proof. In regions far from the boundary of the hole, the $\bar{\tau}_{i'}$, defined in (4.6), are small and therefore the optimization function in (4.3) consists of only the first two terms, E_1 and E_2. Following Theorem 3.5, the order of approximation in these regions is $O(h^2)$, where $h = \max(h_1, h_2)$ and h_1 and h_2 are defined in Subsection 3.3.2 with respect to the P and Q points-sets.

In regions in the ϵ-neighborhood of the boundary of the hole, the representative distance between the points is bounded by $2r$ (we suppose that $h \leq r$, otherwise there is no hole). In these regions, there are points $q_i \in Q$ which are at a distance $O(r)$ from points in the set P. Therefore, the overall order of approximation at a new point is a combination of the two, namely $\leq C_1 h^2 + C_2 r^2$, with C_1, and C_2 constants. □

4.3.2 Method Validation

Theorem 4.2. *Let $\mathcal{M}$ be a d-dimensional manifold in $\mathbb{R}^n$, where d is an unknown intrinsic dimension. Let H be a convex hole in $\mathcal{M}$ bounded by $B(c, r)$. Suppose that the scattered data points $P = \{p_j\}_{j=1}^{J}$ were sampled near the manifold $\mathcal{M}$, h_1 is selected as defined in Section* 3.3.2*, and the h-ρ condition is satisfied with respect to $\mathcal{M}$. Also let $Q^{(0)} = \{q_i^{(0)}\}_{i=1}^{I}$ be initial set of points sampled from P. Then the gradient descent iterations for minimizing* (4.3) *result with a quasi-uniformly distributed point-set Q, that approximate and reconstruct the manifold, as well as recover missing information inside the hole.*

Proof. The definition of the R-MLOP method (4.3) together with (4.4) and (4.1) implies that in regions far from the hole, where $\hat{\tau}_i$ are small, the term E_3 term does not play a significant role. Therefore, in regions far from the hole the R-MLOP algorithm behaves like MLOP, and by Theorem 3.4 it converges and reconstructs the manifold at these regions.

In an ϵ-neighborhood of the boundary of H, where $\bar{\tau}_i$ is close to 1, the target function (4.3) consist of only the last two terms. The term E_2 is responsible for the uniform distribution of the points Q, and therefore we will consider here only the contribution of the term E_3 to hole

repairing.

Let us analyze the role of the term E_3, and prove that it indeed amends the hole. Namely, we show that at each iteration, the points in the ϵ-neighborhood of the boundary of the hole move towards the center of the hole. Let $\epsilon > 0$ be a small number, and let $q_{i'}^{(k)}$ be a point in the ϵ-neighborhood of the boundary of the hole H. We would like to prove that after a gradient descent step the point $q_{i'}^{(k+1)} = q_{i'}^{(k)} - \gamma_i \frac{\partial E_3}{\partial q_{i'}}$ moves towards the center of the hole.

The gradient in (4.9) can be rewritten as

$$\frac{\partial E_3}{\partial q_{i'}} = 2b_{i'} \sum_{\substack{i \in I \\ i \neq i'}} (q_{i'}^{(k)} - q_i^{(k)}) \hat{w}_{i,i'} \,, \tag{4.15}$$

where $b_{i'} = \sum_{\substack{i \in I \\ i \neq i'}} \left(1 - \frac{2}{h_2^2} \|q_{i'}^{(k)} - q_i^{(k)}\|^2\right) \hat{w}_{i,i'}$.

In what follows we analyze the direction of $\sum_{\substack{i \in I \\ i \neq i'}} (q_{i'}^{(k)} - q_i^{(k)}) \hat{w}_{i,i'}$ and the sign of $b_{i'}$.

1. **The weighed sum $\vec{a} = \sum_{\substack{i \in I \\ i \neq i'}} (q_{i'}^{(k)} - q_i^{(k)}) \hat{w}_{i,i'}$ is a vector pointing towards the direction of the center of the hole.** Indeed, the term $\sum_{\substack{i \in I \\ i \neq i'}} (q_{i'}^{(k)} - q_i^{(k)}) \hat{w}_{i,i'}$ is a weighted sum of vectors, each directed from $q_i^{(k)}$ to $q_{i'}^{(k)}$ (see Figure 4.1 left, vectors marked in black). We rewrite this sum according to the direction of the vector $\vec{r}_{i',i} = q_{i'}^{(k)} - q_i^{(k)}$, i.e., towards the center of the hole or not, using a dot product. Let $\vec{v}$, defined as $\vec{v} = \frac{c - q_{i'}^{(k)}}{\|c - q_{i'}^{(k)}\|}$, be a unit vector originating at a point $q_{i'}^{(k)}$ and pointing in the direction of the center of the hole. Then $\vec{a}$ can be decomposed into a sum of vectors in the same direction as $\vec{v}$, and a sum of vectors pointing in a different direction,

$$\vec{a} = \sum_{\frac{\vec{r}_{i',i}}{\|\vec{r}_{i',i}\|} \cdot \vec{v} \geq 0} \vec{r}_{i',i} \hat{w}_{i,i'} + \sum_{\frac{\vec{r}_{i',i}}{\|\vec{r}_{i',i}\|} \cdot \vec{v} < 0} \vec{r}_{i',i} \hat{w}_{i,i'} \,.$$

   The hyperplane $x \cdot \vec{v} = 0$ separates between vectors $\vec{r}_{i',i}$, these that are in the direction of the center of the hole, from these that are pointing in different direction. Assuming uniform distribution of the points, since there are no points within the hole, there are more points that satisfy the $\frac{\vec{r}_{i',i}}{\|\vec{r}_{i',i}\|} \cdot \vec{v} \geq 0$ condition, then the second one. As a result, the vector $\vec{a}$ point towards the center of the hole.

2. **The term $b_{i'}$ satisfies the inequality $b_{i'} < 0$.** Let us define $b(r) = \left(1 - \frac{2}{h_2^2} r^2\right) e^{-\frac{r^2}{h^2}}$, where $r = \|q_{i'} - q_i\|$. We analyze the behavior of the function $b(r)$, and plot it in Figure 4.1 (right). Based on Remark 3.4, the term $e^{-\frac{r^2}{h^2}}$ vanishs for $r > 2\sqrt{2}h_2$. In addition, we note that $b(r) \geq 0$ for $r \in [0, \frac{h_2}{\sqrt{2}}]$; $b(r) < 0$ for $r \in (\frac{h_2}{\sqrt{2}}, 2\sqrt{2}h_2]$ and it reaches its extrema at $r = 0$ and $r = \pm\frac{\sqrt{3}h_2}{\sqrt{2}}$. Using these observations, we rewrite the definition of $b_{i'}$ as

$$b_{i'} = \sum_{0 \leq r \leq \frac{h_2}{\sqrt{2}}} b(r) - \sum_{\frac{h_2}{\sqrt{2}} < r \leq 2\sqrt{2}} |b(r)|\,, \tag{4.16}$$

In what follows we bound the first term in (4.16) from above by c_1, and the second term from bellow, by c_2 and show that $c_1 < c_2$.

Let N_0 be the estimated number of points within ball $B_0 = B(q_{i'}^{(k)}, \frac{h_2}{\sqrt{2}})$, then the first term in (4.16) can be bounded as

$$\sum_{0 \leq r \leq \frac{h_2}{\sqrt{2}}} b(r) \leq N_0 \times \max_{r \in [0, \frac{h_2}{\sqrt{2}}]} b(r) \leq N_0\,.$$

We now turn to bounding the second term in (4.16) from bellow. We divide the interval $(\frac{h_2}{\sqrt{2}}, 2\sqrt{2}h_2]$ in to four intervals, i.e., for $A_1 = (\frac{h_2}{\sqrt{2}}, A]$, $A_2 = [A, \frac{\sqrt{3}h_2}{\sqrt{2}}]$, $A_3 = [\frac{\sqrt{3}h_2}{\sqrt{2}}, C]$, $A_4 = [C, 2\sqrt{2}h_2]$, where the points A and C are the center of the intervals $[\frac{h_2}{\sqrt{2}}, \frac{\sqrt{3}h_2}{\sqrt{2}}]$, $[\frac{\sqrt{3}h_2}{\sqrt{2}}, 2\sqrt{2}h_2]$ respectfully (ie., $A = \frac{(1+\sqrt{3})h_2}{2\sqrt{2}}$, $C = \frac{(4+\sqrt{3})h_2}{2\sqrt{2}}$, see Figure 4.1 right). Let us consider four balls all centered at $q_{i'}^{(k)}$ with varying radii $B_1 = B(q_{i'}^{(k)}, A)$, $B_2 = B(q_{i'}^{(k)}, \frac{\sqrt{3}h_2}{\sqrt{2}})$, $B_3 = B(q_{i'}^{(k)}, C)$ and $B_4 = B(q_{i'}^{(k)}, 2\sqrt{2}h_2)$, as well as the estimate number of points in these balls N_1, N_2, N_3 and N_4, respectfully.

In these notations, we notice that

$$\begin{aligned}\sum_{\frac{h_2}{\sqrt{2}} < r \leq 2\sqrt{2}} |b(r)| \geq \sum_{r \in A_2} |b(r)| + \sum_{r \in A_3} |b(r)| \geq N'_2 \min_{r \in A_2} |b(r)| + N'_3 \min_{r \in A_3} |b(r)| \\ \geq N'_2 b(A) + N'_3 b(C) \geq 0.3N'_2 + 0.1N'_3 \end{aligned} \tag{4.17}$$

where N'_2 be the estimate of the number of points in $B_2 \backslash B_1$, and the estimate of the number of points N'_3 in $B_3 \backslash B_2$, i.e., $N'_2 = N_2 - N_1$ and $N'_3 = N_3 - N_2$.

In order to estimate N'_1, and N'_2 in terms of N_0 we assume that the proportion of the

number of points in a support does not change with radius changes. Let V_0, V_1, V_2, V_3 be the volumes of the of the balls B_0, B_1, B_2, B_3 respectively. Then, relying on the proportion consistency assumption we estimate N_1 using $\frac{V_0}{V_1} = \frac{N_0}{N_1}$, where the volume of a ball with radius $\frac{h_2}{\sqrt{2}}$ in $\mathbb{R}^d$ is $V_0 = \pi^{d/2}(\frac{h_2}{\sqrt{2}})^d/c(d)$, and the volume of a ball with radius A is $V_1 = \pi^{d/2}A^d/c(d)$ (where c is Euler's gamma function). Thus, $N_1 = (\frac{(1+\sqrt{3})h_2}{2\sqrt{2}})^d N_0$. Using similar considerations $N_2 = \sqrt{3}^d N_0$, and $N_3 = (\frac{(4+\sqrt{3})h_2}{2\sqrt{2}})^d N_0$. Finally, after calculating N'_2 and N'_3, and substituting to (4.17) one receives that the second term is larger than $c_2 = c_3 N_0$, where $c_3 > 1$. As a result the second term in (4.16) is dominant and $b_{i'} < 0$.

We conclude that the weighed sum $\sum_{\substack{i\in I \\ i\neq i'}} (q_{i'} - q_i)\hat{w}_{i,i'}$ is directed towards the hole, b is negative, thus $\frac{\partial E_3}{\partial q_{i'}}$ is pointing outside the hole. In addition, our numerical investigation showed that taking a small step size of $0.25\gamma_k$, where γ_k is defined in 3.7, result in a positive γ_k during the gradient descent iterations for points in the ϵ-neighborhood of the boundary of the hole. As a result, the point $q_{i'}^{(k)}$ in (4.8) moves towards the hole and recovers missing information.

□

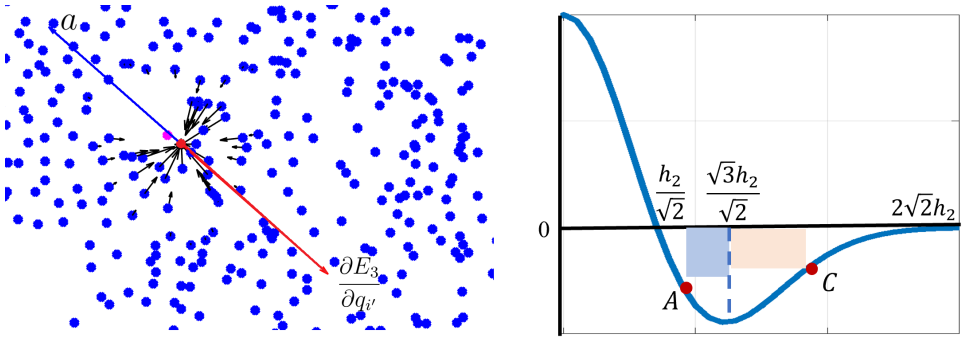


Figure 4.1: Illustration of the two stages in the proof of R-MLOP validity. Left: Analysis of the direction of movement of a $q_{i'}^{(k)}$ point (in red) after an amendment step (the magenta point) on a real case example. The forces that the neighboring points q_i accent on the current point are in indicated black. The weighted sum in the direction $\vec{a}$ is shown in blue, while the gradient of E_3 is shown in red. Thus the point $q_{i'}^{(k)}$ moves opposite to that of the gradient, towards the center of the hole. Right: A plot of the $b(r)$ function, along with important key points which aid in bounding $b_{i'} < 0$.

4.3.3 Complexity of the R-MLOP Algorithm

The R-MLOP algorithm is based on the MLOP algorithm, with the addition of the term E_3. As specified in Corollary 3.2, the complexity of MLOP is $O(nmJ + kI(nm\hat{I} + \hat{J}))$, where n

is the dimension of the data and m is the dimension to which one reduces the dimension of the data for the procedure of calculating the norm ($m \ll n$). Thus, we need to evaluate the complexity of the term E_3. The gradient of this term involves the differences $q_{i'} - q_i$ between a given point $q_{i'}$ and points $q_i \in Q$. This calculation is performed for each iteration. Thus, the total complexity of the R-MLOP method is $O(nmJ + kI(nm\hat{I} + \hat{I} + \hat{J}))$.

Corollary 4.1. *Given a set of points $P = \{p_j\}_{j=1}^{J}$ sampled near a d-dimensional manifold $\mathcal{M} \in \mathbb{R}^n$, let $Q = \{q_i\}_{i=1}^{I}$ be a set of points which will provide the desired the manifold reconstruction and repairing. Then the complexity of the* R-MLOP *algorithm is $O(nmJ + kI(nm\hat{I} + \hat{J}))$, where the number of iterations k is bounded as in Theorem* 3.7*, m is the smaller dimension to which one reduces the dimension of the data to, $m \ll n$ as described in* 3.3.1*, and $\hat{I}$ and $\hat{J}$ are the numbers of points from the sets Q and P, respectively, in the support of the weight function $w_{i,j}$. Therefore, the approximation is linear in the ambient dimension n, and does not depend on the intrinsic dimension d.*

4.4 Approximating the Locations of the Holes and Their Volume

Hole identification is an ill-posed problem since usually the geometry of the manifold is unknown, therefore it is hard to know whether a hole really exists or the manifold was sampled poorly. Therefore, the best scenario for fixing the missing information is when one can manually identify the holes that need to be amended. However, in real-life scenarios, this information is not always available, and thus estimating the location and radius of the hole are required.

In this subsection, we propose a method that relies on point density consideration to identify the boundary of the hole. It is reasonable to assume that near a boundary of a hole the density of the points drops. However, low density does not characterize only regions close to a hole but also can stem from non-uniform sampling. In addition, for an open manifold, low-density values can also characterize the bounders of the manifold. As described in the introduction Section 4.1 the problem of hole identification was addressed with various methods, where the most prominent one, for the point-cloud case, is the k-nearest neighbors considerations, that is

closely related to the point density. However, the challenge of dealing with manifold boundaries was not taken into account.

In what follows we propose a procedure for identifying the boundary of a hole in a manifold, while taking into account the challenges mentioned above. Given scatter data $P = \{p_j\}_{j=1}^{J}$ sampled from a manifold $\mathcal{M}$, which satisfy the h-ρ condition with respect to $\mathcal{M}$. We propose addressing the hole identification in several phases each dealing with another challenge raised by low density:

1. Quasi-uniform sample the manifold using the vanilla MLOP method.
2. Identification of the boundary of the manifold.
3. Identification of the boundary of the hole.
4. Estimate the location of the hole and its volume.

First, we apply the MLOP as a pre-step in order to overcome the case of non-uniform sampling of the P point data. Next, we classify the quasi-uniform point-set Q, produced by the MLOP, to be either: a) boundary of the manifold; b) boundary of the hole; c) neither of them. Let $h_{0,2}$ be the fill-distance of the Q points (as defined in definition 3.2), and let ρ_2 be the density of Q that holds the (3.1) inequality, i.e., $\#\{Q \cap \bar{B}(y, kh_{0,2})\} \leq \rho_2 k^n, \quad k \geq 1, \quad y \in Q$. We notice that on the manifold boundary the number of points does not grow at the same rate as k grows. Thus, in order to identify the boundary of the manifold one needs to analyze the change in the number of points for kh for $k = 1, 2$, and select the q_i points that their change in $\#\{Q \cap \bar{B}(y, kh_{0,2})\}$ is insignificant. Last, the points on the boundary of the hole are identified as the ones with a low number of points, that do not belong to the manifold boundary. Algorithm 3 summarize this process.

Algorithm 3 Estimating the Location of the Hole

1: **Input:** $P = \{p_j\}_{j=1}^{J} \subset \mathbb{R}^n$

2: **Output:** Parameters of the hole r, c

3: Create quasi-uniform points set Q via vanilla MLOP.

4: Identify points on the manifold boundary, by calculating the number of points for h, $2h$, and use the notation $a_i = \frac{\#\{Q \cap \bar{B}(y,2h_{0,2})\}}{\#\{Q \cap \bar{B}(y,h_{0,2})\}}, \quad y \in Q$ to identify the boundary points q_i such that $a_i < median(a_i)$.

5: Identification the points on the boundary of the hole, such that the number of points at this points is low, and they does not belong to the manifold boundary.

6: Clean outlines in the set of boundary points.

7: Approximate the location of the hole and its volume. Estimate the radius r of the hole as half of the max distance between the boundary points, and the center c of the hole as the center of mass of the boundary points.

4.5 Numerical Examples

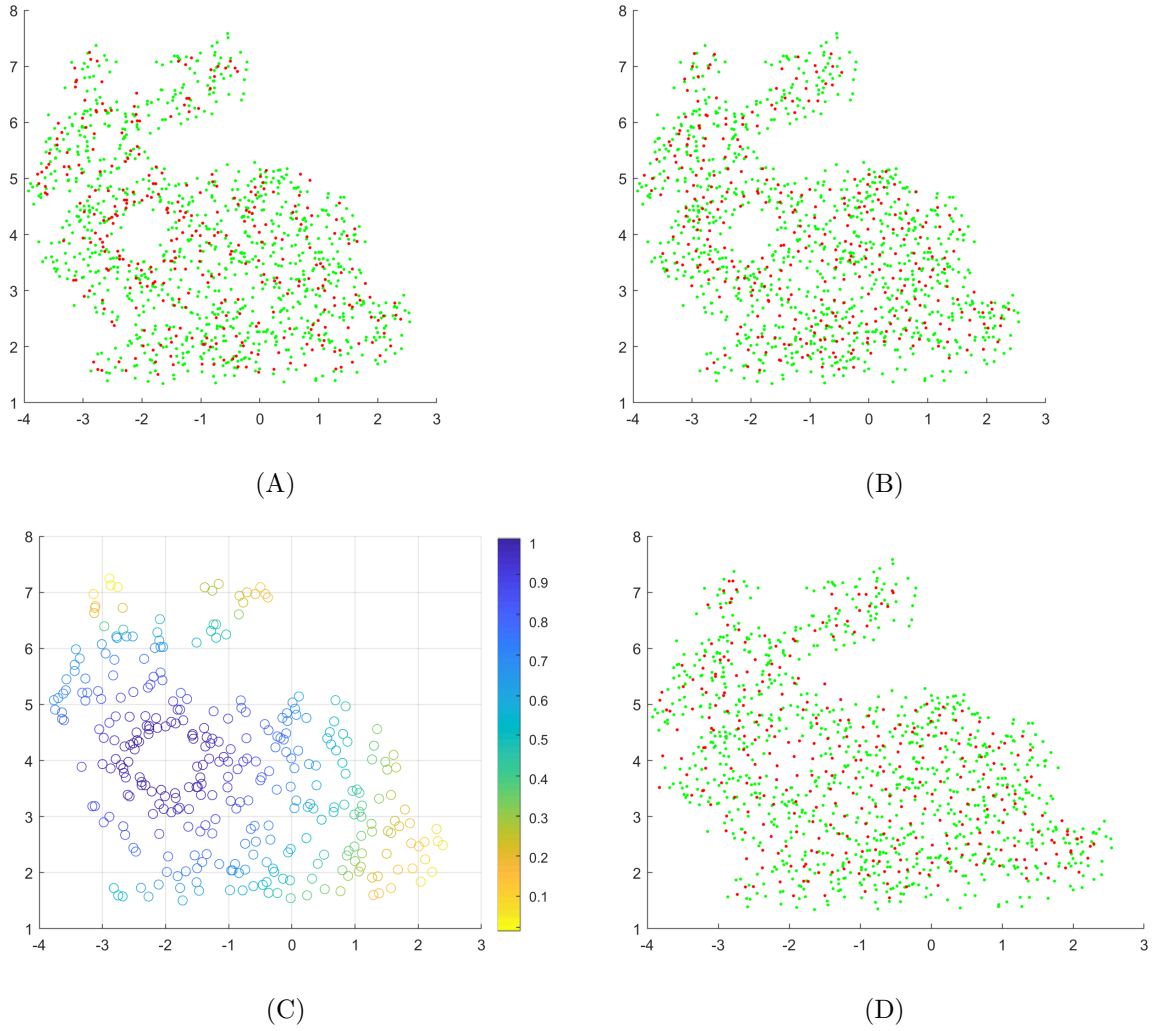
In this subsection, we demonstrate the efficiency of our method on 3D surfaces as well as on manifolds in higher dimension, for single and multiple hole repairing. The numerical setup was usually built by sampling known manifold data, and artificially creating holes in it. It should be noted that in all of the examples below we relied on the fact that the location of the hole and its radius were known.

4.5.1 Data Repairing in Low-Dimensional Space

We start by demonstrating our data completion method on 3D scenes of the bunny and the dragon taken from the Stanford Scanning Repository [86]. We loaded the two published models, and randomly sampled 1000 points which served as the initial P points. Next, a hole was artificially created by removing points from a chosen location (the hole in the bunny was in its neck, while the hole in the dragon was in his head), which resulted in about 950 points. Later, a subset of 350 points was sampled to construct the Q set. In addition, we slightly increased

the density of the Q points near the boundary of the hole. An illustration of the initial settings for the repairing algorithm can be found in Figure 4.2 (A), Figure 4.3 left.

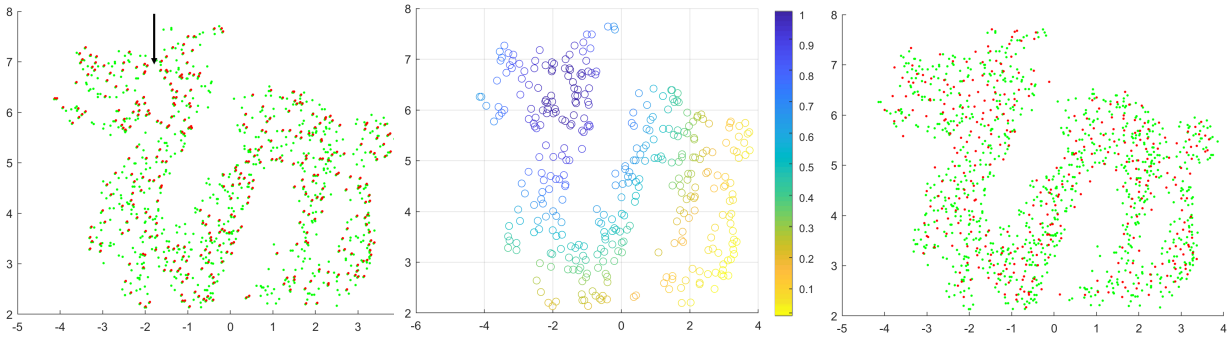
In our numerical experiment, we compared the result of applying the vanilla MLOP with the result of R-MLOP. First, we applied the plain MLOP algorithm on our data, which resulted in a quasi-uniform sampling (Figure 4.2 (B)). As expected, the reconstruction maintained its proximity to the P points and did not recover the missing information. Next, using the hole information, we calculated the proximity coefficient T in (4.6) (its values are presented in Figure 4.2 (C), Figure 4.3 middle). Finally, we executed the repairing algorithm described above. The amending result after 30 iterations can be seen in Figure 4.2 (D), Figure 4.3, right. One can see that the existing holes were repaired successfully with uniform sampling.



(A) (B)

(C) (D)

Figure 4.2: Amending the scattered data for the Stanford bunny. (A) The initial scattered data in size of 1K sampled from the bunny model (the initial P points in green, the initial Q points in red). (B) The picture produced by the plain MLOP, which results in a quasi-uniform sampling of the bunny: the hole is not amended. (C) The weights T of the hole which are closer to 1. (D) The hole in the bunny was amended using the R-MLOP algorithm.

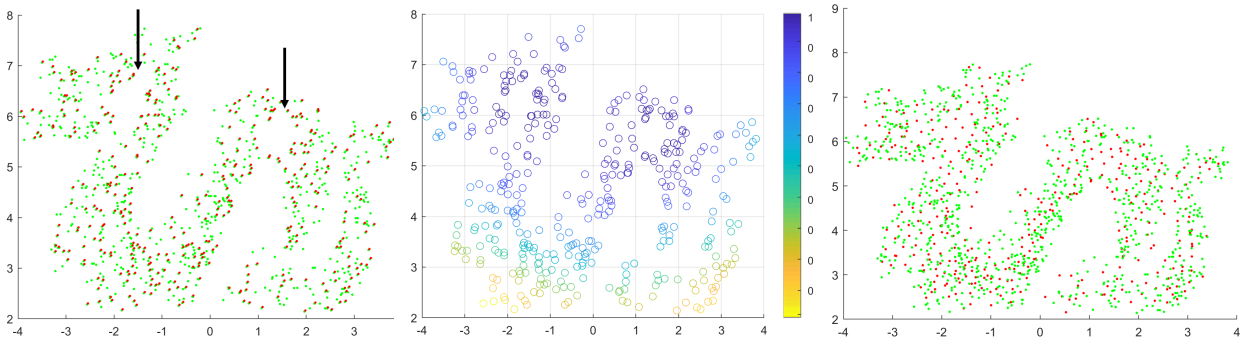


(A) (B) (C)

Figure 4.3: Amending the scattered data for the Stanford dragon. Left: The initial scattered data sampled from the dragon model, with a hole artificially created in its head (the initial P points in green, the initial Q points in red). Middle: The weights T of the hole, which are closer to 1 near the hole location. Right: The result produced by the R-MLOP algorithm.

4.5.2 Multiple Holes Repair

In the next example, we applied the multiple holes repair methodology to the Stanford dragon, which was sampled with P and Q as described above. Two holes locations were selected, one in the dragon head and the other in its tail, and points around them were removed (see Figure 4.4, left). Subsequently, we calculated the enhanced proximity weights T; are presented in Figure 4.4, middle. The result produced by the surface amending algorithm is presented in Figure 4.4, right. One can see that missing information on the two holes was successfully recovered with quasi-uniform sampling.



(A) (B) (C)

Figure 4.4: Amending the scattered data for the Stanford dragon. Left: The initial scattered data sampled from the dragon model, with two holes artificially created, one in its head and another in its tail - marked with arrows (the initial P points in green, the initial Q points in red). Middle: The weights of the hole T which are closer to 1 near the hole location. Right: The result produced by the R-MLOP algorithm.

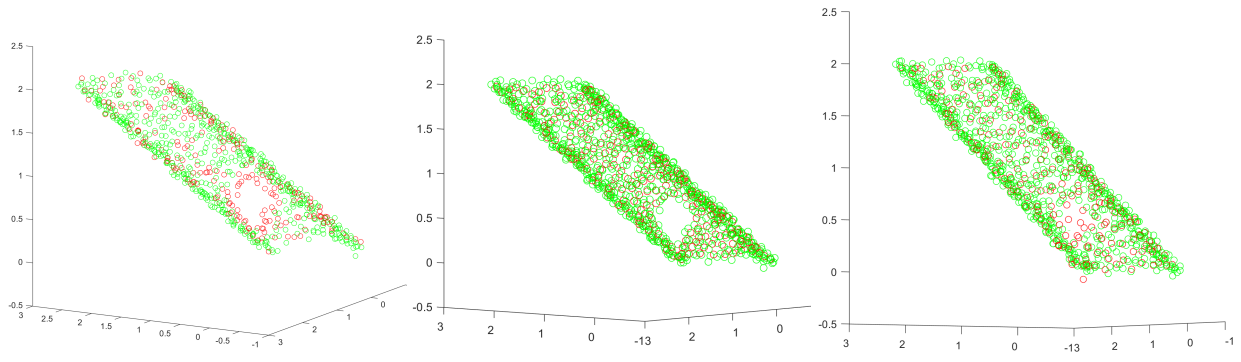
4.5.3 Manifold Repairing in High-Dimensional Space

In this subsection, we demonstrate the data completion method on several examples of a low-dimensional manifold embedded in a high-dimensional space. Specifically, we embedded the two-dimensional and six-dimensional cylindrical structure into $\mathbb{R}^{60}$, as well as a cone structure with multiple holes, which was embedded into $\mathbb{R}^{60}$. In both cases we artificially created a hole in the data around certain point (Figure 4.5, Figure 4.6 and Figure 4.7, left). We then applied the R-MLOP method for data completion. The details of the data creation can be found below.

First, we embedded a two-dimensional cylindrical structure into a 60-dimensional linear space. We sampled the structure using the parameterization

$$p = tv_1 + \frac{R}{\sqrt{2}}(\cos(u)v_2 + \sin(u)v_3)\,,$$

where $v_1 = [1, 1, 1, 1, 1, \ldots, 1]$, $v_2 = [0, 1, -1, 0, 0, \ldots, 0], v_3 = [1, 0, 0, -1, 0, \ldots, 0]$ ($v_1, v_2, v_3 \in \mathbb{R}^{60}$), $t \in [0, 2]$ and $u \in [0.1\pi, 1.5\pi]$. Using this representation, 790 uniformly distributed (in parameter space) points were sampled with uniformly distributed noise (i.e., $U(-0.1, 0.1)$). The initial Q set was constructed by randomly sampling 230 points (Figure 4.5, left). Next, we applied the plain MLOP algorithm on our data, which resulted in a quasi-uniform sampling (Figure 4.5, middle)). As expected, the reconstruction maintained its proximity to the P points, and did not recover the missing information. Finally, we executed the R-MLOP algorithm, and after 70 iterations the manifold was amended with quasi-uniformly sampling (Figure 4.5, right).



(A) (B) (C)

Figure 4.5: Cylindrical structure embedded into a 60-dimensional space. The first three coordinates of the point-set are shown. Left: Scattered data with uniformly distributed noise $U(-0.1; 0.1)$ (green), and the initial point-set $Q^{(0)}$ (red), with an artificial hole created. Middle: The point-set generated by the MLOP algorithm. Left: The result produced by R-MLOP algorithm.

4.5.4 Six-dimensional cylindrical structure

Next, we tested our method on higher-dimensional manifolds by utilizing an n-sphere to generate an $(n+1)$-dimensional cylinder (in the example of the two-dimensional cylinder, we used a circle to generate the structure). Here, we utilized a five-dimensional sphere to build a six-dimensional manifold, using the parameterization

$$x_1 = R\cos(u_1)\,, \quad x_2 = R\sin(u_1)\cos(u_2), \quad \ldots, \quad x_6 = R\sin(u_1)\sin(u_2)\cdots\sin(u_5)\sin(u_6)\,.$$

We then embedded the sampled data in a 60-dimensional space by the parametrization

$$p = tv_0 + R^2[x_1, x_2, x_3, x_4, x_5, x_6, 0, \ldots, 0]\,, \tag{4.18}$$

where $R = 1.5$, $t \in [0, 2]$, $u_i \in [0.1\pi, 0.6\pi]$, and $v_0 \in \mathbb{R}^{60}$ is a vector with 1's in positions $1, ..., d+1$ and 0 in the remaining positions. We randomly sampled the P-points from the six-dimensional cylindrical structure and artificially created a hole in a known location and of known size. We embedded the sampled data in a 60-dimensional space. This process resulted in 1180 points in the P-set. Next, uniformly distributed noise $U(-0.2; 0.2)$ was added to the points, and then a subset of 460 points was sampled to construct the Q set. To avoid trying to visualize a six-dimensional manifold, we plot here the cross-section of the cylindrical structure in three dimensions.

Our numerical experiment included several executions. First, we applied the plain MLOP algorithm on our data, which resulted in quasi-uniform sampling (Figure 4.6 (B)). As expected, the reconstruction maintained its proximity to the P points, and did not recover the missing information. Next, using the hole information we calculated the proximity coefficient T in (4.6) (the resulting values are presented in Figure 4.6 (C)). Finally, we executed the repairing algorithm described above. The amending result after 100 iterations is shown in Figure 4.6 (D). As one can see, the existing holes were repaired successfully with the high-dimensional cylindrical structure with uniform sampling.

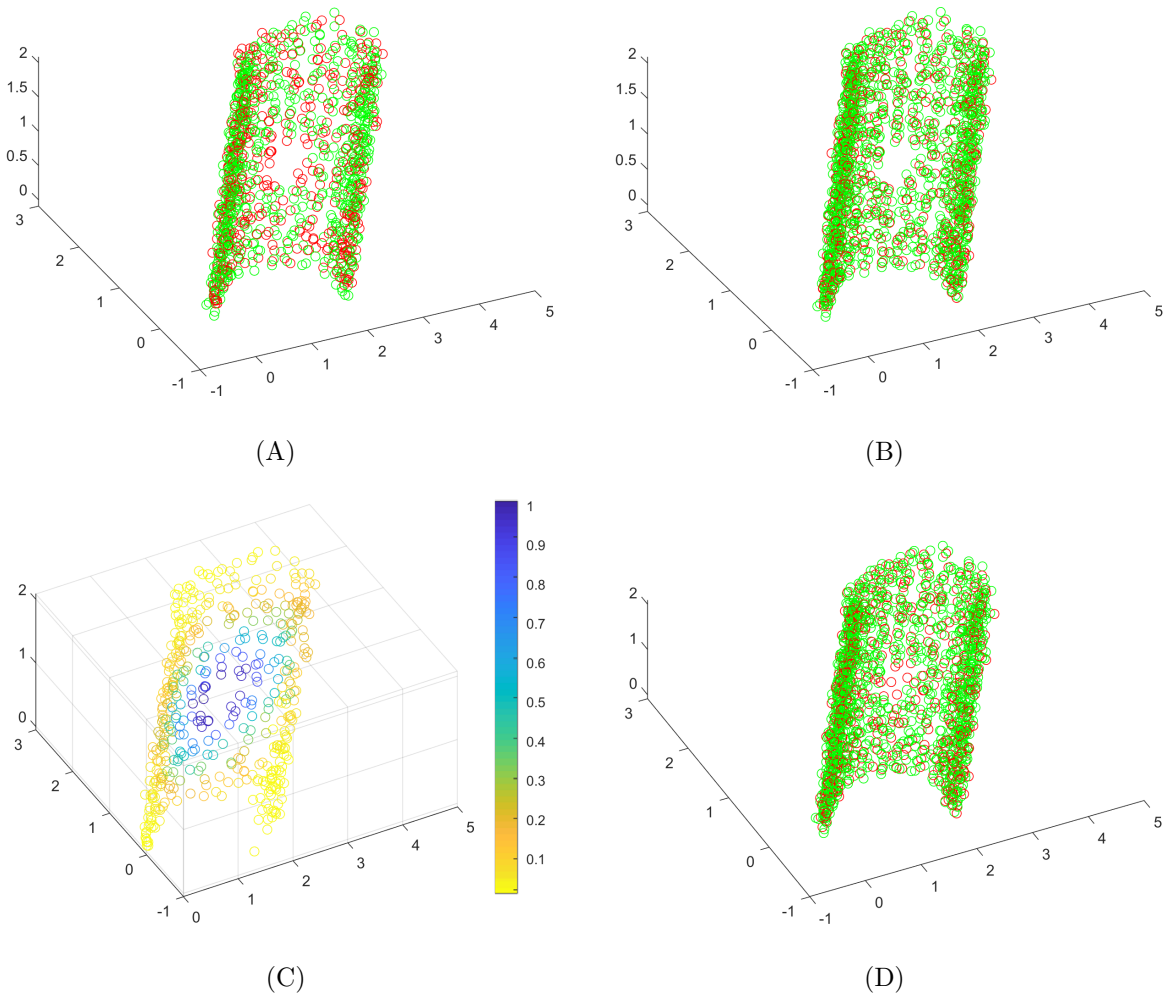


Figure 4.6: Amending the scattered data from a six-dimensional cylindrical structure embedded into a 60-dimensional space. The cross section of the cylindrical structure by a hyperplane in which the first four coordinates are greater then -0.5 is shown. (A) The initial scattered data of size 1180 sampled from the manifold (the initial P points in green, the initial Q points in red). (B) The result after applying the plain MLOP, which produced a quasi-uniform sampling of the six-dimensional cylindrical structure, the hole is not amended. (C) The weights of the hole T which are closer to 1 near the hole location. (D) The result produced by R-MLOP algorithm.

4.5.5 Multiple Holes Repair

Last, we demonstrate the ability of the R-MLOP algorithm to cope with a geometric structure of different dimensions at different locations and with multiple holes. Here we combined a 3-dimensional manifold, namely, a cone structure, with a one-dimensional manifold, namely, a line segment. This object was embedded into a 60-dimensional linear space. We used the cone

parameterization

$$p = tv_1 + \frac{e^{-R^2}}{\sqrt{2}}(\cos(u)v_2 + \sin(u)v_3)\,,$$

where $v_1 = [1, 1, 1, 1, 0, \ldots, 0], v_2 = [0, 1, -1, 0, 0, \ldots, 0], v_3 = [1, 0, 0, -1, 0, \ldots, 0]$, $(v_1, v_2, v_3) \in \mathbb{R}^{60}$, $t \in [0, 2]$, $R \in [0, 2.5]$, and $u \in [0.1\pi, 1.5\pi]$. We sampled 850 points from the structure with added uniformly distributed noise of magnitude 0.2. The initial set $Q^{(0)}$ of size 140 was randomly sampled (Figure 4.7 (A)). Next, we applied the plain MLOP algorithm on our data, which produced a quasi-uniform sampling (Figure 4.7 (B))). As expected, the reconstruction maintained its proximity to the P points, and did not recover the missing information. Later, we calculated the coefficient T using equation (4.6) (its values range from 0 to 1 and correspond to yellow and blue color in Figure 4.7 (C)). Finally, we executed the R-MLOP algorithm, and after 200 iterations the manifold was amended with a quasi-uniformly sampling (Figure 4.7 (D)).

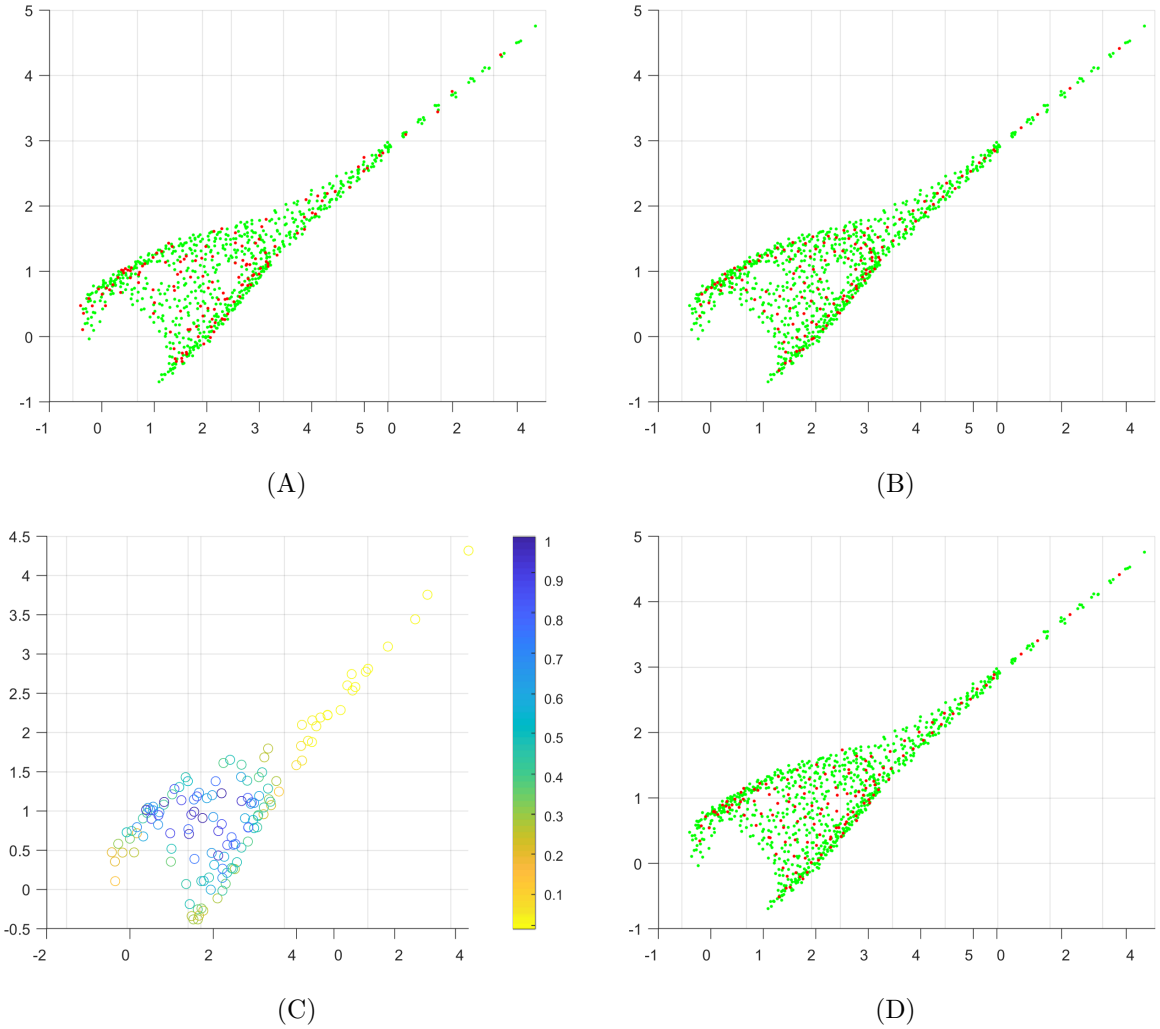


Figure 4.7: Cone structure embedded into a 60-dimensional space. The first three coordinates of the point-set are shown. (A) Scattered data with uniformly distributed noise $U(-0.1; 0.1)$ (green), and the initial point-set $Q^{(0)}$ (red), with an artificial hole created. (B) The point-set generated by the MLOP algorithm. (C) The weights of the hole T which are closer to 1 near the hole location. (D) The result produced by the R-MLOP algorithm.

Chapter 5

Approximation of Functions on a Manifold in High Dimensions

5.1 Introduction

In this chapter, we consider the following formulation of the problem of approximation of functions in a high dimensional space. Let $f : \mathbb{R}^n \to \mathbb{R}^s$ be a function, and let $\{f(x_i)\}_{i=1}^K$ be its values on a given sample set of points $\{x_i\}_{i=1}^K \subset \mathbb{R}^n$ with noise present both in the domain of the function and in its codomain. The goal of the approximation is to estimate the values of the function at a new set of points. While in low dimensions numerous methods were suggested to solve this problem (e.g., splines, or Moving Least-Squares [83]), in high dimensions this is a challenging task due to the presence of noise and the curse of dimensionality. For instance, with respect to the latter challenge, if one merely assumes that the function is smooth, then approximation rates deteriorate severely with the growth of the dimension, the reason being that the amount of sampled data should grow exponentially with respect to the dimension if one wishes to maintain the same order of approximation.

We categorize high-dimensional approximation methods according to whether domain of the function to be approximated. If no assumptions are made on the data domain, several methods were suggested. For example, solutions which treat non-smooth multivariate functions, [5], are based on sparse occupancy trees [16], Radial Basis Functions [47] (which we will discuss in

detail below), or address the problem in the case where the values of the function lie on a manifold [71].

In many situations, the high-dimensional data reside on a low-dimensional manifold, and this information can be exploited to improve the approximation via one of the following two approaches: approximating in low dimension after dimension reduction, or alternatively approximating in high dimension. At times, reducing the dimension (e.g., in PCA [103], Multidimensional Scaling [39], Linear Discriminant Analysis [68], Locality Preserving Projections [75], Locally Linear Embedding [107], ISOMAP [122], Diffusion Maps [36], and Neural Networks in their general form, [88]) can lead to a better approximation (in terms of handling the challenge of the dimensionality, as well as the noise in the data). However, it may be non-efficient if the data volume is very large, and in addition, may result in information loss (due to some assumptions that need to be made on the data, e.g., regarding the data geometry, or the intrinsic dimension).

On the other hand, the assumption that the data reside on a manifold can be utilized in order to improve the approximation in high dimensions. Approximation of functions on manifolds is studied using local polynomials [15], wavelets [37], local linear regression [15], or neural networks [6, 31, 108]. For smooth functions on $[0,1]^N$ which depend on a much smaller number l of variables, a solution was suggested in [43]. In addition, a recent paper proposed a solution based on Moving Least-Squares (MLS) [116] that was designed to deal with noisy data with good rates of approximation.

In this chapter, we propose a method of approximation of functions that leverages the advantages of the Manifold Locally Optimal Projection (MLOP) algorithm [57] to complement the strengths of the method of Radial Basis Functions (RBF) [21, 47]. We introduce this duet for approximation in high dimensions under noisy conditions (both in the domain and in the codomain of the function). In what follows we will provide a short introduction to the RBF method as well as to the Locally Weighted Average Approximation method (which will be used to contrast the RBF numerical results). In the next section, we will explain the proposed methodology, and demonstrate the improvement in approximation via several numerical examples.

Radial Basis Functions constitute a very useful and convenient multivariate interpolation tool [21,47]. Given the values of a function $f : \mathbb{R}^n \to \mathbb{R}^s$ at center points $x_i \in \mathbb{R}^n$, $i = 1, ..., K$, we approximate the value of f at a new point x by the formula

$$\tilde{f}(x) = \sum_{i=1}^{K} \lambda_i \phi(\|x - x_i\|)\,, \tag{5.1}$$

where $\phi : \mathbb{R}_+ \to \mathbb{R}$ is a radial basis function and λ_i are scalar parameters chosen to maintain interpolation at the center points, i.e., $\tilde{f}(x_i) = f(x_i)$. For examples of possible choices of radial basis functions, see [22,133]. In the numerical examples presented in subsection 5.4 we choose to use the following Gaussian RBF with local support:

$$\begin{aligned}
\phi_1(r) &= \exp\{-(r/h)^2\},\\
\phi_2(r) &= \exp\{-(r/h)^2\}\big(1 + r/h\big),\\
\phi_3(r) &= \exp\{-(r/h)^2\}\big(15 + 15(r/h) + 6(r/h)^2 + (r/h)^3\big).
\end{aligned}$$

In what follows, we will state a theorem proved in [137] on the order of approximation of a general form of RBF in a Sobolev space of a method that uses a RBF of general form, namely

$$\tilde{f}(x) = \sum_{j=1}^{K} \lambda_j \phi_w(\|x - x_j\|) + \sum_{i=1}^{I} \alpha_i p_i(x)\,, \tag{5.2}$$

where $\phi_w := \phi(\cdot/w)$, with w depending on the fill-distance h of the points $X = \{x_j\}_{j=1}^{K}$, $p_1, ..., p_I$ is a polynomial basis for Π_m, and the coefficients λ_j and α_j are chosen to satisfy the linear system $\hat{f}(x_j) = f(x_j)$ for $j = 1, ..., K$ and $\sum_{j=1}^{K} \lambda_j p_i(x_j) = 0$, $i = 1, ..., I$.

To state the theorem on the order of approximation of the RBF method we nee some additional definitions.

Definition 5.1. *For any $k \in \mathbb{N}$, p, the Sobolev space $W_p^k(\Omega)$ is defined as*

$$W_p^k(\Omega) := \left\{ f : \|f\|_{k,L_p(\Omega)} := \left(\sum_{|\alpha|_1 \le k} \|D^\alpha f\|^p_{L_p(\Omega)} \right)^{1/p} < \infty \right\}, \tag{12}$$

for $p < \infty$, and as

$$W_\infty^k(\Omega) := \left\{ f : \|f\|_{k,L_\infty(\Omega)} := \sum_{|\alpha|_1 \le k} \|D^\alpha f\|_{L_\infty(\Omega)} < \infty \right\}, \tag{12'}$$

for $p = \infty$.

Definition 5.2. *Let $X = \{x_j\}_{j=1}^K$ be a set of points with fill distance h and separation distance $\delta = \min_{1 \le i \ne j \le N} \|x_i - x_j\|/2$. Then we say that X is quasi-uniformly distributed if there exists a constant $\eta > 0$ independent of X such that*

$$2\delta \le h \le \eta\delta\,. \tag{5.3}$$

Definition 5.3. *Let ϕ_w be a radial basis function, and let $\widehat{\phi}_w$ be its Fourier transforms. Define the supremum of the norm of $\widehat{\phi}_w$ as*

$$M_{\phi,w}(r) := \sup_{\theta \in B(0,r)} \|\widehat{\phi}_w(\theta)\|^{-1/2}\,. \tag{5.4}$$

Definition 5.4. *Let f be as defined in* (5.2)*, and ϕ_w be a radial basis function. Then the norm of the corresponding error functional is defined as*

$$P_{\phi,X}(x) = \sup_{\|f\|_\phi \ne 0} \frac{\|f(x) - \widetilde{f}(x)\|}{\|f\|_\phi}, \quad \text{where } \|f\|_\phi = \int_{\mathbb{R}^d} \frac{\|\widehat{f}(\theta)\|^2}{\widehat{\phi}_w(\theta)} d\theta\,. \tag{5.5}$$

Finally, we are ready to state the promised theorem, proven in [137]

Theorem 5.1. *Let $X = \{x_j\}_{j=1}^K$ be a set of quasi-uniformly distributed scattered points (see Definition* 5.2*), and let $\widetilde{f}(x)$, defined as in* (5.1)*, be an interpolant to f on X using the radial basis function $\phi_w = \phi(\cdot/w)$. Let $M_{\phi,w}(r)$ with $r > 0$, be defined as in* (5.4)*. Assume that there exists a constant $\delta_0 > 0$ such that*

$$P_{\phi,X/w}(x/w) M_{\phi,w}(\delta_0/h) \le o(h^k)\,. \tag{5.6}$$

Then, for every function $f \in W_\infty^k(\Omega)$, with $k \in \mathbb{N}$, the error of the RBF *method is estimated*

as

$$\|f - \widetilde{f}\|_{L_\infty(\Omega)} = o(h^k)\,. \tag{5.7}$$

Remark 5.1. *As a result, an important key advantage of the* RBF *method is that it performs better on quasi-uniform samples. This property will be utilized in the next section.*

We will also give a short introduction to the Locally Weighted Average Approximation, which will be used as a reference in the section devoted to our numerical examples. Given the values of the function $f : \mathbb{R}^n \to \mathbb{R}^s$ at the points $x_i \in \mathbb{R}^n$, the locally weighted average approximation of f at a point x is defined as

$$f(x) = \frac{\sum_i w_i f(x_i)}{\sum_i w_i}\,,$$

where $w_i = e^{-\frac{\|x_i - x\|^2}{h^2}}$, and h is the fill-distance of the points $\{x_i\}_{i=1}^I$. Concerning the accuracy of the method, we note that the locally weighted average approximation reconstructs constant functions.

5.2 Approximation of Functions on a Manifold

Let $\mathcal{M}$ be a smooth d-dimensional manifold in $\mathbb{R}^n$, where $d \ll n$. Let $P = \{p_j\}_{j=1}^J$ be a set of points which were sampled from $\mathcal{M}$ and are affected by noise, and let $f : \mathcal{M} \to \mathbb{R}^s$ be a smooth function. Given noisy measurements of f at the points in P, the approximation problem has two steps:

1. Find a noise-free representation of the manifold $\mathcal{M}$ and the noise-free values of f.
2. Estimate the value of the function f at a new given point x.

Our solution for the first step is based on generalizing the MLOP method designed for manifold denoising to the case of function denoising. The key idea of the solution is to embed the approximation problem in a higher-dimensional space, and denoise the data there. Given the input data, which consists of the set of points $\{p_j\}_{j=1}^J \subset \mathbb{R}^n$ and the set of values $\{f(p_j)\}_{j=1}^J \subset \mathbb{R}^s$, we define a new point-set $\widehat{P} = \{\widehat{p}_j\}_{j=1}^J$ to be the graph of a function f, i.e. as the set of

ordered pairs, where $\widehat{p_j} = (p_j, f(p_j))$ is the pairing of p_j with the value $f(p_j)$. The points in $\widehat{P}$ are now considered as data points in $\mathbb{R}^{n+s}$, taken from the $\widehat{\mathcal{M}} = \text{Graph}\, f = \{(x, f(x)) : x \in \mathcal{M}\}$. The newly defined set $\widehat{\mathcal{M}}$, being the graph of a smooth function f defined on a smooth d-dimensional manifold, is itself a smooth d-dimensional manifold. It should be noted that prior to the embedding in the $(n+s)$-dimensional space, the values of f should be normalized to the maximum value of the p_j coordinates, in order to avoid them dominating the p_j entries during the norm calculations of the MLOP algorithm.

In this setting, we are now denoising a d-dimensional manifold embedded in $n+s$ dimensions. We apply the MLOP method on the new data set $\widehat{P}$, and look for a clean dataset $\widehat{Q} = \{\widehat{q_i}\}_{i=1}^{I} \subset \mathbb{R}^{n+s}$ which will serve as a noise-free approximation of $\widehat{\mathcal{M}}$. The main advantage of this approach is that with a single MLOP execution on the $\{\widehat{p_j} = (p_j, f(p_j))\}$ data in $\mathbb{R}^{n+s}$ we produce a noise-free set $\widehat{Q} \subset \mathbb{R}^{n+s}$, which in fact consist of a noise-free set Q, that reconstructs the manifold $\mathcal{M}$, and an estimate of the clean value of the function evaluated at these points, $\widetilde{f}(Q)$.

In the second step, we address the problem of evaluating the function at a new point $z \in \mathcal{M}$. The outcome of the first step is a set of points which is not only noise-free, but also quasi-uniformly distributed on the manifold $\widehat{\mathcal{M}}$. This key idea paves the way towards estimating the value of the function at a new given point on $\mathcal{M}$, or near $\mathcal{M}$, with a good order of approximation. Our solution is based on utilizing the RBF approximation, as defined in (5.1), while setting the centers at the cleaned points Q and appending the corresponding cleaned function values $\widetilde{f}(Q)$. These steps are summarized in Algorithm 4:

Algorithm 4 Function Approximation on a Manifold in High Dimensions

1: **Input:** $P = \{p_j\}_{j=1}^{J} \subset \mathbb{R}^n$, $\{f(p_j)\}$, $\{z_k\}_{k=1}^{K}$ ▷ where $\{z_k\}_{k=1}^{K}$ is a set of new points for function approximation

2: **Output:** $Q = \{q_i\}_{i=1}^{I} \subset \mathbb{R}^n$, $\widetilde{f}(Q)$, $\{\widetilde{f}(z_k)\}_{k=1}^{K}$

3: Denoise the input data by running MLOP with $\widehat{P} = (P, f(P)) \subset \mathbb{R}^{n+s} \to \widehat{Q} = (Q, \widetilde{f}(Q))$

4: **for** each $z_k \in \{z_k\}_{k=1}^{K}$ **do**

5: Approximate $\widetilde{f}(z_k)$ via RBF, with centers set at Q and the corresponding $\widetilde{f}(Q)$ values

6: **end for**

5.3 Theoretical Analysis of the Method

In this section we discuss some of the theoretical aspects of the proposed approach to approximation of functions. Our analysis relies on the theory of the MLOP as well as the RBF method. Specifically, we can use the MLOP results proved in Chapter 3 regarding the convergence of the MLOP algorithm to a stationary point, its rate of convergence, and its the complexity estimated for the problem at hand. We also build on the results about the order of approximation of the MLOP. Thus, we can state the following theorem on the order of approximation for the our approximation problem at hand.

5.3.1 Order of Approximation

Theorem 5.2 (Order of approximation)**.** *Let $P = \{p_j\}_{j=1}^{J}$ be a set of points sampled from a d–dimensional C^2 manifold $\mathcal{M}$ without noise that satisfy the h-ρ condition. Let $f : \mathcal{M} \to \mathbb{R}^s$ be a smooth multivariate function given at points of P. Suppose that $f \in W_\infty^k(\Omega)$ and fulfills all the conditions of Theorem* 5.1*. Then:*

(i) *For fixed ρ and δ, there is a set of points $Q = \{q_i\}_{i=1}^{I} \subset \mathbb{R}^n$ which approximates $\mathcal{M}$ with order $O(\hat{h}^2)$, where $\widehat{h} = \max\{\widehat{h}_1, \widehat{h}_2\}$ and $\widehat{h}_1$ and $\widehat{h}_2$ are defined in Subsection* 3.3.2 *with respect to the high-dimensional data $\widehat{P} = \{(p_j, f(p_j))\} \subset \mathbb{R}^{n+s}$. Moreover, the order of approximation of f on the set Q is also $O(\hat{h}^2)$.*

(ii) *The order of approximation of f at a new point is less than $C_1\hat{h}^2 + C_2 h_2^k$ when using the* RBF *approximation with centers at Q, and $\tilde{f}(Q)$, where h_2 is the optimal support of $\hat{w}_{i,i'}$ as defined in Subsection* 3.3.2*, and C_1 and C_2 are constants.*

Proof. Given the input data, which consists of the points $\{p_j\}_{j=1}^{J}$ and the function values $\{f(p_j)\}_{j=1}^{J}$, we define a new point-set, which is the graph of the function f, $\widehat{P} = \{\widehat{p}_j\}_{j=1}^{J}$ sampled from a new manifold $\widehat{\mathcal{M}} \in \mathbb{R}^{n+s}$, where $\widehat{p}_j$ are defined as the pairing of the p_j-data with the corresponding values $f(p_j)$. In this setting, we use the MLOP algorithm to denoise a d-dimensional manifold embedded in $(n+s)$-dimension. Thus, by applying the MLOP method to the new data we obtain the set $\widehat{Q}$ of points that reconstruct $\widehat{\mathcal{M}}$. From Theorem 3.5 it follows that $\widehat{Q}$ approximates $\widehat{\mathcal{M}}$ with the order $O(\widehat{h}^2)$, where $\widehat{h}$ is the representative distance

introduced in Definition 3.3 for the extended data set $\widehat{P} = \{(p_j, f(p_j))\} \in \mathbb{R}^{n+s}$. Recall that $\widehat{Q}$ is in fact a combination of a noise-free set Q, which reconstructs the manifold $\mathcal{M}$, and an estimate of the clean values of the function at these points in Q, $\widetilde{f}(Q)$. Therefore, the order of approximation of Q to $\mathcal{M}$, and of the estimated values $\widehat{f}$ to f is $O(\widehat{h}^2)$.

Subsequently, we use an RBF approximation with theoretical order of approximation $O(h^k)$ (as stated in Theorem 5.1). In our case the relevant h is h_2, the support size of $\widehat{w}_{i,i'}$ as in Definition 3.3. Therefore, the overall order of approximation for a new point is a combination of the two orders, namely $\leq C_1\widehat{h}^2 + C_2 h_2^k$, with C_1, and C_2 constants. □

5.3.2 Complexity of the Approximation of Functions

Theorem 5.3. *Let $P = \{p_j\}_{j=1}^J$ be a set of points sampled near a d-dimensional manifold $\mathcal{M} \subset \mathbb{R}^n$ and let $Q = \{q_i\}_{i=1}^I$ be a set of points which will provide the desired noise-free manifold reconstruction. Let $f : \mathcal{M} \to \mathbb{R}^s$ be a multivariate function given at the points of P. Then the complexity of the approximation of f via the* MLOP *algorithm for the denoising step is $O((n+s)mJ + kI((n+s)m\widehat{I} + \widehat{J}) + nmI + I\log_2^2 s)$, and for evaluating f at a new point is $O(nm + I)$, where the number of iterations k is bounded as in Theorem* 3.7*, m, with $m \ll n$, is the smaller dimension to which we reduce the dimension of the data, and $\widehat{I}$ and $\widehat{J}$ are the numbers of Q-points and P-points, respectively, in the support of the weight function $\hat{w}_{i,i'}$ and $w_{i,j}$.*

Proof. The estimate of the complexity of the algorithm can be separated into two steps: **pre-processing** and **evaluating the function at a new point**. The pre-processing step consists of applying the MLOP algorithm in $\mathbb{R}^{n+s}$, as well as finding the RBF coefficients λ_i by solving the Least-Squares problem. By Theorem 3.2, the complexity of applying the MLOP in the higher dimension $n+s$ is $O((n+s)mJ + kI((n+s)m\widehat{I} + \widehat{J}))$. The output of this stage is a points set Q of size I, and the corresponding set of values $\widetilde{f}(Q)$. From this point on, all the approximation operations are performed on Q, and if $I \ll J$ then we can increase sufficiently the efficiency. The next part of the pre-processing step is to evaluate the radial basis function ϕ for each $q_i \in Q$, which costs $O(nmI)$, and then to find the λ_i by solving the Least-Squares problem, which takes $O(I\log_2^2 s)$ (as shown in [87]). As a result, the complexity of the pre-

processing step is $O((n+s)mJ + kI((n+s)m\widehat{I} + \widehat{J}) + nmI + I\log_2^2 s)$.

Finally, using the λ_i already found, we evaluate the function at a new point in time $O(nm+I)$. It should be stressed that although the pre-processing steps are cost-effective, they are executed once before the function is approximated at a new points set. Thus, if the number of new points for which the approximation needs to be found is large, then the pre-processing steps have less effect on the runtime. □

5.4 Numerical Examples

In what follows we present several numerical experiments to demonstrate the advantages of our methodology. We can point out two strengths of the proposed approximation approach. On the one hand, denoising the data domain as well as the function codomain plays an important role in the approximation of functions. On the other hand, sampling the manifold quasi-uniformly improves significantly the approximation of functions by means of classical approximation methods on new data.

Given data sampled from a manifold with noise, and the noisy values of a function f at these points, we follow the function approximation procedure described above. Specifically, we define a new problem in $\mathbb{R}^{n+s}$, and apply k MLOP iterations to clean the newly defined manifold, which results in a new point-set $Q^{(k)}$, and the corresponding cleaned value set $\widetilde{f}(Q^{(k)})$. Next, we randomly select 100 points, $\{z_i\}_{i=1}^{100}$, from a clean reference dataset, and estimate the values of the function f at these points using both the RBF approximation, where the centers of the RBF function are taken at the points of $Q^{(k)}$ (with the radial basis function set to either ϕ_1, ϕ_2, or ϕ_3, as defined below equation (5.1)), and the locally weighted average approximation (defined by formula 5.8). As a result, the approximation at the new points relies on the clean quasi-uniformly distributed $Q^{(k)}$ points, as well as on the clean values $\widetilde{f}(Q^{(k)})$. In all the stages above we evaluate the accuracy of the approximation as the relative maximum error of the L_1 norm of the difference between the value of $\widetilde{f}$ at the new point and the value of f at the closest point in the reference dataset, as well as the root-mean-square error and the standard deviation.

5.4.1 Approximating smooth and non-smooth functions on one dimensional manifold in high dimension

We start with two examples of functions, one smooth and the other non-smooth, both on a one-dimensional manifold embedded in a high-dimensional space. Although in principle the approximation requires a smooth function, it can still be applied to a non-smooth function, provided that we end up with a smoothed result. Specifically, we consider the case of the manifold $O(2)$ of orthogonal matrices, embedded in a 60-dimensional linear space by using the parameterization

$$p = [\cos(\theta), -\sin(\theta), \sin(\theta), \cos(\theta), 0, \ldots, 0]\,, \tag{5.8}$$

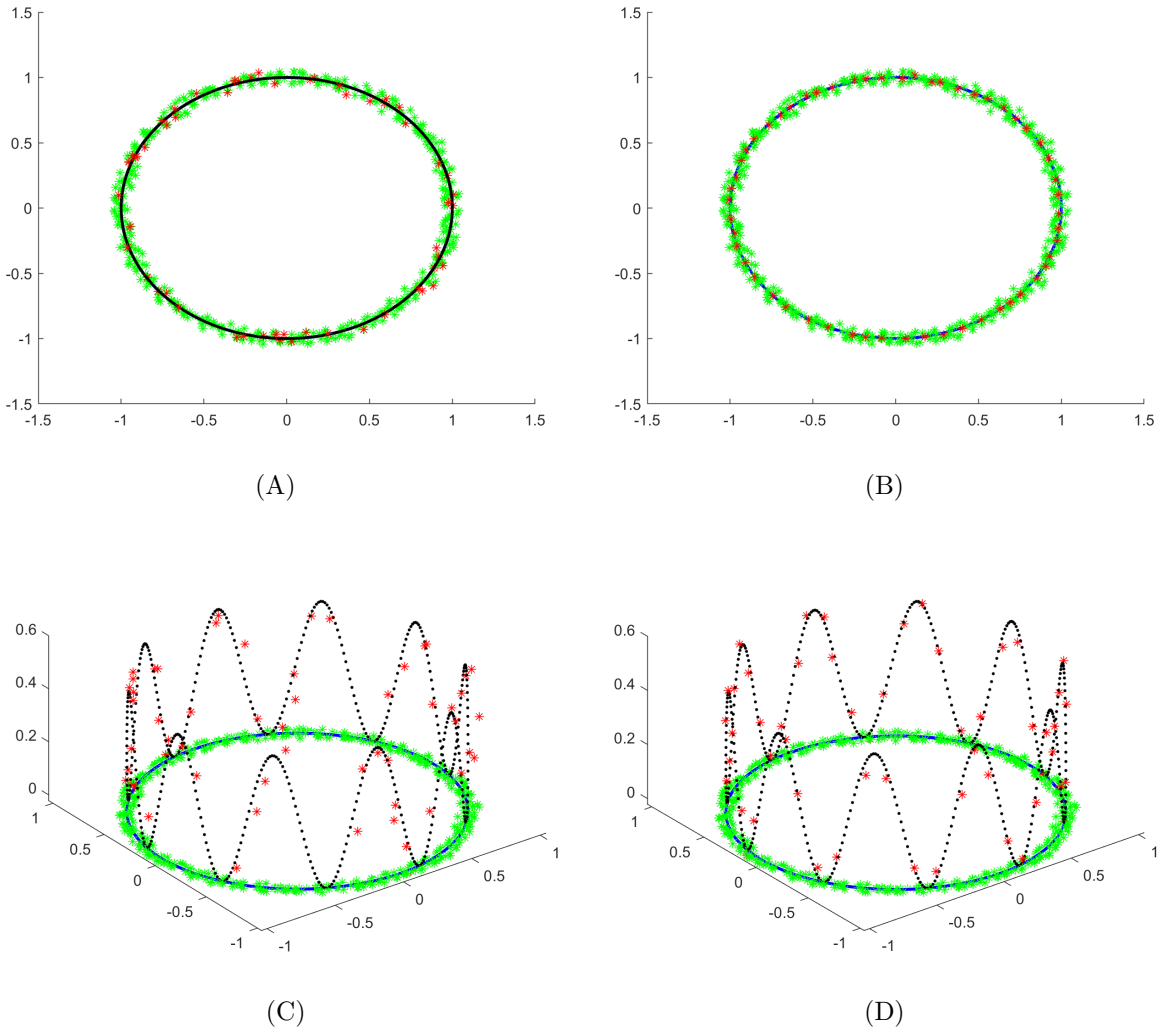
where $\theta \in [-\pi, \pi]$. The input dataset $\widehat{P}$ was constructed by sampling 500 equally distributed points in the parameter space. Next, we randomly sampled an orthogonal matrix $A \in \mathbb{R}^{60\times 60}$, and created a new point-set via the non-trivial vector embedding

$$P = A\widehat{P}\,. \tag{5.9}$$

Subsequently, we added a uniform noise $U(-0.1, 0.1)$, and initialized the set Q by selecting 55 points from P. Figure 5.1 (A) illustrates the first two coordinates of the points in our set (after multiplication by the matrix A^{-1}). The noisy sample points are shown in green, while the initial reconstruction points are shown in red.

We start by approximating the **smooth function** $f(x) = \frac{1}{4}(1 + \sin(10\,\theta))$, where θ corresponds to the value used in the expression (5.8) of p. Next, a uniform noise $U(-0.1, 0.1)$ was added in the codomain (see Figure 5.1 (C)), and then we applied the MLOP algorithm, which reconstructed the manifold (Figure 5.1 (B)), as well as denoised function values (Figure 5.1 (D)). Table 5.1 summarizes the errors that correspond to different scenarios. We first notice that due to the denoising effect on Q points the maximum relative error decreased from 0.31 to 0.13 for noisy data $Q^{(0)}$ as opposed to the clean data $Q^{(150)}$. Next, we can also see the benefits of using the MLOP algorithm prior to approximating the function with the RBF method on the new data. In the present example the maximum relative error of the best RBF execution decreased dramatically from 0.66 when running the RBF with centers at $Q^{(0)}$, to an 0.12 when

running on the quasi-uniform data, and with very low error variance. A quick comparison between the RBF method and the approximation via locally weighted average shows that the latter loses the battle to RBF (even though it produce better results on the clean data versus the noisy one).



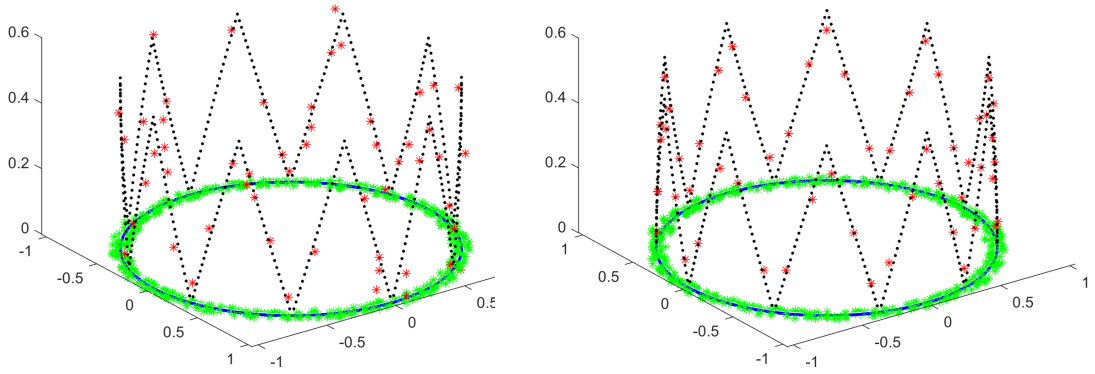
(A) (B)

(C) (D)

Figure 5.1: Manifold of orthogonal matrices embedded in a 60-dimensional space. Shown are the first two coordinates of the point-set (after multiplication by A^{-1}). (A) Scattered data with uniformly distributed noise $U(-0.1; 0.1)$ (green), and the initial point-set $Q^{(0)}$ (red). (B) The resulting point-set of the MLOP algorithm after 150 iterations, $Q^{(150)}$ (red), overlaying the noisy samples (green). (C) The initial function values evaluated at the original point-set $Q^{(0)}$ with noise $U(-0.1, 0.1)$. The black line shows the noise-free reference data. (D) Smooth function approximation via the MLOP algorithm at the data points $Q^{(150)}$.

We then applied the approximation procedure to the **non-smooth** function given by $f(x) = \frac{1}{6}(1+\arccos(\cos(10\,\theta)))$. We evaluated the function at the P-points and added the uniform noise $U(-0.1, 0.1)$ (see Figure 5.2 (A)). Then, we applied the MLOP algorithm, which resulted in a reconstructed manifold, as well as denoised function values; see Figure 5.2 (B). As this figure

shows, the non-smooth function f is approximated reliably. This is also reflected in the errors listed in Table 5.1, which shows that the approximation error decreased from 0.2 to 0.1 after the denoising procedure. The advantages of the MLOP approach are also demonstrated by the error decrease, for the best choice of radial basis function, from 0.77 to 0.15 for approximation on a new point-set. This shows the robustness of the approximation process with respect to the clean data. Here again, we see that the RBF method produces a better approximation then the weighted average.



(A) (B)

Figure 5.2: Manifold of orthogonal matrices embedded in a 60-dimensional space. Shown are the first two coordinates of the point-set (after multiplication by A^{-1}). Scattered data with uniformly distributed noise $U(-0.1; 0.1)$ (green), and the Q point-set (red). Left: The initial function values evaluated at the original $Q^{(0)}$-points with noise $U(-0.1, 0.1)$. The black line shows the noise-free reference data. Right: Approximation of our non-smooth function via MLOP at the data points $Q^{(150)}$.

Table 5.1: Summary of the maximum and root mean squared errors with standard deviations errors of approximation of functions on the $O(2)$ manifold embedded into 60-dimensional space

	$f(x) = \frac{1}{4}(1 + \sin 10x)$		$f(x) = \frac{1}{6}(1 + \arccos(\cos 10x))$	
	Max relative error	RMSE $\pm$ var	Max relative error	RMSE $\pm$ var
Error over $Q^{(k)}$				
$f(Q^{(0)})$	0.31	0.06 ± 0.0011	0.2	0.04 ± 0.0007
$f(Q^{(150)})$	0.13	0.03 ± 0.0003	0.1	0.03 ± 0.0002
Error over 100 new points				
RBF, ϕ_1, centers at $Q^{(0)}$, noisy f	0.66	0.14 ± 0.0079	0.77	0.16 ± 0.010
RBF, ϕ_1, centers at $Q^{(150)}$, cleaned $\widetilde{f}$	0.12	0.03 ± 0.0002	0.15	0.03 ± 0.0004
RBF, ϕ_2, centers at $Q^{(150)}$, cleaned $\widetilde{f}$	0.12	0.03 ± 0.0002	0.16	0.03 ± 0.0003
RBF, ϕ_3, centers at $Q^{(150)}$, cleaned $\widetilde{f}$	0.24	0.05 ± 0.0009	0.21	0.05 ± 0.0007
Weighted average	0.28	0.10 ± 0.0018	0.32	0.09 ± 0.0024

In what follows we demonstrate our function approximation methodology on several examples of a low-dimensional manifold embedded in high-dimensional space. Specifically, we embedded a two-dimensional cylindrical structure and then a six-dimensional cylindrical structure in $\mathbb{R}^{60}$. We start with the two-dimensional cylindrical structure. We sampled the structure using the parameterization

$$p = tv_1 + \frac{R}{\sqrt{2}}\big(\cos(u)v_2 + \sin(u)v_3\big)\,,$$

where $v_1 = [1, 1, 1, 1, 1, \ldots, 1]$, $v_2 = [0, 1, -1, 0, 0, \ldots, 0], v_3 = [1, 0, 0, -1, 0, \ldots, 0]$ ($v_1, v_2, v_3 \in \mathbb{R}^{60}$), $t \in [0, 2]$ and $u \in [0.1\pi, 1.5\pi]$. Using this representation, 800 equally distributed (in parameter space) points were sampled with uniformly distributed noise (i.e., $U(-0.1, 0.1)$). We evaluated the function $f(t, u) = 1.3(1 + \sin(0.5u + 1.5t))$ at these points, and constructed the initial Q-set by randomly sampling 150 points (see Figure 5.3 (A) for the P and Q data, and Figure 5.3 (C) for the values of the function at the Q-points). The representative distances of the P-set and Q-set were $h_1 = 0.19$ and $h_2 = 0.27$, resepctively. Next, we applied the MLOP

algorithm on the new data of $(P, f(P))$, and extracted the sets Q and $\widetilde{f}(Q)$ (see Figure 5.3 (B) for the cleaned Q-points, and Figure 5.3 (D) for the cleaned values at the Q-points). In addition, we approximated the values of f at 100 new points, randomly selected from the reference data. The evaluation error results are summarized in Table 5.2. The maximum relative L_1 error and the RMSE accompanied with the variance are summarized in Table 5.2. It should be noted that since we compare the maximum error to clean data, the relative error can exceed 1. One can see that the new data RBF with ϕ_2 as well as ϕ_3 achieve the lowest errors.

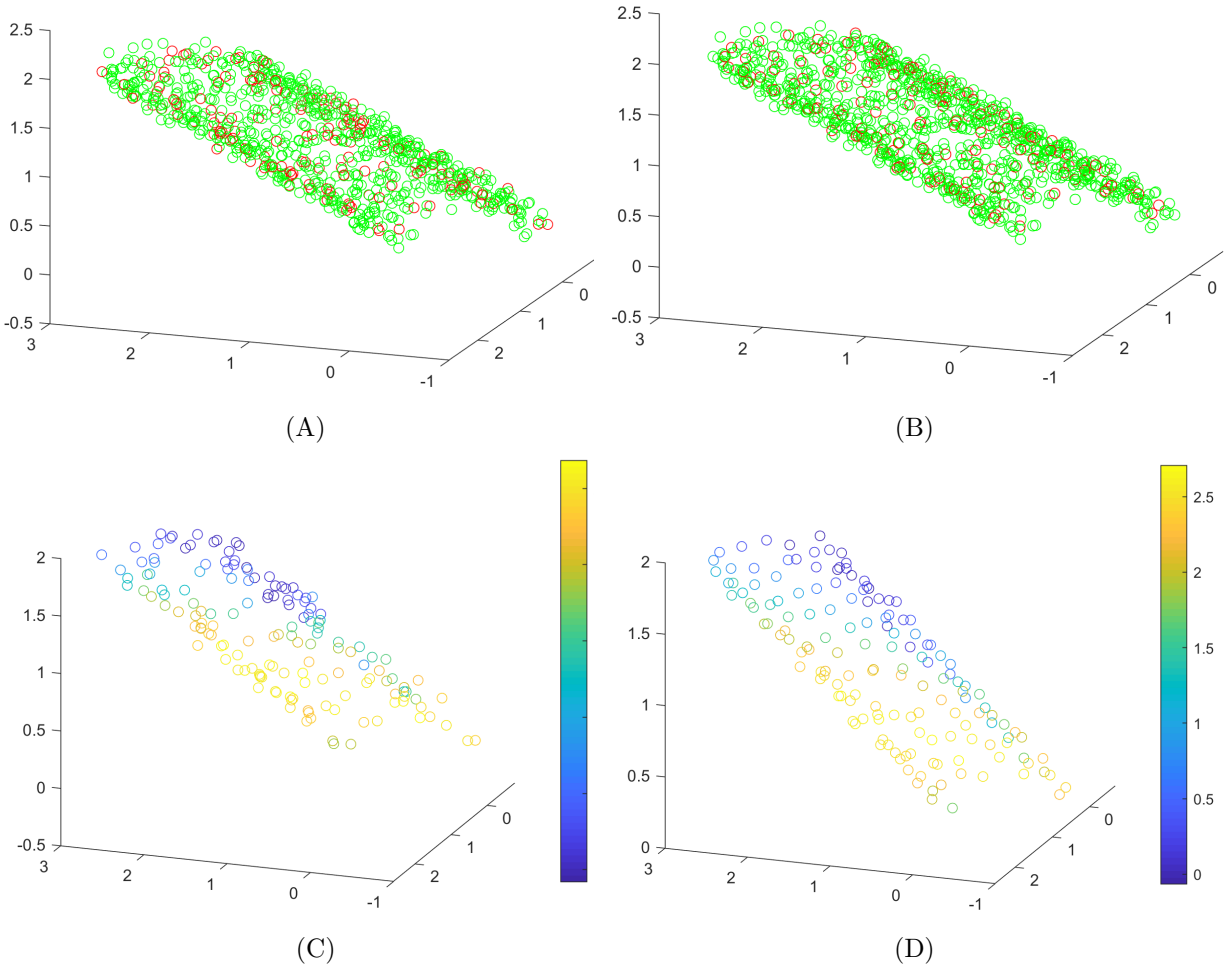


Figure 5.3: Two-dimensional cylindrical structure embedded in a 60-dimensional space. The first three coordinates of the point-set are shown. (A) Scattered data with uniformly distributed noise $U(-0.1; 0.1)$ (green), and the initial point-set $Q^{(0)}$ (red). (B) The resulting point-set of the MLOP algorithm after 200 iterations, $Q^{(300)}$ (red), overlaying the noisy samples (green). (C) The initial values of the function at the original $Q^{(0)}$-points with noise $U(-0.1, 0.1)$. (D) MLOP approximation at the data points $Q^{(300)}$.

5.4.2 Six-dimensional cylindrical structure

Next, we tested our method on higher-dimensional manifolds by utilizing an n-sphere to generate an $(n+1)$-dimensional cylinder (in the example of the two-dimensional cylinder, we used a circle to generate the structure). Here, we utilized a five-dimensional sphere to build a six-dimensional manifold, using the parameterization

$$x_1 = R\cos(u_1)\,, \quad x_2 = R\sin(u_1)\cos(u_2), \quad \ldots, \quad x_6 = R\sin(u_1)\sin(u_2)\cdots\sin(u_5)\sin(u_6)\,.$$

We then embedded the sampled data in a 60-dimensional space by the parametrization

$$p = tv_0 + R^2[x_1, x_2, x_3, x_4, x_5, x_6, 0, \ldots, 0]\,, \tag{5.10}$$

where $R = 1.5$, $t \in [0, 2]$, $u_i \in [0.1\pi, 0.6\pi]$, and $v_0 \in \mathbb{R}^{60}$ is a vector with 1's in positions $1, ..., d+1$ and 0 in the remaining positions. We randomly sampled the P-points from the six-dimensional cylindrical structure and embedded the sampled data in a 60-dimensional space. This process resulted in 1200 points in the P-set. We evaluated the function $f(u_1, \ldots, u_6) = \sum_{i=1}^{6} u_i$ at these points, and constructed the initial Q-set by randomly sampling 460 points. Next, uniformly distributed noise $U(-0.2; 0.2)$ was added to the points. To avoid trying to visualize a six-dimensional manifold, we plot here the cross-section of the cylindrical structure in three dimensions. In Figure 5.4 (A) we present the P- and Q-data, and in Figure 5.4 (C) the values of the function at the Q-points. The representative distances of the P-set and Q-set were $h_1 = 0.24$ and $h_2 = 0.37$, respectively. We then applied the MLOP algorithm on the new data of $(P, f(P))$, and extracted the sets Q and $\widetilde{f}(Q)$ (see Figure 5.4 (B) for the cleaned Q-points, and Figure 5.4 (D) for the cleaned function values at the Q points). In addition, we approximated the values of the function at 100 new points, randomly selected from the reference data. The maximum relative L_1 error and the RMSE accompanied with the variance are summarized in Table 5.2. One can see that the new data RBF with ϕ_2 and the weighted average achieve lower errors.

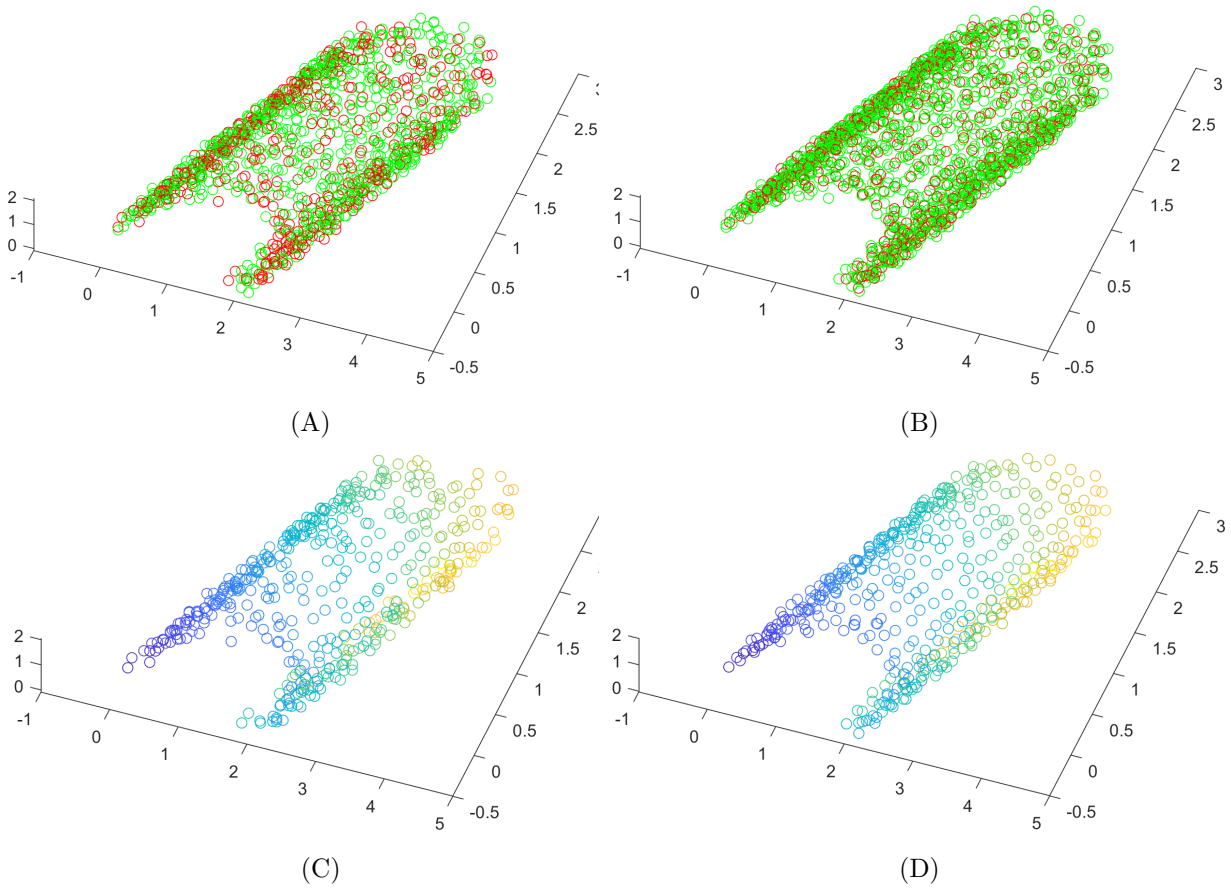


Figure 5.4: Six-dimensional cylindrical structure embedded in a 60-dimensional space. Plot of the cross-section of the cylindrical structure in three dimensions. (A) Scattered data with uniformly distributed noise $U(-0.2; 0.2)$ (green), and the initial point-set $Q^{(0)}$ (red). (B) The resulting point-set of the MLOP algorithm after 300 iterations, $Q^{(300)}$ (red) overlaying the noisy samples (green). (C) The initial function values evaluated at the original $Q^{(0)}$ points with noise $U(-0.2, 0.2)$. (D) MLOP approximation at the data points $Q^{(300)}$.

Table 5.2: Summary of the maximum and root mean squared error with standard deviations errors of approximation of a function on a two-dimensional and six-dimensional cylindrical manifold embedded in a 60-dimensional space

	2D cylinder in $\mathbb{R}^{60}$		6D cylinder in $\mathbb{R}^{60}$	
	Max relative error	RMSE $\pm$ var	Max relative error	RMSE $\pm$ var
Error over $Q^{(k)}$				
$f(Q^{(0)})$	0.11	0.09 ± 0.0029	0.066	0.08 ± 0.0018
$f(Q^{(300)})$	0.06	0.05 ± 0.0012	0.054	0.06 ± 0.0012
Error over 100 new points				
RBF with ϕ_1, centers at $Q^{(0)}$, noisy f	1.37	0.3 ± 0.04	0.42	0.4 ± 0.063
RBF with ϕ_1, centers at $Q^{(300)}$, cleaned $\widetilde{f}$	0.38	0.13 ± 0.007	0.26	0.26 ± 0.026
RBF with ϕ_2, centers at $Q^{(300)}$, cleaned $\widetilde{f}$	0.11	0.05 ± 0.0009	0.08	0.07 ± 0.0018
RBF with ϕ_3, centers at $Q^{(300)}$, cleaned $\widetilde{f}$	0.10	0.04 ± 0.0006	0.13	0.04 ± 0.0007
Weighted average	0.2	0.1 ± 0.0024	0.09	0.06 ± 0.0027

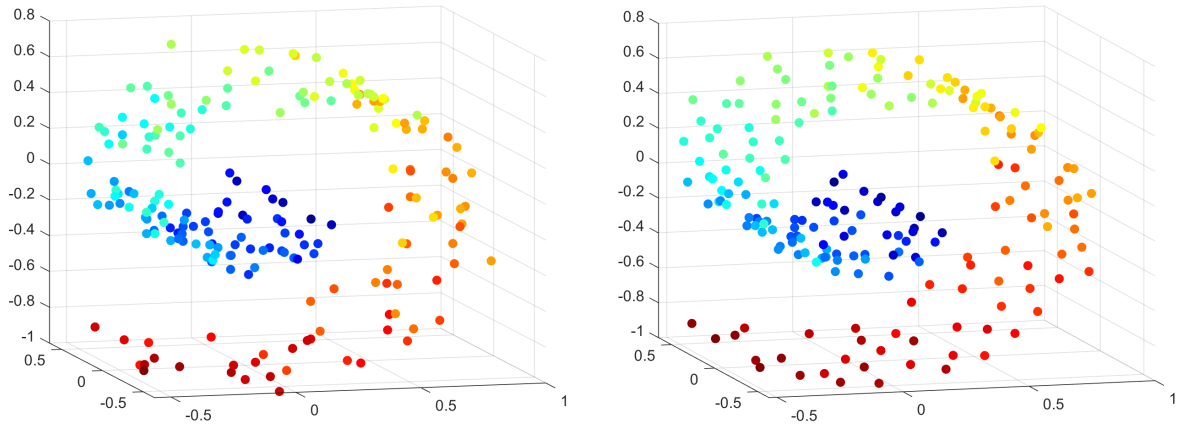
5.4.3 Robustness to Noise

In the following example, we examine the effect of the noise level in the target domain on the quality of the approximation. To do this numerically, we sampled a function over a Swiss Roll using the parameterization

$$p = \frac{1}{10}[x, y, z, 0, \ldots, 0]\,,$$

where $x = t\sin(t)$, y is a random number in the range $[-6, 6]$, and $z = t\cos(t)$, with $t = 8k/n+2$ and $k \in \mathbb{N}$. The approximated function was $f(p) = t$. We created a Swiss Roll with 800 data points, and randomly sampled 200 points as the initial Q-set. We added noise with various magnitudes (0.1, 0.2, 0.5, and 0.7) to the P-points as well as to the values of f at the P-points. For example, Figure 5.5 left shows a case of approximation with uniformly distributed noise $U(-0.2, 0.2)$, while the right plot presents the denoised version. We see that the data were cleaned both in the domain and in the codomain of the function. In Figure 5.6, we plot

the error values under various noise scenarios in the codomain, both for the noisy $Q^{(0)}$ data and the error of the RBF approximation on the $Q^{(300)}$ data. One can see that although the approximation error increases on the noisy data (from 0.03 to 0.22), the approximation error on clean and quasi-uniformly distributed data error grows only moderately (from 0.02 to 0.18). We also see that at high levels of noise (e.g., 0.7) the accuracy is good. This shows the strengths of our approach, and justifies the need for data denoising as well as uniform sampling before approximation algorithms are applied.



(A) (B)

Figure 5.5: Swiss Roll embedded in $\mathbb{R}^{60}$. The figure depicts the first tree coordinates. Left: The initial values of the function at the original $Q^{(0)}$ points with noise $U(-0.2, 0.2)$, with values indicated by the color. Right: MLOP function approximation at the data points $Q^{(300)}$ cleaned via the MLOP.

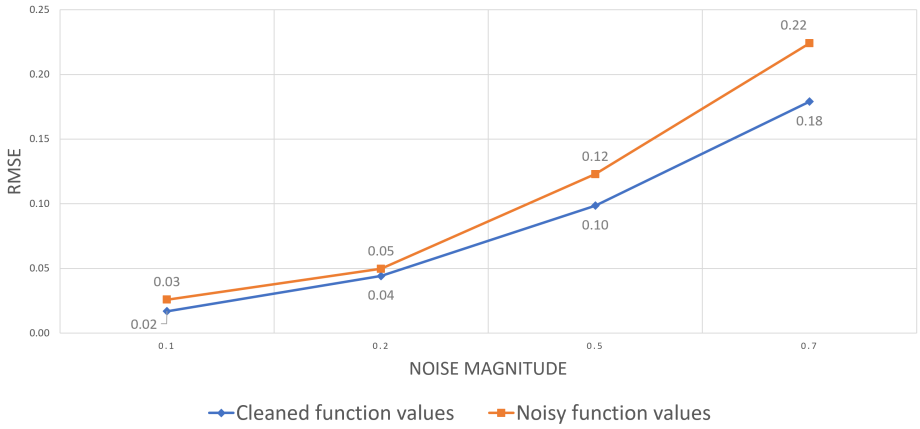


Figure 5.6: Effect of noise level on the accuracy of function approximation for a Swiss Roll embedded in a 60-dimensional space. The RMSE error evaluated on the original noisy data is shown in orange, while the RMSE error on the cleaned data is presented in blue.

Chapter 6

Manifold Compressed Sensing

6.1 Introduction

In the current era, we are flooded with data, since its acquisition became affordable and easy in a multitude of fields (starting from color, hyperspectral, and medical imaging, continuing with various biological applications, and lately, with the rise of language processing, also in textual data). This data availability raises several challenges, the most notable of which relate to storing the data and processing it efficiently. While numerous methods were proposed for the two indicated issues (e.g., increasing computational units, or developing efficient (sketching) algorithms), at times, simply compressing the entire data can provide a good solution for both of the problems. Data compression aims at finding the most concise representation of a signal that will provide enough information for the target application.

The Compressed Sensing (CS) field can be viewed as a possible solution for these tasks. When usually dealing with compressed sensing, one typically addresses the problem of recovering a *single* signal. The problem can be formulated as following: Given some set of measurements of an unknown signal, the goal is to recover the signal reliably. Alas, with the influx of big data a new challenge arose - compressing and recovering a signal which is a *data-set*. Often, a reasonable assumption is that the data lay on a manifold; then this property can be utilized to create a compressed representation of the data.

There are several motivating factors to pursue manifold sensing and recovering. On the one hand, sensing data densely with a desired density is not always possible for practical reasons. Thus, un-compressing sampled data with respect to a given density is essential. On the other hand, densely sampled data are not always a blessing, due to considerations of space, presence of noise, and computational power. At times a densely sampled manifold is not necessary optimal for training and can increase the size of the model that learned it. For example, when using a Neural Network (NN) to learn a densely sampled dataset, a deeper network may be required in order to cope with nuances as well as the noise affecting the data. However, a smaller network would be sufficient in "clean" settings, i.e., if the essence of the data would be sterilized. Another important question that arises is whether the densely sampled data at hand represent the entire manifold. For instance, a common belief in machine learning is that NNs need a lot of data for training. However, this not always true, and it is better to have a limited amount of data, sampled sparsely from the entire domain, rather than a great amount of densely sampled data, which only partly represents the learned world. In this chapter we will pose and discuss the question of sensing and compressing a signal which is a manifold.

Prior to addressing manifold compression, we consider the challenging problem of compressed sensing of an individual signal. Introduced in 2006 with the works of Donoho [46] and Candes, Romberg, and Tao [25], who showed that a finite-dimensional signal a sparse or compressible representation can be recovered from a small set of linear, non-adaptive measurements. Subsequently, this direction continued with a wide range of studies, and compressed sensing gained a lot of attention in the past years (a quick search in the google shows that there are more than 100K articles today that deal with CS; see [9, 41, 49] to mention just a few).

The key idea of compressed sensing (CS) is to recover a sparse signal from very few non-adaptive linear measurements by means of convex optimization. In mathematical terms, the problem of a single signal compressed sensing is formulated as follows: we wish to recover a vector $x = \{x_i\}_{i=1}^n \in \mathbb{R}^n$ from given observations $y = Ax$, where A is the so-called sensing $m \times n$ matrix, and y is $m \times 1$ vector. An assumption is that either x itself is sparse, or that there exists an orthonormal basis or a frame ϕ such that $x = \phi c$, with c sparse. Thus, we are dealing with an underdetermined linear system of equations with sparsity as prior information

about the vector to be recovered. There are two main theoretical questions in CS. First, how should we design the sensing matrix A to ensure that it preserves the information in the signal x? Second, how can we recover the original signal x from measurements y?

A natural solution would be to use L_0, however, one usually replaces it by L_1. Thus, a commonly the solution to the CS problem is found via L_1-optimization of Basis Pursuit [33]:

$$\begin{aligned} &\min \|x\|_1 \\ &\text{subject to } Ax = y \end{aligned} \tag{6.1}$$

Due to the fact that for very large data sets L_1 minimization is often not practical, various other types of recovery algorithms were suggested. For a detailed discussion of these solutions see [49].

Along the years, various theoretical aspects of CS were studied. The theoretical foundations of this revolutionary technique were built in the pioneering works of Kotelnikov, Nyquist, Shannon, and Whittaker on the sampling continuous-time band-limited signals [81, 100, 109, 130]. Their results demonstrate that signals, images, videos, and other data can be exactly recovered from a set of uniformly spaced samples taken at the so-called Nyquist rate of twice the highest frequency present in the signal of interest. The theoretical investigation of the CS continued with the question of what properties should the sensing A matrix have in order to preserve the required information? Next, in [45] uniqueness conditions for minimization problems were studied. In [26] the question of *how many linear measurements of the signal are needed in order to ensure that the signal can be recovered with precision ϵ in the Euclidean metric* was addressed. In addition, in [27] the author provided an error bound on the recovered signal under noise-free as well as noisy conditions.

6.2 Manifold Compression

While compressed sensing of one signal gained a lot of attention, manifold compressed sensing is still open and regarded as a non-trivial problem in high-dimensional space. In fact, our preliminary investigation of the literature on manifold compressed sensing showed that this problem

received little attention. We emphasize that manifold compressed sensing is different from the problem of compressed sensing on manifolds (which was also dealt with, see e.g., [32]). In certain situations manifold compression problems can be reformulated as signal compression, thus, overcoming the new challenge [138]. In high-dimensional space, manifold compressed sensing is tightly related to manifold reconstruction. Although manifold reconstitution gained attention with the works [17, 18, 34, 57, 92, 99, 118], the problem was not formulated as a manifold compression\recovery problem. The problem of manifold compression and sensing was addressed in the works of [10, 30, 104].

In this section, we consider the fundamental problem of sensing data from a manifold and using it as a basis for recovering the manifold from a limited set of measurements (see Figure 6.1 for illustration). In this setting, we wish to recover a manifold with a given density from sparse observations, sensed from the manifold with noise. In what follows, we will describe our solution for this problem, which relies on the MLOP method, and discuss some of its theoretical aspects (e.g., the existence of the solution, uniqueness, and the order of approximation).

There are two main theoretical questions in manifold CS. First, how to design the sensing protocol so as to ensure that it preserves the information in the manifold signal? Second, how can we recover the original signal from the sparse measurements? These two questions should be addressed in the presence of noise and outliers (stemming from imperfect acquisition equipment).

Expressed in mathematical terms, let $\mathcal{M} \subset \mathbb{R}^n$ be a d-dimensional manifold, the signal which we would like to sense and recover. Let $P = \{p_j\}_{j=1}^{J} \subset \mathbb{R}^n$ be a point-set sampled densely from the manifold with noise and outliers. Let Φ be a sensing method, which samples the P-points sparsely, i.e., $Q = \Phi(P)$, where $Q = \{q_i\}_{i=1}^{I} \subset \mathbb{R}^n$ and $I < J$. Then the **Manifold Compressed Sensing Problem** can be formulated as follows: Recover $\mathcal{M}$ with a given density ν from the set Q. Figure 6.1 illustrates all the necessary steps, starting with the sensing and up to the recovery of the manifold.

6.2.1 Manifold Decompression

We will first address the decompression methodology. Given a set of points Q, denoted for brevity by $P' = \{p'_j\}_{j=1}^{J'}$ (where $p'_j = q_i$), we propose to recover $\mathcal{M}$ as a set of points $Q' = \{q'_i\}_{i=1}^{I'} \subset \mathbb{R}^n$ sampled with density ν from the noise-free manifold (where $I' > J'$). In order to recover the manifold reliably, we impose two constraints. First, we demand that the new points keep their proximity to the sampled points. The second constraint deals with uniformly sampling. Thus, we propose to recover the manifold by echoing the signal compressed sensing solution (6.1), using the formulation

$$\min \sum_{q'_i \in Q'} \sum_{p'_j \in P'} \|q'_i - p'_j\|_{H_\epsilon} w_{i,j} \tag{6.2}$$

subject to

$$\min \sum_{q'_i \in Q} \sum_{q'_{i'} \in Q' \setminus \{q'_i\}} \eta(\|q'_i - q'_{i'}\|) \hat{w}_{i,i'} \tag{6.3}$$

where the weights $w_{i,j}$ are given by rapidly decreasing smooth functions. In our implementation we used $w_{i,j} = \exp\left\{-\|q_i - p_j\|^2 / h_1^2\right\}$ and $\hat{w}_{i,i'} = e^{-\frac{\|q'_i - q'_{i'}\|^2}{h_2^2}}$, where the "norm" $\|\cdot\|_{H_\epsilon}$ is defined as $\|v\|_{H_\epsilon} = \sqrt{v^2 + \epsilon}$ [85], where $\epsilon > 0$ is a fixed parameter (in our case we take $\epsilon = 0.1$), and h_1 and h_2 are the support size parameters of $w_{i,k}$ (with $k = i'$ or $k = j$) which guarantee a sufficient amount of P' or Q' points for the reconstruction. For additional details on how to estimate the support size, see Subsection 3.3.2). Also, $\eta(r)$ is a decreasing function such that $\eta(0) = \infty$, in our case we set $\eta(r) = \frac{1}{3r^3}$.

Remark 6.1. *The problem in* (6.2)-(6.3) *is a reformulation of the* MLOP *algorithm introduced in* (3.1). *The flexibility of the* MLOP *method provides out-of-the-box manifold compression without additional effort. Namely, we can choose the number of reconstruction points Q appearing in equation* (3.1)*, and perform manifold compression and recovery. Figure* 6.1 *gives an example of executing* MLOP *for compressing and decompressing a manifold of cylindrical structure embedded to* $\mathbb{R}^{60}$ (*for additional details about the manifold construction see example* 4.5.3). *In this example, we first have a dense sampling of the manifold with* 800 *data points* (*left*)*; the latter are compressed into* 100 *points middle*)*, and finally the manifold structure is*

reconstructed with 800 *points* (*right*).

Following the remark above, we will refer to the sensing and reconstruction methodology as *Compressed Sensing MLOP* (CS-MLOP).

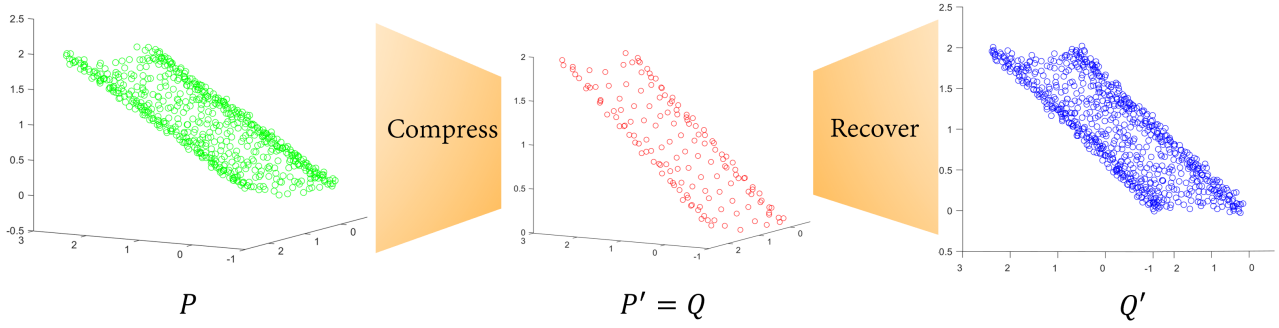


Figure 6.1: Example of compression and recovery of a cylindrical structure embedded into $\mathbb{R}^{60}$. The original sampled manifold with high density (left). 800 points are encoded to a compressed representation of 100 points (middle), and later decoded to a dense representation of 800 points (right).

Practical Considerations. In manifold recovery, the goal is to un compress the information from the manifold (P), with a higher amount of points (Q). While in the sensing problem it is straightforward to form the initial set $Q^{(0)}$ by randomly selecting points from P, in the decompression case it is not clear how to initialize the $Q^{(0)}$-points (since $I > J$). In order to overcome this issue, we initialize the first J points from $Q^{(0)}$ with the P points. Next, the other $I - J$ points are set by randomly selecting points from P and adding a small amount of noise, to avoid point collisions in the constraint (6.3).

6.2.2 Local Manifold Decompression

Another question which arises in the manifold compressed sensing problem, is whether recovering the manifold in a certain area is possible. Thus, instead of reconstructing the entire manifold, recovering only the desired part can save precious time. Accordingly, given a point $x \in R^n$ we would like to recover the manifold in a neighborhood of x of a certain radius r. We modify the equations (6.2), (6.3) and add a new constraint, which limits the approximation to the desired location:

$$\min \sum_{q'_i \in Q'} \sum_{p'_j \in P'} \|q'_i - p'_j\|_{H_\epsilon} w_{i,j} \tag{6.4}$$

subject to

$$\min \sum_{q'_i \in Q} \sum_{q'_{i'} \in Q' \setminus \{q'_i\}} \eta(\|q'_i - q'_{i'}\|)\hat{w}_{i,i'} \quad \text{and} \quad \min \sum_{q'_i \in Q'} e^{\frac{\|q'_i - x\|^2}{r^2}} \tag{6.5}$$

The new term plays the role of an attraction force towards the point x. As a result, all the Q'-points concentrate only in the neighborhood of x. Subsequently, this fact should be taken into account when choosing the number of Q'-points, so that the desired density will be maintained. Once the new term is ready, the solution is found via gradient descent, where the gradient of the new term is given as $\frac{2(q'_i - x)}{r^2} e^{\frac{\|q'_i - x\|^2}{r^2}}$.

6.2.3 Manifold Sensing

A few remarks on the sensing method. The optimal procedure is to sample the manifold without noise and almost uniformly. These two aspects can play a significant role when recovering the signal. Consequently, we propose sensing the manifold using the MLOP methodology, in which the number of Q-points will be smaller than the number of P- points.

6.2.4 Theoretical Aspects of the Method

The CS-MLOP is based on the MLOP methodology, therefore most of the theoretical aspects are inherited from the MLOP. The existence of the solution, the rate of convergence, as well as the complexity of MLOP were extensively dealt with in Section 3.4. In addition, we proved that the rate of approximation is $O(h^2)$.

6.3 Numerical Examples

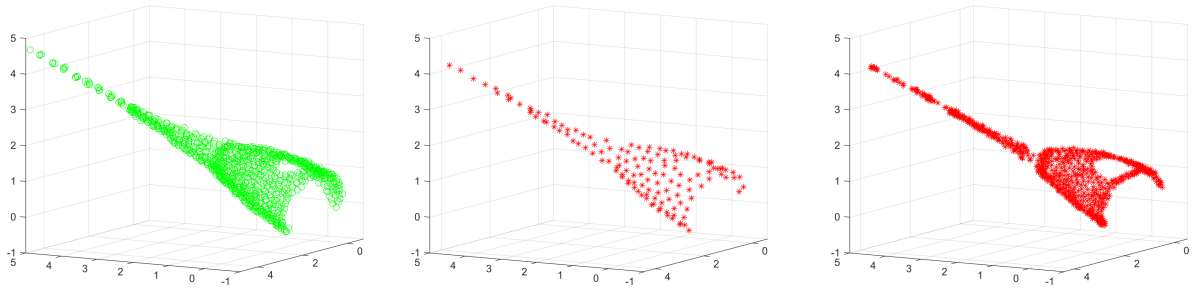
In this section, we demonstrate the sensing and decompression mechanizes on several examples that were introduced in previews chapters. We begin with the cone structure with missing data, used in the Manifold Repairing chapter. In this example, we illustrate the property of the sensing and decompression algorithm for maintaining the manifold geometry.

The geometry we considered is a combination of a 3-dimensional manifold, namely, a cone structure, with a one-dimensional manifold, namely, a line segment. This object was embedded

into a 60-dimensional linear space. The cone's parameterization used was

$$p = tv_1 + \frac{e^{-R^2}}{\sqrt{2}}(\cos(u)v_2 + \sin(u)v_3)\,,$$

where $v_1 = [1,1,1,1,0,\ldots,0], v_2 = [0,1,-1,0,0,\ldots,0], v_3 = [1,0,0,-1,0,\ldots,0]$, $(v_1, v_2, v_3) \in \mathbb{R}^{60}$, $t \in [0,2]$, $R \in [0,2.5]$, and $u \in [0.1\pi, 1.5\pi]$. Then the manifold was sampled and uniform noise $U(-0.1, 0.1)$ was added. In addition, a hole was created in the structure, which resulted in a set P of size 860 (Figure 6.2 left). Next, we sensed the Q set with a compression rate of 5.7, by applying the MLOP method (Figure 6.2 middle). Last, we utilized the compressed data Q to reconstruct the manifold with higher density, with a new point-set of 670 points (Figure 6.2 right). As it can be seen during the sensing and recovering flow, with various densities, the manifold geometry was preserved.



(A) (B) (C)

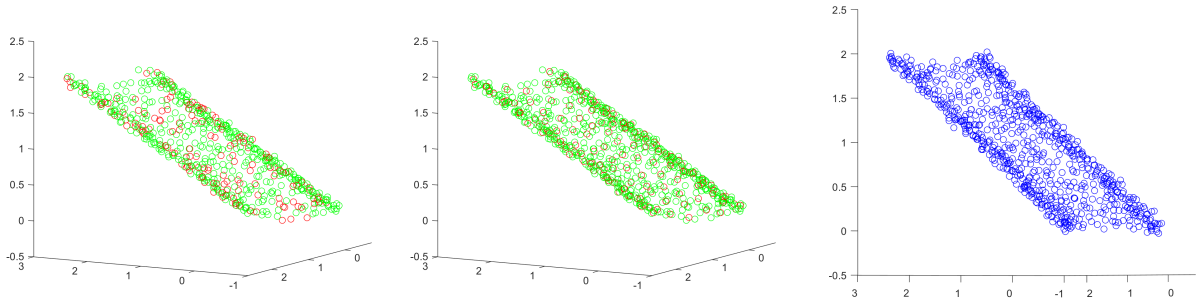
Figure 6.2: Manifold compressed sensing of a cone structure with a hole, embedded into 60-dimensional space. Left: In green are the noisy P-points sampled from the manifold, and the initial compressed data (in red). Middle: the uniformly sensed points generated by executing the MLOP, with a compression rate of 5.7. Right: the decompression of the manifold geometry maintains the geometry of the original structure.

Our next example is a cylindrical structure, a two-dimensional manifold embedded into a 60-dimensional linear space. We sampled the structure using the parameterization

$$p = tv_1 + \frac{R}{\sqrt{2}}(\cos(u)v_2 + \sin(u)v_3)\,,$$

where $v_1 = [1,1,1,1,1,\ldots,1]$, $v_2 = [0,1,-1,0,0,\ldots,0], v_3 = [1,0,0,-1,0,\ldots,0]$ $(v_1, v_2, v_3 \in \mathbb{R}^{60})$, $t \in [0,2]$ and $u \in [0.1\pi, 1.5\pi]$. First we sampled 600 points from the manifold with uniform noise of $U(-0.1, 0.1)$, which resulted in the set P (Figure 6.3 left). Next, we sensed the set Q with compression rate 4, by applying the MLOP method (Figure 6.3 middle). Last,

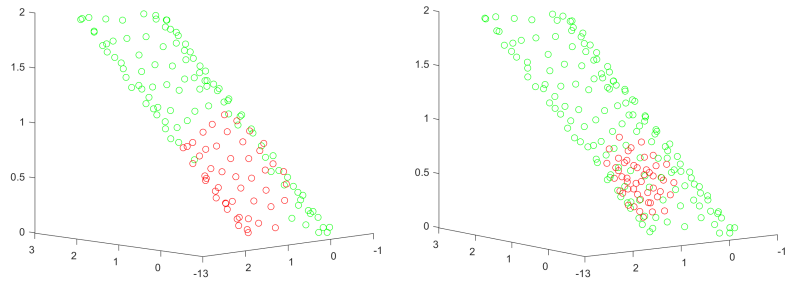
we utilized the compressed data Q for the reconstruction of the manifold with higher density, with a new point-set of 750 points (Figure 6.3 right). This simple example illustrates the capabilities of the manifold remote sensing, in which we are able to sense and decompress the data with a desired density.



(A) (B) (C)

Figure 6.3: Manifold compressed sensing of a cylindrical structure embedded into 60-dimensional space. Left: In green are the noisy P-points sampled from the manifold, and the initial compressed data (in red). Middle: the uniformly sensed points obtained by executing the MLOP, with a compression rate of 4. Right: the decompression of the manifold geometry with a density higher than that of the original P-points.

Subsequently, we demonstrate the ability of reconstructing the manifold locally using the last example of the cylindrical structure. Thus, having sensed the manifold with 150 points which form the set Q, we would like to decompress the manifold locally in a specific location. First, we sample 50 points in the desired area from the Q points (Figure 6.4 left). Next, we use the procedure described in Subsection 6.2.2 for the reconstruction process, with a given reconstruction radius. As can be seen in Figure 6.4 right, the method converges to a local reconstruction, with the desiered radius.



(A) (B)

Figure 6.4: Local manifold recovery illustrated on the cylindrical structure manifold embedded into 60-dimensional space. The original compressed points are shown in green, while the reconstruction is shown in red. Left: The initial point-set sampled around the point of interest. Right: The reconstruction within a specific distance from the point of interest.

Chapter 7

Discussion and Future Research Directions

The big-data era gave rise to many challenges related to processing, analyzing and understanding high-dimensional data. Among these challenges are the presence of noise, outliers, incomplete data, or insufficient data. In this thesis we introduced a set of tools for addressing these issues, raised by high-dimensional data, in an efficient and robust manner. In this chapter, we present a summary of the main tools proposed in the main text. We also present potential further research directions and other developments that the current work may lead to.

In the present research, we addressed the approximation challenges faced when dealing with high-dimensional data. We started by introducing a methodology for reconstructing and denoising high-dimensional data sets sampled from a manifold. Thus, given a set of points sampled from a manifold with noise and outliers, the goal is to reconstruct the original manifold in a noise-free manner. The Manifold Locally Optimal Projection (MLOP) method proposed in this thesis extends the well-known Locally Optimal Projection (LOP) method introduced by Lipman et al. in [90] for approximating surfaces in 3D. In order to overcome several challenges raised by high-dimensional data, various changes are made to the vanilla LOP. The manifold reconstruction method is based on a non-convex optimization problem, which leverages a generalization of the L_1-median to higher dimensions while generating quasi-uniformly distributed

points in the manifold reconstruction. We prove that the MLOP method converges to a local stationary solution with a bounded linear rate of convergence when the starting point is close enough to the local minimum. In addition, we showed that the manifold order of approximation is $O(h^2)$, and the complexity is linear in the ambient dimension and does not depend on the intrinsic dimension.

Next, we addressed another challenge concerning manifold repairing in high-dimension. Given a noisy point-cloud situated near a manifold of unknown intrinsic dimension, with holes, we aim at finding a noise-free reconstruction of the manifold that will amend the holes and complete the missing information. The proposed solution extends the MLOP method by introducing an additional term which fills in the missing data. Our approach is accompanied by a procedure for locating the holes. In addition, we prove the validity of the proposed methodology and demonstrate it on several examples.

Subsequently, we turn to the problem of approximation of functions on a manifold in high dimensions. Give a set of points, and the values of an unknown function evaluated at these points, the goal is to find the approximation of the function on a new dataset. Although in low dimensions this problem did receive a lot of attention in approximation theory, in high dimensions the solution is challenged by the curse of conditionality. In our solution, we propose to combine the best of both worlds, the MLOP and the Radial Basis Functions (RBF). The RBFs can handle high-dimensional data; however, it performs better on uniform clean samples. Using the MLOP method for noise removal and the generation of a quasi-uniform manifold sampling, as a pre-processing step for the RBF improves the approximation results dramatically.

The theoretical analysis of each of the methods proposed in this thesis is accompanied by numerical examples, each stretching the limits of the methodology. We demonstrate the effectiveness of our approaches by considering different manifold topologies, with various intrinsic dimensions and with various amounts of noise and outliers, and show that the methods are not limited to manifolds, but can also handle any geometry by treating the case of a "manifold" of different dimensions at different locations.

Future Research Directions

The approaches discussed in this thesis introduce a set of tools for handling high-dimensional data and open the door to future work. The introduced framework has a high potential of shedding light on how to address various challenges proposed by high-dimensional data. We propose here below a short list of possible research directions to follow in the future.

Classification on a Manifold

A simple out-of-the-box solution is to address the task of classification in high dimensions. By its definition, the MLOP method looks for k-service centers in the data, such that they will minimize the distance from the neighboring points. This results in a k-L_1-medians of the given data. This key point can be used for the classification task, where each point can be classified based on its proximity to one of the k-L_1-medians found. The MLOP advantage of handling noisy data in classification is of special interest.

Optimization on a Manifold

In the past decade, optimization gained a lot of attention, especially with the rise of Neural Network computing (NN). Optimization algorithms are the pillar stones of the NNs, as they are in charge of constructing the networks, by learning from the training examples. In our research, we propose introducing a new optimization process, which will take into account the topology of the data. We would like to utilize the manifold structure of the data to improve the optimization process, by incorporating the manifolds' information into the NN optimization function. We propose extending the MLOP framework to deal with this task, by modifying the definition of the cost function G to include the optimized function $f : \mathcal{M} \to \mathbb{R}^m$. By extending the definition of MLOP algorithm, the gradient descent iterations will find not only the optimal manifold reconstruction, but also minimize the function.

Intrinsic Dimension Estimation

A common assumption, when dealing with high-dimensional data, is that the Intrinsic Dimension (ID) of the data is low. In past decades, the challenge of finding the ID was tackled in many studies, both based on heuristics [23, 124] and geometric considerations (e.g., [38, 69]).

The existing methods can be categorized as local and global, with each approach facing different problems due to the scale it chooses to address the problem. Despite the various suggested solutions, estimating the ID it is still a challenging task for real-life examples. Specifically, the question is *how to find the intrinsic dimension in the presence of noise and outliers?*

We propose utilizing the MLOP framework to estimate the intrinsic dimension. Instead of viewing each point individually, or accompanied by its neighborhood, we suggest analyzing the entire manifold during the reconstruction process. In general, the intrinsic dimension can be viewed as the number of independent parameters that define the geometry of the manifold. Thus, if the intrinsic dimension is known to be d, then the manifold could be represented in d-dimensional space without loss of information. Thus, our goal is to find the smallest dimension of a vector space, where the data could be embedded. During the gradient descent iterations, each point q_i in the desired set Q moves on the original manifold. Thus, we propose analyzing the trajectory of each q_i point during the gradient descent iterations and find the minimal subspace into which this movement can be embedded. Namely, we wish to find the intrinsic dimension of the manifold as the number of the dominant principal components of the trajectory of each point.

Time Series Representation via Manifold Modeling

Time series analysis has become widely used for representing non-stationary data such as financial, epidemic, climatic, as well as criminal records. Time series analysis aims at extracting meaningful information from data that were sequentially generated by a dynamic process. Modeling the non-linear dynamics of a signal is often performed using a linear space. We propose a different approach, where the non-linear dynamics of the time series are represented using functions on manifolds. We will use the new representation to address several main problems in time series: 1) recovering missing data, 2) data denoising, and 3) forecasting.

We suggest introducing a new representation of time series data. Driven by applications, we can regard at the data at time t as a pair of *cause* and *effect*. For example, in crime forecasting [14], weather, political situation and location were utilized as causes of criminal events. Herein we assume that the cause data can in fact be described by a set of variables, and so we can

model its underlying geometry using a manifold. Thus, we can look at the data in time t, as a function on a manifold, where the manifold is the cause data which must be analyzed, and the function is the consequence. In this setting, the time series is represented as a sequence of functions on manifolds (see Figure 7.1. for illustration).

Laid in mathematical terms, let $P_t = \{p_j^t\}_{j\in J} \subset \mathbb{R}^n$ be the cause data acquired, and let $F_t = f_j^t = F(p_j^t) \subset \mathbb{R}$ be the observed outcome generated by the data, where the function F is unknown. Thus, the goal of the proposed research is two-fold: (a) Reconstructing the geometry of the manifolds, of each and every time series; (b) Given a new set of point $P_* = \{p_j^*\}_{j\in J}$, predict the corresponding F_*. We propose using the methodology of approximation of functions on a manifold presented in Chapter 5 to address this task. Thus, we intend to apply the manifold reconstruction methodology for each series in the given data. Thus, given the cause data $P_t = \{p_j^t\}_{j\in J} \subset \mathbb{R}^n$, and the effect data $F_t = f_j^t = F(p_j^t) \subset \mathbb{R}$ we will first find the reconstruction of each such time set, which will result in pairs of $(\hat{P}_t, \hat{F}_t)$. Next, we will address the forecasting challenge using our framework. Given a new set of points $P_* = \{p_j^*\}_{j\in J}$, we would like to find the corresponding values F_*. This will be done using approximation of functions over the high-dimensional data, using the method introduced in Chapter 5.

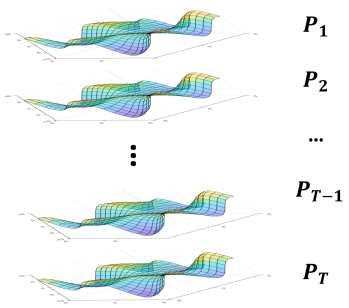


Figure 7.1: Illustration of the proposed time series representation as a series of functions on manifolds.

Bibliography

[1] Aamari, E., Levrard, C., et al.: Nonasymptotic rates for manifold, tangent space and curvature estimation. The Annals of Statistics **47**(1), 177–204 (2019)

[2] Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: International Conference on Database Theory, pp. 420–434. Springer (2001)

[3] Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., Silva, C.T.: Computing and rendering point set surfaces. IEEE Transactions on Visualization and Computer Graphics **9**(1), 3–15 (2003)

[4] Allen-Zhu, Z., Li, Y., Song, Z.: A convergence theory for deep learning via over-parameterization. In: International Conference on Machine Learning, pp. 242–252. PMLR (2019)

[5] Amir, A., Levin, D.: High order approximation to non-smooth multivariate functions. Computer Aided Geometric Design **63**, 31–65 (2018)

[6] Andras, P.: High-dimensional function approximation with neural networks for large volumes of data. IEEE Transactions on Neural Networks and Learning Systems **29**(2), 500–508 (2017)

[7] Attene, M., Campen, M., Kobbelt, L.: Polygon mesh repairing: An application perspective. ACM Computing Surveys (CSUR) **45**(2), 15 (2013)

[8] Bachmayr, M., Dahmen, W., DeVore, R., Grasedyck, L.: Approximation of high-

dimensional rank one tensors. Constructive Approximation **39**(2), 385–395 (2014)

[9] Baraniuk, R.G.: Compressive sensing [lecture notes]. IEEE Signal Processing Magazine **24**(4), 118–121 (2007)

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

[11] Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA Journal of Numerical Analysis **8**(1), 141–148 (1988)

[12] Bendels, G.H., Schnabel, R., Klein, R.: Detecting holes in point set surfaces. The Journal of WSCG **14** (2006)

[13] Berger, M., Tagliasacchi, A., Seversky, L.M., Alliez, P., Guennebaud, G., Levine, J.A., Sharf, A., Silva, C.T.: A survey of surface reconstruction from point clouds. In: Computer Graphics Forum, vol. 36, pp. 301–329 (2017)

[14] Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, pp. 417–424 (2000)

[15] Bickel, P.J., Li, B., et al.: Local polynomial regression on unknown manifolds. In: Complex Datasets and Inverse Problems, pp. 177–186. Institute of Mathematical Statistics (2007)

[16] Binev, P., Dahmen, W., Lamby, P.: Fast high-dimensional approximation with sparse occupancy trees. Journal of Computational and Applied Mathematics **235**(8), 2063–2076 (2011)

[17] Boissonnat, J., Ghosh, A.: Manifold reconstruction using tangential Delaunay complexes. Discrete & Computational Geometry **51**(1), 221–267 (2014)

[18] Boissonnat, J., Guibas, L.J., Oudot, S.Y.: Manifold reconstruction in arbitrary dimensions using witness complexes. Discrete & Computational Geometry **42**(1), 37–70 (2009)

[19] Boissonnat, J.D.: Geometric structures for three-dimensional shape representation. ACM Transactions on Graphics (TOG) **3**(4), 266–286 (1984)

[20] Brown, B.: Statistical uses of the spatial median. Journal of the Royal Statistical Society: Series B (Methodological) **45**(1), 25–30 (1983)

[21] Buhmann, M.D.: Radial Basis Functions: Theory and Implementations, vol. 12. Cambridge University Press, Cambridge Monographs on Applied and Computational Science (2003)

[22] Buhmann, M.D., De Marchi, S., Perracchione, E.: Analysis of a new class of rational RBF expansions. IMA Journal of Numerical Analysis **40**(3), 1972–1993 (2020)

[23] Camastra, F.: Data dimensionality estimation methods: a survey. Pattern Recognition **36**(12), 2945–2954 (2003)

[24] Candes, E.J., Plan, Y.: Matrix completion with noise. Proceedings of the IEEE **98**(6), 925–936 (2010)

[25] Candès, E.J., Romberg, J., Tao, T.: Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on Information Theory **52**(2), 489–509 (2006)

[26] Candes, E.J., Tao, T.: Near-optimal signal recovery from random projections: Universal encoding strategies? IEEE Transactions on Information Theory **52**(12), 5406–5425 (2006)

[27] Candes, E.J., et al.: The restricted isometry property and its implications for compressed sensing. Comptes Rendus Mathematique **346**(9-10), 589–592 (2008)

[28] Chalmovianskỳ, P., Jüttler, B.: Filling holes in point clouds. In: Mathematics of Surfaces, pp. 196–212. Springer (2003)

[29] Chazal, F., Cohen-Steiner, D., Mérigot, Q.: Geometric inference for probability measures. Foundations of Computational Mathematics **11**(6), 733–751 (2011)

[30] Chen, G., Little, A.V., Maggioni, M.: Multi-resolution geometric analysis for data in high dimensions. In: Excursions in Harmonic Analysis, Volume 1, pp. 259–285. Springer (2013)

[31] Chen, M., Jiang, H., Liao, W., Zhao, T.: Efficient approximation of deep ReLU networks for functions on low dimensional manifolds. In: Advances in Neural Information Processing Systems, pp. 8174–8184 (2019)

[32] Chen, M., Silva, J., Paisley, J., Wang, C., Dunson, D., Carin, L.: Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds. IEEE Transactions on Signal Processing **58**(12), 6140–6155 (2010)

[33] Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Review **43**(1), 129–159 (2001)

[34] Cheng, S.W., Dey, T.K., Ramos, E.A.: Manifold reconstruction from point samples. In: SODA, vol. 5, pp. 1018–1027 (2005)

[35] Cohen-Or, D., Levin, D., Remez, O.: Progressive compression of arbitrary triangular meshes. Proceedings of Visualization '99, IEEE (1999)

[36] Coifman, R.R., Lafon, S., Lee, A.B., Maggioni, M., Nadler, B., Warner, F., Zucker, S.W.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. Proceedings of the National Academy of Sciences **102**(21), 7426–7431 (2005)

[37] Coifman, R.R., Maggioni, M.: Diffusion wavelets. Applied and Computational Harmonic Analysis **21**(1), 53–94 (2006)

[38] Costa, J.A., Girotra, A., Hero, A.: Estimating local intrinsic dimension with k-nearest neighbor graphs. In: IEEE/SP 13th Workshop on Statistical Signal Processing, 2005, pp. 417–422. IEEE (2005)

[39] Cox, T.F., Cox, M.A.: Multidimensional Scaling. Chapman and Hall, London (2000)

[40] Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Transactions on image processing **13**(9), 1200–1212 (2004)

[41] Davenport, M.A., Duarte, M.F., Eldar, Y.C., Kutyniok, G.: Introduction to compressed sensing. 1. In: Compressed Sensing. Theory and Applications, Cambridge University Press. (2012)

[42] Davis, J., Marschner, S.R., Garr, M., Levoy, M.: Filling holes in complex surfaces using volumetric diffusion. In: Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission, pp. 428–441. IEEE (2002)

[43] DeVore, R., Petrova, G., Wojtaszczyk, P.: Approximation of functions of few variables in high dimensions. Constructive Approximation **33**(1), 125–143 (2011)

[44] Domingos, P.M.: A few useful things to know about machine learning. Commun. ACM **55**(10), 78–87 (2012)

[45] Donoho, D.L., Elad, M.: Optimally sparse representation in general (nonorthogonal) dictionaries via L1 minimization. Proceedings of the National Academy of Sciences **100**(5), 2197–2202 (2003)

[46] Donoho, D.L., et al.: Compressed sensing. IEEE Transactions on Information Theory **52**(4), 1289–1306 (2006)

[47] Dyn, N., Levin, D.: Iterative solution of systems originating from integral equations and surface interpolation. SIAM Journal on Numerical Analysis **20**(2), 377–390 (1983)

[48] Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image processing **15**(12), 3736–3745 (2006)

[49] Eldar, Y.C., Kutyniok, G. (eds.): Compressed Sensing: Theory and Applications. Cambridge University Press (2012)

[50] Faigenbaum, S., Shaus, A., Sober, B., Turkel, E., Piasetzky, E.: Evaluating glyph binarizations based on their properties. In: Proceedings of the 2013 ACM symposium on

Document engineering, pp. 127–130. ACM (2013)

[51] Faigenbaum, S., Sober, B., Finkelstein, I., Moinester, M., Piasetzky, E., Shaus, A., Cordonsky, M.: Multispectral imaging of two hieratic inscriptions from Qubur el-Walaydah. Ägypten und Levante/Egypt and the Levant pp. 349–353 (2014)

[52] Faigenbaum, S., Sober, B., Moinester, M., Piasetzky, E., Bearman, G.: Multispectral imaging of Tel Malhata ostraca. Tel Malhata: a central city in the biblical Negev **1**, 510–513 (2015)

[53] Faigenbaum, S., Sober, B., Shaus, A., Moinester, M., Piasetzky, E., Bearman, G., Cordonsky, M., Finkelstein, I.: Multispectral images of ostraca: Acquisition and analysis. Journal of Archaeological Science **39**(12), 3581–3590 (2012)

[54] Faigenbaum-Golovin, S.: Anisotropic moving least squares. Master's thesis, School of Mathematical Sciences, Tel-Aviv University (2014)

[55] Faigenbaum-Golovin, S., Levin, D.: Approximation of functions over manifolds in high dimension from noisy scattered data (2020). forthcoming

[56] Faigenbaum-Golovin, S., Levin, D.: Manifold compressed sensing: reconstruction of high-dimensional scattered data from highly incomplete information (2020). forthcoming

[57] Faigenbaum-Golovin, S., Levin, D.: Manifold reconstruction and denoising from scattered data in high dimension via a generalization of L_1-median (2020). forthcoming

[58] Faigenbaum-Golovin, S., Levin, D.: Manifold repairing, reconstruction and denoising from scattered data in high dimension (2020). forthcoming

[59] Faigenbaum-Golovin, S., Levin, D., Piasetzky, E., Finkelstein, I.: Writer characterization and identification of short modern and historical documents: Reconsidering paleographic tables. In: Proceedings of the ACM Symposium on Document Engineering 2019, pp. 1–4 (2019)

[60] Faigenbaum-Golovin, S., Mendel-Geberovich, A., Shaus, A., Sober, B., Cordonsky, M.,

Levin, D., Moinester, M., Sass, B., Turkel, E., Piasetzky, E., et al.: Multispectral imaging reveals biblical-period inscription unnoticed for half a century. PLOS ONE **12**(6), e0178400 (2017)

[61] Faigenbaum-Golovin, S., Rollston, C.A., Piasetzky, E., Sober, B., Finkelstein, I.: The Ophel (Jerusalem) ostracon in light of new multispectral images. Semitica **57**, 113–137 (2015)

[62] Faigenbaum-Golovin, S., Shaus, A., Sober, B., Finkelstein, I., Levin, D., Moinester, M., Piasetzky, E., Turkel, E.: Computerized paleographic investigation of Hebrew Iron Age ostraca. Radiocarbon **57**(2), 317–325 (2015)

[63] Faigenbaum-Golovin, S., Shaus, A., Sober, B., Levin, D., Na'aman, N., Sass, B., Turkel, E., Piasetzky, E., Finkelstein, I.: Algorithmic handwriting analysis of judah's military correspondence sheds light on composition of biblical texts. Proceedings of the National Academy of Sciences **113**(17), 4664–4669 (2016)

[64] Faigenbaum-Golovin, S., Shaus, A., Sober, B., Turkel, E., Piasetzky, E., Finkelstein, I.: Algorithmic handwriting analysis of the Samaria inscriptions illuminates bureaucratic apparatus in biblical israel. PLOS ONE **15**(1), e0227452 (2020)

[65] Federer, H.: Curvature measures. Transactions of the American Mathematical Society **93**(3), 418–491 (1959)

[66] Fefferman, C., Ivanov, S., Kurylev, Y., Lassas, M., Narayanan, H.: Fitting a putative manifold to noisy data. In: Conference on Learning Theory, pp. 688–720 (2018)

[67] Finkelstein, I., Evian, S.B.D., Boaretto, E., Cabanes, D., Cabanes, M.T., Eliyahu-Behar, A., Faigenbaum, S., Gadot, Y., Langgut, D., Martin, M., et al.: Reconstructing ancient Israel: integrating macro-and micro-archaeology. Hebrew Bible and Ancient Israel **1**(1), 133–150 (2012)

[68] Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics **7**(2), 179–188 (1936)

[69] Granata, D., Carnevale, V.: Accurate estimation of the intrinsic dimension using graph distances: Unraveling the geometric complexity of datasets. Scientific Reports **6**, 31377 (2016)

[70] Grohs, P., Perekrestenko, D., Elbrächter, D., Bölcskei, H.: Deep neural network approximation theory. arXiv preprint arXiv:1901.02220 (2019)

[71] Grohs, P., Sprecher, M.: Projection-based quasiinterpolation in manifolds. SAM Report **23** (2013)

[72] Guo, X., Xiao, J., Wang, Y.: A survey on algorithms of hole filling in 3d surface reconstruction. The Visual Computer **34**(1), 93–103 (2018)

[73] Harrison, J.: Soap film solutions to Plateau's problem. Journal of Geometric Analysis **24**(1), 271–297 (2014)

[74] He, G., Cheng, X.: A virtual restoration strategy of 3d scanned objects. In: Advances in Computer Science, Intelligent System and Environment, pp. 621–627. Springer (2011)

[75] He, X., Niyogi, P.: Locality preserving projections. In: Advances in Neural Information Processing Systems, pp. 153–160 (2004)

[76] Huang, H., Li, D., Zhang, H., Ascher, U., Cohen-Or, D.: Consolidation of unorganized point clouds for surface reconstruction. ACM Transactions on Graphics (TOG) **28**(5), 176 (2009)

[77] Huang, H., Wu, S., Gong, M., Cohen-Or, D., Ascher, U., Zhang, H.R.: Edge-aware point set resampling. ACM Transactions on Graphics (TOG) **32**(1), 9 (2013)

[78] Iyengar, S.S., Kouri, D.J., Parker, G.A., Hoffman, D.K.: Estimating bounds on the highest and lowest eigenvalues of any matrix. Theoretical Chemistry Accounts **103**(6), 507–517 (2000)

[79] Jun, Y.: A piecewise hole filling algorithm in reverse engineering. Computer-Aided Design **37**(2), 263–270 (2005)

[80] Kearney, A., Veeriah, V., Travnik, J., Pilarski, P.M., Sutton, R.S.: Learning feature relevance through step size adaptation in temporal-difference learning. arXiv preprint arXiv:1903.03252 (2019)

[81] Kotel'nikov, V.A.: On the carrying capacity of the "ether" and wire in telecommunications. In: Materials of the First All-Union Conference on Questions of Communication (Russian), Izd. Red. Upr. Svyzai RKKA, Moscow, 1933 (1933)

[82] Lee, J.D., Simchowitz, M., Jordan, M.I., Recht, B.: Gradient descent only converges to minimizers. In: Conference on Learning Theory, pp. 1246–1257 (2016)

[83] Levin, D.: The approximation power of moving least-squares. Mathematics of Computation **67**(224), 1517–1531 (1998)

[84] Levin, D.: Mesh-independent surface interpolation. In: Geometric Modeling for Scientific Visualization, pp. 37–49. Springer (2004)

[85] Levin, D.: Between moving least-squares and moving least-ℓ_1. BIT Numerical Mathematics **55**(3), 781–796 (2015)

[86] Levoy, M., Gerth, J., Curless, B., Pull, K.: The Stanford 3d scanning repository. URL http://www-graphics.stanford.edu/data/3dscanrep **5** (2005)

[87] Li, L.: A new complexity bound for the least-squares problem. Computers & Mathematics with Applications **31**(12), 15–16 (1996)

[88] Lin, T., Zha, H.: Riemannian manifold learning. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(5), 796–809 (2008)

[89] Lipman, Y., Cohen-Or, D., Levin, D.: Error bounds and optimal neighborhoods for MLS approximation. In: Proceedings of the fourth Eurographics Symposium on Geometry Processing, pp. 71–80 (2006)

[90] Lipman, Y., Cohen-Or, D., Levin, D., Tal-Ezer, H.: Parameterization-free projection for geometry reconstruction. In: ACM Transactions on Graphics (TOG), vol. 26, p. 22.

ACM (2007)

[91] Lopuhaa, H.P., Rousseeuw, P.J., et al.: Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. The Annals of Statistics **19**(1), 229–248 (1991)

[92] Maggioni, M., Minsker, S., Strawn, N.: Multiscale dictionary learning: non-asymptotic bounds and robustness. The Journal of Machine Learning Research **17**(1), 43–93 (2016)

[93] Mahmoudi, M., Sapiro, G.: Fast image and video denoising via nonlocal means of similar neighborhoods. IEEE Signal Processing Letters **12**(12), 839–842 (2005)

[94] Mendel-Geberovich, A., Faigenbaum-Golovin, S., Shaus, A., Sober, B., Cordonsky, M., Piasetzky, E., Finkelstein, I., Milevski, I.: A renewed reading of Hebrew ostraca from cave a-2 at Ramat Beit Shemesh (Nahal Yarmut), based on multispectral imaging. Vetus Testamentum **69**(4-5), 682–701 (2019)

[95] Mendel-Geberovich, A., Shaus, A., Faigenbaum-Golovin, S., Sober, B., Cordonsky, M., Piasetzky, E., Finkelstein, I.: A brand new old inscription: Arad ostracon 16 rediscovered via multispectral imaging. Bulletin of the American Schools of Oriental Research **378**(1), 113–125 (2017)

[96] Milasevic, P., Ducharme, G.: Uniqueness of the spatial median. Annals of Statistics **15**(3), 1332–1333 (1987)

[97] Moenning, C., Dodgson, N.A.: Intrinsic point cloud simplification. Proc. 14th GraphiCon **14**, 23 (2004)

[98] Nesterov, Y.: Lectures on Convex Optimization, Springer Optimization and Its Applications, vol. 137. Springer (2018)

[99] Niyogi, P., Smale, S., Weinberger, S.: Finding the homology of submanifolds with high confidence from random samples. Discrete & Computational Geometry **39**(1-3), 419–441 (2008)

[100] Nyquist, H.: Certain topics in telegraph transmission theory. Transactions of the American Institute of Electrical Engineers **47**(2), 617–644 (1928)

[101] Ostrov, D.N.: Boundary conditions and fast algorithms for surface reconstructions from synthetic aperture radar data. IEEE transactions on geoscience and remote sensing **37**(1), 335–346 (1999)

[102] Papyan, V.: The full spectrum of deepnet hessians at scale: Dynamics with SGD training and sample size. arXiv preprint arXiv:1811.07062 (2018)

[103] Pearson, K.: LIII. On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**(11), 559–572 (1901)

[104] Reznikov, A., Saff, E.: The covering radius of randomly distributed points on a manifold. International Mathematics Research Notices **2016**(19), 6065–6094 (2016)

[105] Richard, M.M.O.B.B., Chang, M.Y.S.: Fast digital image inpainting. In: Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2001), Marbella, Spain, pp. 106–107 (2001)

[106] Rolnick, D., Veit, A., Belongie, S., Shavit, N.: Deep learning is robust to massive label noise. arXiv preprint arXiv:1705.10694 (2017)

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

[108] Shaham, U., Cloninger, A., Coifman, R.R.: Provable approximation properties for deep neural networks. Applied and Computational Harmonic Analysis **44**(3), 537–557 (2018)

[109] Shannon, C.E.: Communication in the presence of noise. Proceedings of the IRE **37**(1), 10–21 (1949)

[110] Shaus, A., Faigenbaum-Golovin, S., Sober, B., Turkel, E.: Potential contrast - a new image quality measure. Electronic Imaging **2017**(12), 52–58 (2017)

[111] Shaus, A., Gerber, Y., Faigenbaum-Golovin, S., Sober, B., Piasetzky, E., Finkelstein, I.: Forensic document examination and algorithmic handwriting analysis of Judahite biblical period inscriptions reveal significant literacy level. PLOS ONE **15**(9), 1–15 (2020). DOI 10.1371/journal.pone.0237962. URL `https://doi.org/10.1371/journal.pone.0237962`

[112] Shaus, A., Sober, B., Faigenbaum-Golovin, S., Mendel-Geberovich, A., Levin, D., Piasetzky, E., Turkel, E.: Statistical inference in archaeology: Are we confident? Rethinking Israel: Studies in the History and Archaeology of Ancient Israel in Honor of Israel Finkelstein, Eisenbrauns, Winona Lake pp. 389–401 (2017)

[113] Shaus, A., Sober, B., Faigenbaum-Golovin, S., Mendel-Geberovich, A., Piasetzky, E., Turkel, E.: Facsimile creation: Review of algorithmic approaches. Alphabets, Texts and Artefacts in the Ancient Near East, Studies Presented to Benjamin Sass, edited by I. Finkelstein, C. Robin, and T. Römer (Van Dieren Éditeur, Paris) (2016)

[114] Singer, A., Zhao, Z., Shkolnisky, Y., Hadani, R.: Viewing angle classification of cryo-electron microscopy images using eigenvectors. SIAM Journal on Imaging Sciences **4**(2), 723–759 (2011)

[115] Small, C.G.: A survey of multidimensional medians. International Statistical Review/Revue Internationale de Statistique, pp. 263–277 (1990)

[116] Sober, B., Aizenbud, Y., Levin, D.: Approximation of functions over manifolds: A moving least-squares approach. arXiv preprint arXiv:1711.00765 (2017)

[117] Sober, B., Faigenbaum, S., Beit-Arieh, I., Finkelstein, I., Moinester, M., Piasetzky, E., Shaus, A.: Multispectral imaging as a tool for enhancing the reading of ostraca. Palestine Exploration Quarterly **146**(3), 185–197 (2014)

[118] Sober, B., Levin, D.: Manifold approximation by moving least-squares projection (MMLS). arXiv preprint arXiv:1606.07104 (2016)

[119] Starck, J.L., Candès, E.J., Donoho, D.L.: The curvelet transform for image denoising.

IEEE Transactions on Image Processing **11**(6), 670–684 (2002)

[120] Su, Z.x., Li, Z.y., Cao, J.j., et al.: Curvature-aware simplification for point-sampled geometry. Journal of Zhejiang University SCIENCE C **12**(3), 184–194 (2011)

[121] Tang, J., Wang, Y., Zhao, Y., Hao, W., Ning, X., Lv, K.: A repair method of point cloud with big hole. In: 2017 International Conference on Virtual Reality and Visualization (ICVRV), pp. 79–84. IEEE (2017)

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

[123] Thi, D.T., Fomenko, A.T., Primrose, E., Silver, B.: Minimal Surfaces, Stratified Multivarifolds, and the Plateau Problem, vol. 84. Transactions of mathematical Monographs, American Mathematical Society (1991)

[124] Van Der Maaten, L., Postma, E., Van den Herik, J.: Dimensionality reduction: a comparative. J Mach Learn Res **10**(66-71), 13 (2009)

[125] Vardi, Y., Zhang, C.H.: The multivariate L1-median and associated data depth. Proceedings of the National Academy of Sciences **97**(4), 1423–1426 (2000)

[126] Wang, J., Oliveira, M.M.: Filling holes on locally smooth surfaces reconstructed from point clouds. Image and Vision Computing **25**(1), 103–113 (2007)

[127] Wang, T.H., Krishnamurti, R., Shimada, K.: Restructuring surface tessellation with irregular boundary conditions. Frontiers of Architectural Research **3**(4), 337–347 (2014)

[128] Weber, A.: Theory of the Location of Industries. University of Chicago Press (1929)

[129] Weiszfeld, E.: Sur le point pour lequel la somme des distances de n points donnés est minimum. Tohoku Mathematical Journal, First Series **43**, 355–386 (1937)

[130] Whittaker, E.T.: XVIII. -On the functions which are represented by the expansions of the interpolation-theory. Proceedings of the Royal Society of Edinburgh **35**, 181–194 (1915)

[131] Woodruff, D.P., et al.: Sketching as a tool for numerical linear algebra. Foundations and Trends in Theoretical Computer Science **10**(1–2), 1–157 (2014)

[132] Wright, S.J., Nowak, R.D., Figueiredo, M.A.: Sparse reconstruction by separable approximation. IEEE Transactions on signal processing **57**(7), 2479–2493 (2009)

[133] Wu, Zong-min., Schaback, R.: Local error estimates for radial basis function interpolation of scattered data. IMA journal of Numerical Analysis **13**(1), 13–27 (1993)

[134] Xu, Z., Sun, J.: Image inpainting by patch propagation using patch sparsity. IEEE Transactions on Image Processing **19**(5), 1153–1165 (2010)

[135] Yadav, S.K., Reitebuch, U., Skrodzki, M., Zimmermann, E., Polthier, K.: Constraint-based point set denoising using normal voting tensor and restricted quadratic error metrics. Computers & Graphics **74**, 234–243 (2018)

[136] Yaron, O., Faigenbaum-Golovin, S., Granot, A., Shkolnisky, Y., Goldshleger, N., Eyal, B.D.: Removing moisture effect on soil reflectance properties: A case study of clay content prediction. Pedosphere **29**(4), 421–431 (2019)

[137] Yoon, J.: Spectral approximation orders of radial basis function interpolation on the Sobolev space. SIAM Journal on Mathematical Analysis **33**(4), 946–958 (2001)

[138] Zhang, L., Wei, W., Zhang, Y., Li, F., Shen, C., Shi, Q.: Hyperspectral compressive sensing using manifold-structured sparsity prior. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3550–3558 (2015)

[139] Zhou, D., Jiang, C., Dong, J., LIU, R.: Algorithm of detecting and filling small holes in triangular mesh surface. Computer Aided Drafting, Design and Manufacturing **4**, 33–38 (2014)