Searching, Clustering and Regression on non-Euclidean Spaces

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Xu Wang

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Gilad Lerman

August, 2015

Acknowledgements

I would like to thank my adviser Gilad Lerman for his help and thoughtful advice. This work was supported by NSF CAREER award of Professor Lerman: DMS-09-56072 and Doctoral Dissertation Fellowship of Xu Wang. Professor Lerman has also contributed a lot in this work. I would like to thank my collaborators Stefan Atev, John Wright and Konstantinos Slavakis for helpful discussions. I would like to thank Gilad's research group, from which I got a lot of help. I am especially grateful to Chenyan Wu for her support and many interesting discussions during the years.

Dedication

To my parents and grandparents

Abstract

This is a collection of the works I have done during my PhD research at the University of Minnesota. There are three parts dedicated to different topics, of which abstracts are included below.

Abstract for Part I
The problem of efficiently deciding which of a database of models is most similar to a given input query arises throughout modern computer vision. Motivated by applications in recognition, image retrieval and optimization, there has been significant recent interest in the variant of this problem in which the database models are linear subspaces and the input is either a point or a subspace. Current approaches to this problem have poor scaling in high dimensions, and may not guarantee sublinear query complexity. We present a new approach to approximate nearest subspace search, based on a simple, new locality sensitive hash for subspaces. Our approach allows point-to-subspace query for a database of subspaces of arbitrary dimension d, in a time that depends sublinearly on the number of subspaces in the database. The query complexity of our algorithm is linear in the ambient dimension D, allowing it to be directly applied to high-dimensional imagery data. Numerical experiments on model problems in image repatching and automatic face recognition confirm the advantages of our algorithm in terms of both speed and accuracy.

Abstract for Part II
This part advocates a novel framework for segmenting a dataset in a Riemannian manifold M into clusters lying around low-dimensional submanifolds of M. Important examples of M, for which the proposed clustering algorithm is computationally efficient, are the sphere, the set of positive definite matrices, and the Grassmannian. The clustering problem with these examples of M is already useful for numerous application domains such as action identification in video sequences, dynamic texture clustering, brain fiber segmentation in medical imaging, and clustering of deformed images. The proposed

clustering algorithm constructs a data-affinity matrix by thoroughly exploiting the intrinsic geometry and then applies spectral clustering. The intrinsic local geometry is encoded by local sparse coding and more importantly by directional information of local tangent spaces and geodesics. Theoretical guarantees are established for a simplified variant of the algorithm even when the clusters intersect. To avoid complication, these guarantees assume that the underlying submanifolds are geodesic. Extensive validation on synthetic and real data demonstrates the resiliency of the proposed method against deviations from the theoretical model as well as its superior performance over state-of-the-art techniques.

Abstract for Part III

This part proposes a novel framework for manifold-valued regression and establishes its consistency as well as its contraction rate for a particular setting. Our setting assumes a predictor with values in the interval $[0, 1]$ and response with values in a compact Riemannian manifold M. This setting is useful for applications such as modeling dynamic scenes or shape deformations, where the visual scene or the deformed objects can be modeled by a manifold. The proposed framework is nonparametric and uses the heat kernel on manifolds as an averaging procedure. It directly generalizes the use of the Gaussian kernel (as a natural model of additive noise) in vector-valued regression problems. In order to avoid explicit dependence on estimates of the heat kernel, we follow a Bayesian setting, where Brownian motion on M induces a prior distribution on the space of continuous functions $C([0, 1], M)$. For the case of discretized Brownian motion, we establish the consistency of the posterior distribution in terms of the L_q distances for any $1 \leq q < \infty$. Most importantly, we establish contraction rate of order $O(n^{-1/4+\epsilon})$ for any $\epsilon > 0$. For the continuous Brownian motion we establish weak consistency. As a by-product, we show that the Brownian motion prior Π possesses positive probabilities over the L_∞ neighborhoods of any continuous function in $C([0, 1], M)$.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the ever increasing capability of collecting data, the challenges of data analysis have made the presence not only in the form of huge volumes, but also in the form of "non-Euclidean" (manifold) represented data. This work focuses on developing analytical tools for studying datasets by taking into account their underlying geometric structures. It consists of three parts, which answer fundamental questions arising from different application domains. The first two parts establish novel methods of treating two fundamental tasks in pattern analysis. The first one is searching on manifolds, that is, finding the nearest object on a manifold to a given object; it is an essential problem in recognition. The second one is manifold clustering in non-Euclidean spaces, more specifically, segmenting a dataset lying on (or around) a mixture of submanifolds of a Riemannian manifold. This clustering problem arises for example in video segmentation and brain imaging. It is advantageous over common basic methods that embed the dataset in a Euclidean space and often ignore the intrinsic geometry. The third part studies the Bayesian regression (or smoothing) of the data subject to fixed manifold constraints. This problem is motivated by the recent demand of analyzing shape (or landmark) spaces and time series of images reflecting an aging process, brain development or disease progression. The diffusion process over a constraint domain (manifold) is applied to the data so that the underlying constraint is preserved in the smoothing process.

Chapter 2

Part I: Fast Subspace Search via Grassmannian Based Hashing

[1]

2.1 Introduction

Given a very large database of models, how can we efficiently determine which one that best fits a given input query? This basic question arises repeatedly in computer vision applications such as visual recognition, categorization, image retrieval and beyond. These applications pose two general challenges to the algorithm designer: imagery data (and their features) are typically *high-dimensional*, and databases arising in applications can be very *large scale*.

The large scale often precludes simply comparing the query to each of the models in any reasonable amount of time. Instead, researchers typically resort to more sophisticated *approximate nearest neighbor* techniques, whose query time which is sublinear in the size of the database. For the case in which the query is a vector and the database is also a collection of vectors, these techniques are very well-developed, in both theory and practice [1, 2, 3, 4].

[1] A portion of Part I includes textual material and figures from X. Wang, S. Atev, J. Wright and G. Lerman, Fast Subspace Search via Grassmannian Based Hashing, Computer Vision (ICCV), 2013 IEEE International Conference on, Dec. 2013 under ©2013 IEEE.

However, data in computer vision problems often have rich physical or geometric structure, which may not be well-encoded using point models. For example, photometric or textural properties of a collection of images can often be better represented using linear or affine *subspaces*, rather than a simple point model. In the *approximate nearest subspace* problem, we are given a collection of linear subspaces. The goal is to efficiently determine which of the database subspaces is closest to the input [5]. Good solutions to this problem would allow us to efficiently query large databases which contain much richer representations.

In contrast to approximate nearest neighbor, both the theory and practice of approximate nearest subspace are still developing. The most general known approach is due to Basri et. al. [5]. It maps each subspace $S \subseteq \mathbb{R}^D$ to its orthogonal projection matrix $\mathbf{P}_S$, and then applies an approximate nearest neighbor algorithm to the projection matrices. The advantage of this approach is that it cleanly reduces the subspace problem to the better-understood point search problem. However, because the projection matrix has size $\Theta(D^2)$, the algorithm's performance suffers in high dimensions.[2] Moreover, the mapping from subspace to orthoprojector does not preserve distances (for subspaces of different dimensions), and so performance guarantees for the approximate nearest neighbor algorithm may not pull back to the approximate nearest subspace problem. Algorithms with sublinear query time, but exponential dependence on dimension have also been introduced in [6].

Motivated in part by [5], there has been a flurry of recent work on special cases of this problem. For example the case in which the query is a point and the database subspaces are hyperplanes (dimension $d = D - 1$) has been studied in connection to active learning and large-scale regression [7]. Various approaches based on locality sensitive hashing have been proposed [7, 8, 9]. While various technical obstacles prevent these approaches from guaranteeing sublinear query complexity over all inputs, they have been used effectively in various practical vision problems. In the algorithms community, there is also dedicated work on the special case in which the queries are points and the database consists of affine lines ($d = 1$). For example, Andoni et. al. produce a data structure for this problem that has query time $O(D^3 n^{1/2+t})$ and space complexity $D^2 n^{O(1/(c-1)^2+1/t^2)}$, for any $t > 0$ [10]. Again, the query time $O(D^3)$ could be

[2] [5] suggest using random projections after lifting as one means of controlling the complexity.

problematic in large dimensions.

Moreover, many of the most interesting models for computer vision have a dimension d that falls somewhere in between 1 and $D-1$. For example, linear subspaces spanned by images taken under varying lighting may have dimension between 3 and 9, depending on the properties of the object [11]. Local image patches also typically lie near subspaces of dimension higher than one [12, 13]. So, despite the above progress, there is still a need for algorithms that can guarantee a query time that is sublinear in the number of models n, have good (linear or sublinear) dependence on the ambient dimension D, and can handle the case when the input is a point and the database consists of subspaces of arbitrary dimension d.

Contributions. In this paper, we provide a solution of the approximate nearest subspace (ANS) search problem based on the notion of *locality sensitive hashing* (see e.g., [3]). Our theoretical guarantees for the sub-linear complexity and preprocessing space of our solution distinguish between three types of searches: line-line query (this is equivalent to point-line query as explained in §2.2; it is also equivalent with point-line and point-point queries when the points lie on the sphere); line-subspace search (this is equivalent to point-subspace query as explained in §2.2; and subspace-subspace query. For all of these searches, our preprocessing space is $O(n^{1+\rho} + nDd)$ and query time is $O(Dn^{\rho})$, where d is the largest dimension of subspaces among both query elements and the database elements, D is the ambient dimension and $\rho < 1$. Nevertheless, the precise formulations and their corresponding estimates are different for the three types of searches. For example, each case has a different bound on the maximal possible distance between a given query point and the database and different estimates for its underlying parameters.

When $d = 1$ (that is, for line-line and point-line queries or even for point-point and line-point queries when points are on the sphere) we can remove the asymptotic superlinear dependence of the query time on dimension D under some condition on the query element and the database (that is, for each query element there is a sufficiently close element in the database).

Our theoretical setting is designed to address recognition problems. For example, our unorthodox restriction on the maximal distance between query element and the database (this appears only in some of our statements) can often be met in practice, where query points may be contained in or be sufficiently close to the database. We confirmed in practice the competitive speed and accuracy of our proposed solution on model problems in image repatching and automatic face recognition.

Organization of this paper. In §2.2 we introduce notational conventions and adapt the notion of locality sensitive hashing to the ANS problem. We then generalize a well-known theoretical framework claiming that a locality sensitive hashing family gives rise to a search algorithm with sub-linear time. In §2.3, we propose a concrete hashing family for the Grassmanian manifold $\mathrm{G}(D, d)$ and for the union $\mathrm{G}(D, 1) \cup \mathrm{G}(D, d)$. We then formulate the main theorems of this work detailing the quality of the basic sub-linear search procedure in each one of the three types of searches described above. The details of the ANS algorithm resulting from the locality sensitive hashing family we proposed are outlines in §2.4, whereas §2.5 compares our ANS algorithm method with the ANS algorithm of Basri et al. [5] on model problems in image repatching and automatic face recognition.

2.2 Problem Formulation and Preliminaries

The Grassmannian. Let $\mathrm{G}(D, d)$ denote the *Grassmanian manifold*, i.e., the space of all d-dimensional linear subspaces of $\mathbb{R}^D$. If $0 < d_1 \leq d_2 < D$, $L_1 \in \mathrm{G}(D, d_1)$ and $L_2 \in \mathrm{G}(D, d_2)$, the *principal angles* $\theta_1 \geq ... \geq \theta_{d_1}$ between L_1 and L_2 can be defined as follows [14]: Let $\mathbf{Q}_{L_1}$ and $\mathbf{Q}_{L_2}$ be matrices whose columns are orthonormal bases for L_1 and L_2, respectively. For $i = 1, \ldots, d_1$ let $\sigma_i(\boldsymbol{Q}_{L_1}^T \boldsymbol{Q}_{L_2})$ denote the i-th largest singular value of the matrix $\boldsymbol{Q}_{L_1}^T \boldsymbol{Q}_{L_2}$. The principal angles $\pi/2 \geq \theta_1 \geq \theta_2 \geq \cdots \geq \theta_d \geq 0$, are[3]

$$\theta_i = \arccos(\sigma_{d-i}(\boldsymbol{Q}_{L_1}^T \boldsymbol{Q}_{L_2})), \quad i = 1, \ldots, d_1. \tag{2.1}$$

Using these angles, the distance between L_1 and L_2 is

$$\mathrm{dist}_{\mathrm{G}}(L_1, L_2) = \left(\sum_{i=1}^{d_1} \theta_i^2 \right)^{1/2}. \tag{2.2}$$

[3] Here, we order the principal angles decreasingly, unlike the common arrangement [14] (§12.4.3).

The Grassmannian endowed with this distance is a Riemannian manifold. Notice that if $d_1 = 1$, L_1 is a line, and $\mathrm{dist}_{\mathrm{G}}(L_1, L_2)$ is simply the angle between the line L_1 and the subspace L_2.

Approximate Subspace Search. Motivated by the approximate nearest point search in [15], we define the approximate nearest subspace search problem as follows:

Definition 2.2.1. (R, c)**-approximate subspace search:** *Let X be a set of d-dim. subspaces in $\mathbb{R}^D$ and R, c, δ be positive numbers. A search algorithm is called* (R, c)-approximate subspace search *if it fulfills the following requirement. Given a query subspace L of dimension d, if there is an element L' in X s.t.* $\mathrm{dist}_{\mathrm{G}}(L, L') \leq \mathrm{R}$*, then, an element L'' in X with* $\mathrm{dist}_{\mathrm{G}}(L'', L) < cR$ *is returned with probability* $1 - \delta$*.*

For several applications, the most interesting query problem is the point-subspace query, where the query is a point in $\mathbb{R}^D$ and the database is a subset of $\mathrm{G}(D, d)$. By connecting points with the origin to obtain lines, the point-subspace query problem is reduced to a line-subspace query problem, but with a different metric. That is, instead of measuring the Euclidean distance of the query point to the subspace, we measure the equivalent distance, $\mathrm{dist}_{\mathrm{G}}$, between the line through the query point and the subspace.

The use of this equivalent distance results in a point-subspace query. Indeed, assume that the query point $\boldsymbol{x}_0$ has a principal angle θ_0 and Euclidean distance $||\boldsymbol{x}_0||_2 \sin\theta_0$ w.r.t. the nearest subspace. Our algorithm returns a subspace which has principal angle $c\theta_0$ and Euclidean distance $||\boldsymbol{x}_0||_2 \sin(c\theta_0) < c||\boldsymbol{x}_0||_2 \sin\theta_0$ with the query (the inequality is true for any $c > 1$). This means the solution of the line-subspace query problem is also a solution for the corresponding approximate point-subspace query problem.

Locality Sensitive Hashing. Following [3], we apply the notion of locality sensitive hashing (LSH) family to the subspace search situation. We generalize the definition of [3] for LSH as follows:

Definition 2.2.2. Locality sensitive hashing family for $(X$,Q,$F)$: *Let X be a database, Q be a query set and F be a mapping from $X \times Q$ to* $[0, \infty)$*, which aims to measure the nearness between query and database points. A family $\mathcal{H}$ of functions on $X \cup Q$ with a probability measure $\mathbb{P}$ is called (R, cR, p_1, p_2)-sensitive for $(X$,Q,$F)$ if for* *any* $L_1 \in X$,$L_2 \in Q$*:*

$$\begin{aligned} &\mathbb{P}[h \in \mathcal{H} | h(L_1) = h(L_2)] \geq p_1, \text{ if } F(L_1, L_2) \leq R; \\ &\mathbb{P}[h \in \mathcal{H} | h(L_1) = h(L_2)] \leq p_2, \text{ if } F(L_1, L_2) \geq cR. \end{aligned} \tag{2.3}$$

We require that $p_1 > p_2$ *in order for the corresponding algorithm to work.*

We are interested in two cases. The first case is when $X, Q \subset \mathrm{G}(D, d)$ and $F = \mathrm{dist}_{\mathrm{G}}$. This corresponds to the approximate subspace-subspace query problem. In this case, the definition of LSH family in [3] coincides with Definition 2.2.2. The second case is when $X \subset \mathrm{G}(D, d)$, $Q \subset \mathrm{G}(D, 1)$ and $F = \mathrm{dist}_{\mathrm{G}}$. This corresponds to the approximate point-subspace (equivalently line-subspace) search problem.

The following theorem states that using the general LSH family of Definition 2.2.2, we can easily construct a corresponding locality hashing algorithm. Thus our main issue is to form an LSH family. This theorem is an immediate generalization of a theorem in [3, page 17]. Its proof is the same while replacing the neighborhood $B(q, r)$ of a query q with the set $\{x \in X | F(x, q) < r\}$.

Theorem 2.2.3. *Let* X *be a database,* Q *be a query set,* F *a mapping from* $X \times Q$ *to* $[0, \infty)$ *and denote by* n *the size of* X*. If there is a* (R, cR, p_1, p_2)*-sensitive family* $\mathcal{H}$ *for* (X, Q, F)*, where* $p_1, p_2 \in (0, 1)$*, then one can randomly draw from* $\mathcal{H}$ *to form a set* $\mathcal{G}$ *of vector-valued hash functions from* X *to* $\{0, 1\}^{\lceil \log_{1/p_2} n \rceil}$ *such that for* $\rho = \log(p_1)/\log(p_2)$*:*

- *For any query point in* Q*, the corresponding basic hashing procedure with* $\mathcal{G}$ *requires at most* $O(n^\rho/p_1)$ *evaluations of the hash functions from* $\mathcal{G}$*.*
- *The number of elements in* $\mathcal{G}$ *is at most* $O(n^\rho/p_1)$*. Thus evaluating at* n *points requires storage of order* $O(n^{1+\rho}/p_1)$*. The total storage is the sum of this storage and the storage of the original data.*

The failure probability δ *of the data structure is at most* $1/3 + 1/e$ *(e is Euler's number).*

2.3 Hashing Linear Subspaces

In this section, we describe a general hashing scheme that applies to approximate nearest subspace search problems in which the database consists of d_2-dimensional subspaces, and the query is a d_1-dimensional subspace. We claim (and prove in the supplementary appendix) that this scheme gives a locality sensitive hashing family for two cases of practical importance: $d_1 = d_2$ (subspace-subspace query) and $d_1 = 1, d_2 > 1$ (line-subspace query).

We generate the hashing scheme simply by thresholding the angle between the subspace L and a randomly generated line $\ell \in \mathrm{G}(D,1)$:

Definition 2.3.1. *Let $Q = \mathrm{G}(D,d_1)$ and $X = \mathrm{G}(D,d_2)$. For each line $\ell \in \mathrm{G}(D,1)$ and $0 < \theta_0 < \pi/6$, we associate a function $h_{l,\theta_0} : X \cup Q \longrightarrow \{0,1\}$, via*

$$h_{l,\theta_0}(L) = \begin{cases} 0, & \mathrm{dist}_{\mathrm{G}}(l,L) > \theta_0, \\ 1, & \mathrm{dist}_{\mathrm{G}}(l,L) \leq \theta_0. \end{cases} \tag{2.4}$$

Let $\mathcal{H}_{\theta_0}(d_1,d_2,D)$ denote the set of such functions h_{l,θ_0}, with the uniform measure on $\mathrm{G}(D,1)$. Also denote $\mathcal{H}_{\theta_0}(d,D) = \mathcal{H}_{\theta_0}(d,d,D)$.

Main Properties. For line-line query, this construction gives an LSH family with very good properties:

Theorem 2.3.2. *For any D, and any fixed $0 < \theta_0 < \pi/6$ and $R > 0$, $c > 1$, there exist $p_2 < p_1$ such that the hashing family $\mathcal{H}_{\theta_0}(D,1)$ is a $(R,cR,p_1,p_2)-$locality sensitive hashing family over $(\mathrm{G}(D,1)$,$\mathrm{G}(D,1), \mathrm{dist}_{\mathrm{G}})$.*

In practical applications such as recognition, it is valuable to allow the database to consist of subspaces (say, one subspace per subject). Our construction also yields sublinear-time algorithms for the important case of point-subspace (or equivalently line-subspace) query:

Theorem 2.3.3. *For any D, $d \leq D$, $c > 1$ and $0 < R < \pi/6$, there exist fixed positive real numbers $\theta_0 < \pi/6$ and $0 < p_2 < p_1 < 1$, such that $\mathcal{H}_{\theta_0}(D,d,1)$ is a (R,cR,p_1,p_2) locality sensitive family on $(\mathrm{G}(D,d), \mathrm{G}(D,1), \mathrm{dist}_{\mathrm{G}})$.*

Finally, our construction extends to query subspaces of higher dimensions, i.e., $Q = X = \mathrm{G}(D,d)$, with one caveat: we require R to be small ($R < R_0 \ll 1$), and c to be large ($c > \sqrt{d}$):

Theorem 2.3.4. *For any fixed* $0 < \theta_0 < \pi/6$, *and* $c > \sqrt{d}$, *there are positive real numbers* $0 < p_2 < p_1 < 1$ *depending on* c *and* $R \ll 1$, *such that* $\mathcal{H}_{\theta_0}(D, d)$ *with the induced uniform measure on it is* $(R, cR, p_1, p_2)-$*locality sensitive hashing family over* $(\mathrm{G}(D, d)$,$\mathrm{G}(D, d)$,$\mathrm{dist}_{\mathrm{G}})$.

Algorithmic implications. The sub-linear time in our theoretical guarantees depends on the exponent $\rho = \log(p_1)/\log(p_2)$. In general, ρ depends on the parameters D, d, c, R and θ_0. The choice of R depends on the distribution of the query points within the database and the estimate of ρ improves as R decreases. Ideally, each query element needs to be within distance R to the database. In many practical cases, the query points are contained or sufficiently close to the database and R can be sufficiently small. The "precision" parameter c is chosen according to practical needs (in the case where the query elements are contained in the database it can be arbitrarily large). To make p_2 as small as possible, θ_0 should be chosen to be $\pi/6$ (that is , maximal) and empirical experiments support this choice.

In two situations, we can assert the asymptotic behavior of the exponent ρ. The first case is when both Q and X are subsets of $\mathrm{G}(D, 1)$, D approaches infinity and the query elements are sufficiently close to the database.

Theorem 2.3.5. *If* $Q = X = \mathrm{G}(D, 1)$, $R = \alpha/\sqrt{D}$ $(\alpha > 0)$, $cR = O(1)$ *and* $0 < \theta_0 < \pi/6$ *is fixed, then*

$$\lim_{D\to\infty} \rho(D, d, c, R, \theta_0) \leq 1/(1 + e^{-\alpha^2/2}). \tag{2.5}$$

We note that the line-line query translates to point-point query (that is, nearest-neighbor search), where the points are on the sphere. For the more general cases of line-subspace and subspace-subspace queries, we can show that there exists a decreasing function $f(D) > 0$ such that if $R = f(D)$ and $cR = O(1)$, then $\lim_{D\to\infty} \rho(D, d, c, R, \theta_0) \leq C(d) < 1$. We cannot write an explicit expression of $f(D)$ (see supplementary appendix).

The second case is when $Q \subset X \subset \mathrm{G}(D, d)$ and n approaches infinity. Here ρ can be arbitrarily small as follows.

Theorem 2.3.6. *Assume that* $Q \subset X \subset \mathrm{G}(D, d)$. *For any* $\rho > 0$, *there is a locality sensitive hashing scheme to retrieve points from* X, *whose query time is at most* $O(Dn^\rho)$ *as* n *approaches infinity.*

Theorem 2.3.6 follows from two observations. The first one is that since every query is in the database, we can pick R to be very small and c to be large while keeping cR small. The second observation is that since c is large, the ratio between the logarithms of p_1 and p_2 can be made sufficiently small.

2.4 Algorithm details

Our algorithm exploits standard techniques from locality sensitive hashing to convert the hashing scheme described in the previous section into an efficient algorithm for near-subspace search. In offline preprocessing, we generate a hash table and assign database points to it. This process is described as Algorithm 1. At query time, we are given a new input q. We consider as possible candidates the first N subspaces which hash to the same bins as q, and perform an exhaustive search within this set. This procedure is described in detail in Algorithm 2.

Algorithm 1 Preprocessing

```
Input: Two integers $S \geq 0$, $K \geq 0$, a real number $0 < \theta_0 \leq \pi/6$ and a database $X$ of
  subspaces from $\mathrm{G}(D, d)$.
Output: Hash functions $g_j : X \to \{0,1\}^K$, $1 \leq j \leq S$ and Keys $\{\mathrm{Key}_{jk}^L\}_{1\leq j\leq S, 1\leq k\leq K}^{L\in X}$.
  Steps:
  for $1 \leq j \leq S$ do
    for $1 \leq k \leq K$ do
      • Randomly choose a point $\boldsymbol{x}_{jk}$ from $\mathbb{S}^{D-1}$ ($D-1$ dimensional sphere) according
      to the uniform measure.
      • $l_{jk} :=$ the line connecting $\boldsymbol{x}_{jk}$ and the origin.
    end for
    • Setting $g_j = (h_{l_{j1},\theta_0}, ..., h_{l_{jK},\theta_0})$.
    for $L \in X$ do
      • Let $(\mathrm{Key}_{j1}^L, ..., \mathrm{Key}_{jK}^L) = g_j(L)$.
    end for
  end for
  return $g_j$, $1 \leq j \leq S$ and $\{\mathrm{Key}_{jk}^L\}_{1\leq j\leq S, 1\leq k\leq K}^{L\in X}$.
```

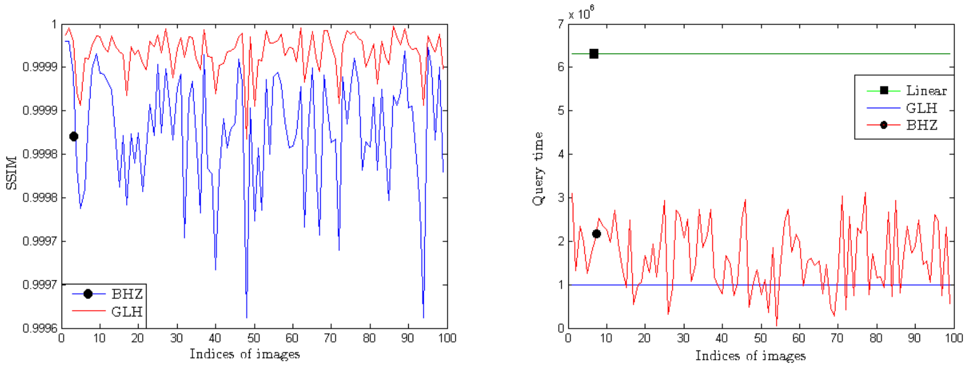
Given the hash table, Algorithm 2 searches for approximate nearest subspace in the following way:

Algorithm 2 Locality sensitive hashing for subspace search

```
Input: An integer $N \geq 0$, A query subspace $L \in \mathrm{G}(D,1) \cup \mathrm{G}(D,d)$, a hash family $\{g_j : X \to \{0,1\}^K\}_{1\leq j\leq S}$, a database $X$ of subspaces in $\mathrm{G}(D,d)$, and Keys $\{\mathrm{Key}^L_{jk}\}^{L\in X}_{1\leq j\leq S, 1\leq k\leq K}$.
Output: A $(\mathrm{R},\mathrm{c})$-approximate nearest subspace of $L$.
  Steps:
  • $\mathcal{L} = \varnothing$
  • Count=0
  for $1 \leq j \leq S$ do
    if Count $\leq N$ then
      • $\mathrm{Key}^L = g_j(L)$.
      • Search for $L' \in X$, s.t. $(\mathrm{Key}^{L'}_{j1}, ..., \mathrm{Key}^{L'}_{jK}) = \mathrm{Key}^L$.
      • If such $L'$ exists and $\mathrm{dist}_{\mathrm{G}}(L', L) \leq cR$, then Count=Count+1 and $\mathcal{L} = \mathcal{L} \cup \{L'\}$
    end if
  end for
  return the subspace in $\mathcal{L}$ closest to $L$ if exists.
```

2.5 Experiments

To evaluate the performance of our Grassmanian-based locality hashing (GLH) scheme in approximate subspace search, we carry out experiments on two model problems and compare the results with results by using the scheme of Basri, Hassner and Zelnik-Manor (BHZ) [5]. In the first problem, patch-based image reconstruction, the ambient dimension is relatively low. Both schemes perform much faster than exact search. While the accuracy of the GLH scheme and the BHZ scheme are comparable, the performance of GLH is more stable across different images. We will then consider a model face recognition experiment, in which the ambient dimension is relatively high. We will see that our GLH scheme obtains reliable results while BHZ scheme fails most of the time.



(a) Repatch error

(b) Running time

Figure 2.1: Mismatch error and number of evaluations: 100 test images from the Berkeley Segmentation Database are used in this experiment.

2.5.1 Image approximation

We follow Basri et al. [5] and try to reconstruct images using a dictionary of subspaces constructed from an arbitrarily chosen image. We use the Berkeley segmentation database [16], which contains 100 test images of size 481 × 321.

We randomly pick one image from this database and also randomly select 1000 pixels from it. Then, 16 different, overlapping 5×5 patches around each pixel are used to produce a $k = 4$ dimensional subspace by taking principal components. This produces a database of 1,000 subspaces. Each of the 100 images is subdivided into nonoverlapping 5×5 patches. For each patch, we search for the closest subspace in the database by using both the locality sensitive hashing scheme and the Basri et al. [5] BHZ scheme. We take the patch in the selected subspace which is closest to the query patch as its approximation.

To measure the quality of reconstruction, we use SSIM [17], which effectively detects the distortion of an image from another image. SSIM equal to 1 means that two images are identical.

Figure 2.1a shows the SSIM errors between original image and its repatched image. Upper (Red) line without dot is the SSIM errors for GLH, and lower (blue) line with dot is that for the BHZ scheme. Figure 2.1a shows that our algorithm performs better

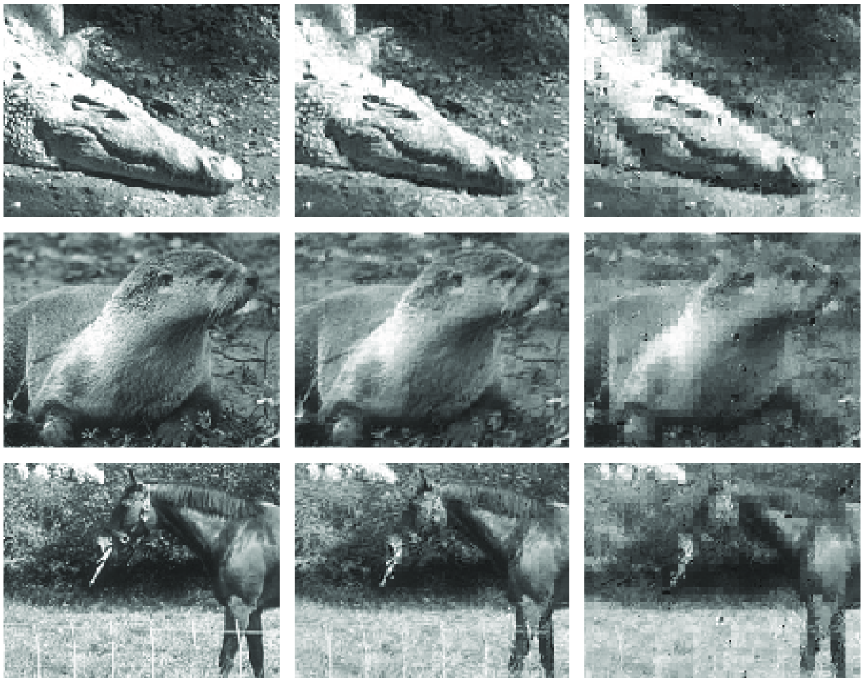


Figure 2.2: Repatch image: three original images are in the first column. images repatched by GLH scheme are in the second column. images repatched by BHZ scheme are in the third column.

than BHZ [5] on all images. Figure 2.1b is the number of evaluation needed to repatch image for each algorithms. Lower flat (blue) line is for GLH scheme, lower volatile (red) line with dot is for BHZ scheme, and upper flat (green) line with box is the linear exact search. Figure 2.1b shows that our algorithm is often faster and more importantly, has significantly less variability in its speed. In Figure 2.2, 3 images are repatched by GLH scheme and BHZ scheme. The first column is original, the second column is repatched by GLH scheme and the third column is repatched by BHZ scheme. Our algorithm is able to obtain better near neighbors, and hence much better visual quality.

2.5.2 Face recognition using the Multi-Pie and Cropped Yale databases

Images of faces with fixed pose under different illumination conditions lie near linear subspaces of dimension 9 (Epstein et al. [18], Ho et al. [19], Basri and Jacobs [11]). Therefore a database of faces can be easily transcribed into a set of subspaces (where each subspace represents a face). If a query is a single image of face, then the problem is to recognize the closest face (subspace) to the given image (point). Alternatively, the query can include several images of the same face under different illumination conditions. In this case, the query is a subspace.

We first used cropped images of the Multi-Pie database (see [20]) with frontview. That is, we used all subfolders of the form 05_0 for all persons of all four sessions of the multiview folder. There were a total of 239 persons and 80 frontview images for each. We cropped these well-aligned images by restricting the set of pixels and then used 23×19 cropped images. We remark that uncropped face images can be easily distinguished by clothing items and thus the recognition problem is easier. The subspaces of different faces (under different illumination conditions) are often close to each other (Epstein et al. [18], Ho et al. [19], Basri and Jacobs [11]). Therefore 9-dimensional (using the dimension upper bound 9 [11]) subspaces are hard to distinguish. We have experimentally found out that the subspaces are approximately of dimension 5.

For each person, we randomly picked 36 23×19 cropped images out of the 80 frontview images, vectorize them to lie in 437 dimensions and recorded their total least squares 5-dimensional subspace (spanned by the top 5 principal components). We created 239 such subspaces (one for each person). For each $d = 1, ..., 10$, we created 239 query subspaces as the span of d randomly picked images from the rest of the 44 images (the ones not used to create the database). The results of the GLH scheme, compared to BHZ scheme, are summarized in Figure 2.3, where for each query dimension d, the darker bar represents the success rate (among 239 query d-dimensional subspaces) of GLH and the brighter bar represents rate of success of the BHZ. GLH performs significantly better for $d \geq 4$. We note that the GLH takes place in 437 dimensions, while BHZ takes place in 95703 dimensions (437*(437+1)/2=95703).

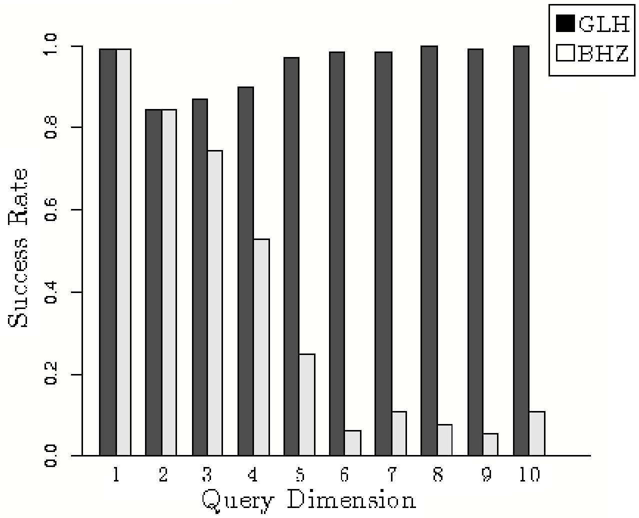


Figure 2.3: Success Rate for different Query dimension

We performed a similar experiment with the Cropped Yale faces database ([21, 22]). In this database, there are 38 persons, where for each person there are 64 different 24×21 images under different illuminations. The database of 38 5-dimensional subspaces is created by randomly choosing 36 images out of the 64 images per person vectorizing them to lie in 504 dimensions and computing their 5-dimensional total least squares subspace. Similarly to the previous experiment, for $d = 1, ..., 10$ we form the query d-dimensional subspaces by the span of randomly chosen d vectors from the other 28 images. Figure 2.4 report the success rate (among 38 queries) of GLH and BHZ for $1 \leq d \leq 10$. GLH performs significantly better than BHZ across all dimensions d. The GLH takes place in 504 dimension, whereas BHZ in 127,260 dimensions (504*505/2=127,260).

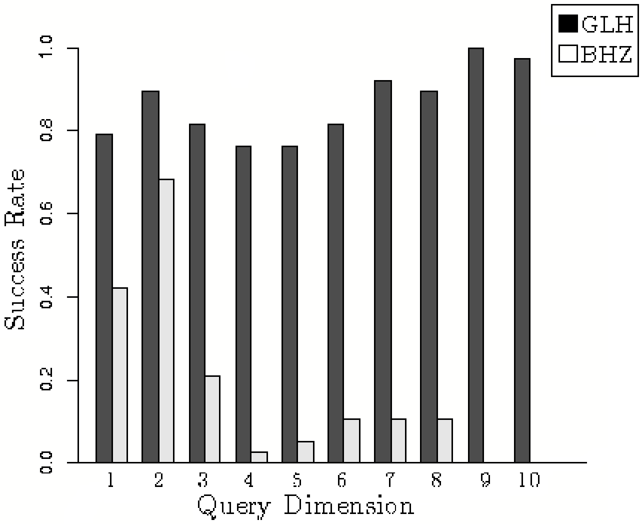


Figure 2.4: Success Rate for different Query dimension

2.6 Conclusion

We have proposed GLH, a sublinear time algorithm for the approximate point-to-subspace query and subspace-to-subspace query problems. It is based on a new locality sensitive hashing family, which takes advantage of the geometric position of different subspaces as encoded in their principal angles with random lines. The GLH algorithm performs stably and reliably in numerical experiments, and in particular outperforms previous approaches to approximate nearest subspace.

Although this method provides good results in our experiments, there is still room for improvement. First, it is desirable to extend the method to also handle affine subspaces. This would require hashing families that encode not only angles but also distances and maintain locality sensitivity hashing. Secondly, as with other algorithms for this problem, in extremely high ambient dimension, GLH shows the greatest advantage over linear scan when the database is very large. Therefore, it is desirable to find hashing families that work well even when the database is small.

2.7 Alternative Formulation: Hashing with points

An alternative formulation. The probabilities p_1 and p_2 are two important parameters. The algorithm performs better if the gap between them is large. In this section, we propose a modified hashing family for which we can write an analytic expression for p_1 and p_2.

Definition 2.7.1. *For each point $\boldsymbol{x} \in \mathbb{R}^D$ and $\eta \in \mathbb{R}^+$, we associate a function $h_{\boldsymbol{x},\eta}$ on* $\mathrm{G}(D,d)$*:*

$$h_{\boldsymbol{x},\eta} : \mathrm{G}(D,d) \longrightarrow \{0,1\}$$

such that for any $L \in \mathrm{G}(D,d)$

$$\begin{aligned} h_{\boldsymbol{x},\eta}(L) &= 0, && \textit{if} \quad \mathrm{dist}_{\mathrm{G}}(\boldsymbol{x},L) > \eta; \\ h_{\boldsymbol{x},\eta}(L) &= 1, && \textit{if} \quad \mathrm{dist}_{\mathrm{G}}(\boldsymbol{x},L) \leq \eta, \end{aligned} \tag{2.6}$$

where $\mathrm{dist}(\boldsymbol{x},L)$ *is the Euclidean distance between $\boldsymbol{x}$ and L.*

Let $\mathcal{H}$ be this family of functions. Let μ be the normal distribution on $\mathbb{R}^D$ with mean $\mathbf{0}$ and variance $\mathbf{1}$ in each direction. Its density function is $e^{-||\boldsymbol{x}||_2^2/2}/(2\pi)^{D/2}$. By identifying $\mathcal{H}$ with R^D, we get a measure μ on this hashing family. Following similar arguments, it is easy to see that $(\mathcal{H},\mu)$ is *locality sensitive hashing* family on $\mathrm{G}(D,d)$.

The maximal value of $\mu(\boldsymbol{x} \in \mathbb{R}^D | h_{\boldsymbol{x}}(L_1) = h_{\boldsymbol{x}}(L_2),\ \mathrm{dist}_{\mathrm{G}}(L_1,L_2) = R)$ as a function of $L_1, L_2 \in \mathrm{G}(D,d)$ is achieved when $\theta_1(L_1,L_2) = R$ and $\theta_i(L_1,L_2) = 0$ for $i = 2,...,d$. The minimal value of $\mu(\boldsymbol{x} \in \mathbb{R}^D | h_{\boldsymbol{x}}(L_1) = h_{\boldsymbol{x}}(L_2),\ \mathrm{dist}_{\mathrm{G}}(L_1,L_2) = R)$ is achieved when $\theta_i(L_1,L_2) = R/\sqrt{d}$ for $i = 1,...,d$.
Therefore,

$$\begin{aligned} p_1 = &\min_{\mathrm{dist}_{\mathrm{G}}(L_1,L_2)\leq R} \mu(\boldsymbol{x} \in \mathbb{R}^D | h_{\boldsymbol{x}}(L_1) = h_{\boldsymbol{x}}(L_2)) \\ &= 1 - 2\mu(\boldsymbol{x} \in \mathbb{R}^D | h_{\boldsymbol{x}}(L_1) = 1, \theta_i(L_1,L_2) = R/\sqrt{d}, \\ &\quad \forall 1 \leq i \leq d) + 2\mu(\boldsymbol{x} \in \mathbb{R}^D | h_{\boldsymbol{x}}(L_1) = h_{\boldsymbol{x}}(L_2) = 1, \\ &\qquad \theta_i(L_1,L_2) = R/\sqrt{d}, \forall 1 \leq i \leq d) \end{aligned} \tag{2.7}$$

To compute the probability in the RHS of (2.8), we note that for the underlying Gaussian measure $\mathrm{dist}^2(\boldsymbol{x},L)$ has chi-squared distribution $\mathcal{X}^2_{D-d}$ with $D-d$ degree of freedom. Therefore, the first probability in the RHS of (2.8) is $F(\eta^2; D-d)$, that is the cdf function at η^2 of the $\mathcal{X}^2_{D-d}$. To compute the second probability, we distinguish between the projection of X onto the orthogonal complement of $L_1 \oplus L_2$, whose distance from L_1 and L_2 distributes like $\mathcal{X}^2_{D-2d}$ (we denote the pdf of this distribution by $f(t; D-2d)$) and the projection onto $L_1 \oplus L_2$. For elements in the latter projection, we assign coordinates $(x_1, y_1, ..., x_d, y_d)$ so that the projection onto L_1 is $(x_1, 0, x_2, 0, ...)$ and its distance from L_2 (obtained by dot product with the normals $\{(\sin\theta_i, -\cos\theta_i)\}_{i=1}^d$ of L_2 in $L_1 \oplus L_2$) is $(\sum_{i=1}^d (x_i \sin\theta_i - y_i \cos\theta_i)^2)^{1/2}$. Using this observation, we obtain that

$$\begin{aligned} p_1 &= 1 - 2F(\eta^2; D-d) \\ &\quad + 2\int_0^{\eta^2} f(t; D-2d)dt \int_{\sum_{i=1}^d y_i^2 \leq \eta^2 - t} \Pi_{i=1}^d \frac{1}{\sqrt{2\pi}} e^{-\frac{y_i^2}{2}} dy_i \\ &\quad \times \int_{\sum_{i=1}^d (x_i \sin\frac{R}{\sqrt{d}} - y_i \cos\frac{R}{\sqrt{d}})^2 \leq \eta^2 - t} \Pi_{i=1}^d \frac{1}{\sqrt{2\pi}} e^{-\frac{x_i^2}{2}} dx_i \end{aligned} \tag{2.8}$$

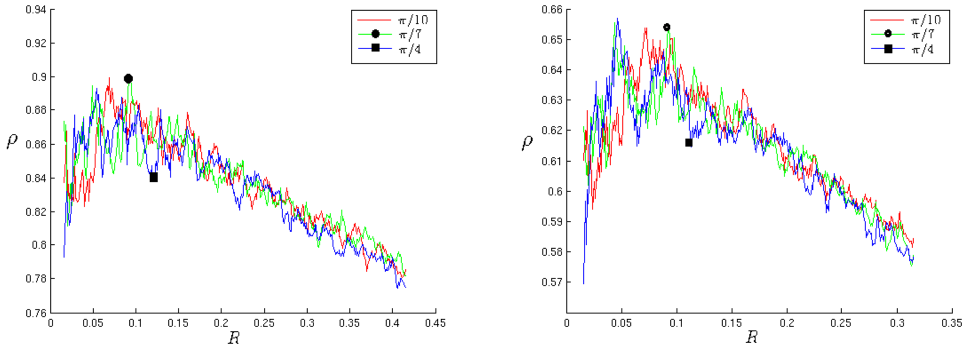
Similarly,

$$\begin{aligned} p_2 &= \max_{\mathrm{dist}_\mathrm{G}(L_1, L_2) \geq cR} \mu(\boldsymbol{x} \in \mathbb{R}^D | h_{\boldsymbol{x}}(L_1) = h_{\boldsymbol{x}}(L_2)) \\ &= 1 - 2F(\eta^2; D-d) + \frac{1}{\pi}\int_0^{\eta^2} f(t; D-d-1)dt \times \\ &\int_{y^2 \leq \eta^2 - t} e^{-\frac{y^2}{2}} dy \int_{(x\sin(cR) - y\cos(cR))^2 \leq \eta^2 - t} e^{-\frac{x^2}{2}} dx \end{aligned} \tag{2.9}$$

2.8 Numerical Investigation of parameters

The sublinearity exponent ρ of GLH algorithm (with N^ρ sublinear time) depends on the probability p_1 and p_2. It is desirable to know the dependence of p_1 and p_2 (or alternatively ρ) on the underlying parameters c, R, θ_0, D and d (also η if using alternative formulation). While it is hard to determine this in theory for the general case, we apply Monte-Carlo integration to estimate p_1 and p_2 in various instances and thus try to infer their dependence on the underlying parameters in these cases. In two paragraphs below, we consider the original formulation and the alternative formulation of GLH respectively.

Original GLH: Hashing with lines In this paragraph, the goal is to show how p_1 and p_2 (and ρ) depends on various parameters. By definition of p_1 and p_2, they



(a) (b)

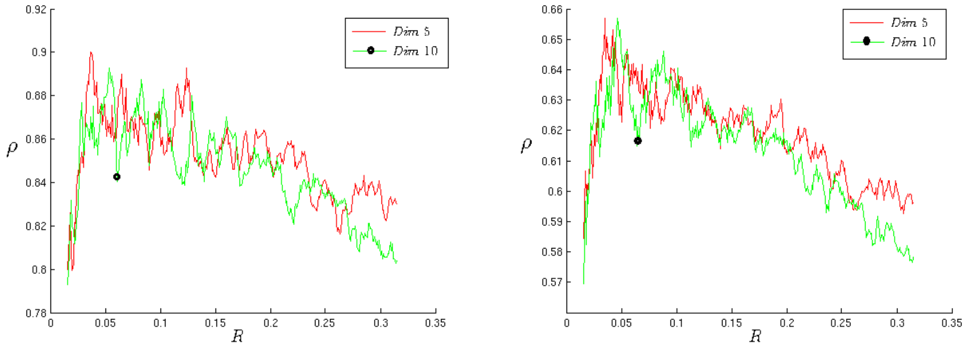
Figure 2.5: Comparing different sizes of neighborhoods

are volumns of particular areas on the sphere with uniform measure. Monte-Carlo integration is chosen to estimate these volumns. In the following experiments, we pick 100,000 random points uniformly from the sphere and check the percentages of points which are in the areas corresponding to p_1 and p_2. These percentages are taken as an estimation of p_1 and p_2.

Here, both the query points and the database are from $\mathrm{G}(10, 1)$. We demonstrate the dependence of ρ on R for some fixed values of c and θ_0, In Figure 2.5, ρ is plotted against R when $\theta_0 = \pi/10, \pi/7, \pi/4$ respectively, where Figure 2.5a $c = 1.1$ and Figure 2.5b $c = 1.5$. In this case, different values of θ_0 result in similar exponents.

Next, we demonstrate the dependence of ρ on D by observing both $D = 5$ and $D = 10$ and maintaining $d = 1$. We also fix $\theta_0 = \pi/4$. The results are shown in Figure 2.6 (in Figure 2.6a $c = 1.5$ and in Figure 2.6b $c = 1.1$). The plot without dot (red) is for $D = 5$ and the plot with dot (green) for $D = 10$. The sublinearity exponent ρ is fairly stable as the ambient dimension increases from 5 to 10.

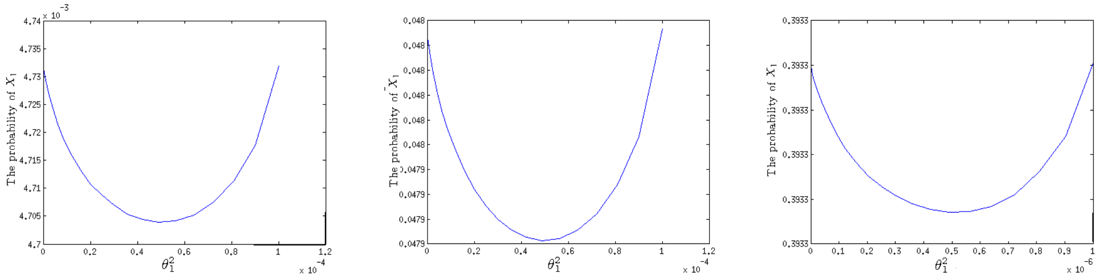
Alternative GLH: Hashing with points We shall use the alternative formulation of GLH defined in 2.7.1. The formulae to calculate p_1 and p_2 are given by 2.8 and 2.9. Denote $\mathrm{Pmin}_R = \min_{\mathrm{dist}_{\mathrm{G}}(L_1,L_2)=R} \mu(\boldsymbol{x} \in \mathbb{R}^D | h_{\boldsymbol{x}}(L_1) = h_{\boldsymbol{x}}(L_2))$ and $\mathrm{Pmax}_R = \max_{\mathrm{dist}_{\mathrm{G}}(L_1,L_2)=R} \mu(\boldsymbol{x} \in \mathbb{R}^D | h_{\boldsymbol{x}}(L_1) = h_{\boldsymbol{x}}(L_2))$.



(a)

(b)

Figure 2.6: Comparing different dimensions of ambient spaces



(a)

(b)

(c)

Figure 2.7: The probability depends on principal angles

In this paragraph, we demonstrate that the condition $c > \sqrt{d}$ in main theorems is necessary to establish LSH property. The observation is that there exists a probability spread (a gap between Pmax and Pmin for a given R) when subspaces are of dimension d larger than one. In other words, $\text{Pmin}_R \neq \text{Pmax}_R$. Because of this spread, to ensure $\text{Pmax}_R > p_1 = \text{Pmin}_R > p_2 = \text{Pmax}_{cR}$ means c can't be close to 1. The theoretical analysis leads to the condition $c > \sqrt{d}$. Numerical results below also support this. Since $\mu(h_{l,\theta_0} \in \mathcal{H} | h_{l,\theta_0}(L_1) = h_{l,\theta_0}(L_2))$ has a linear relation with $\mu(\mathcal{L}_\cap)$, we compute $\mu(\mathcal{L}_\cap)$ instead (In figure 2.7, $\mathcal{L}_\cap$ is shown as X_1).

In figure 2.7, we work with the space $\text{G}(4, 2)$. For each subfigure in 2.7, we fix $R = \sqrt{\theta_1^2 + \theta_2^2}$ (the distance) and η. Then, $\mu(X_1)$ (or $\mu(\mathcal{L}_\cap)$ for consistency) is computed

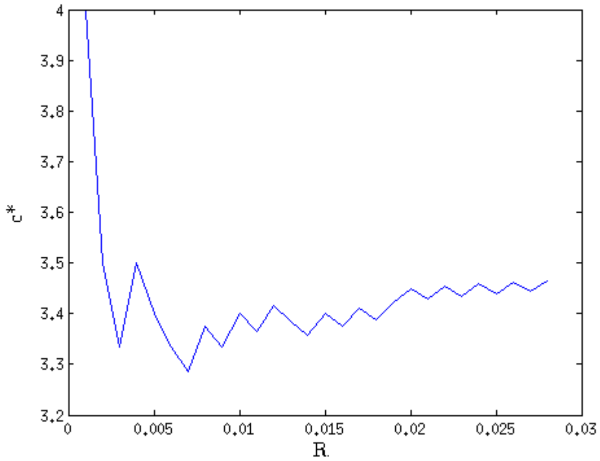


Figure 2.8: The constant c and the dimension d

for different pairs of principal angles (θ_1, θ_2) and plotted against θ_1^2. In figure 2.7a, $R^2 = 0.01$ and $\eta = 0.01$; In figure 2.7b, $R^2 = 0.01$ and $\eta = 0.1$; In figure 2.7c, $R^2 = 0.001$ and $\eta = 1$.

From figure 2.7, it is easy to see that the probability is minimized when both of principal angles are equal given distance R is fixed. This verifies the general theory that states the probability reaches its maximum Pmax_R if there is only one nonzero principal angle and reaches its minimum Pmin_R if all angles are equal.

Now we show how the parameter c is related to the dimension of subspaces. Particularly, we are interested in the minimal value of c that ensures $p_1 > p_2$ for a given R. In figure 2.8, the minimal value of c for each R is plotted against R in the case of $G(20, 10)$. The figures show that the lower bound of constant c is approximately $\sqrt{d}$ where $d = 10$ is the dimension of subspaces.

2.9 Proof of Main Theorems

Since each hash function in $\mathcal{H}_{\theta_0}(D, d_1, d_2)$ corresponds to a line in $\mathbb{R}^D$, we can identify $\mathcal{H}_{\theta_0}(D, d_1, d_2)$ with the unit sphere $\mathbb{S}^{D-1}$ and assign to it a probability measure which is induced by the uniform probability measure on $\mathbb{S}^{D-1}$. We denote throughout this section the measure by μ, that is, $\mathbb{P} := \mu$.

Proof of Theorem 3.3. We fix $0 < \theta_0 < \pi/6$. For $L_1 \in \mathrm{G}(D, d)$ and $L_2 \in \mathrm{G}(D, 1)$, $\mu(h_{l,\theta_0} \in \mathcal{H}_{\theta_0}(D, d, 1) | h_{l,\theta_0}(L_1) = h_{l,\theta_0}(L_2))$ depends only on the principal angle, which is the elevation angle $\theta(L_1, L_2)$ of the line L_2 with respect to the d-dimensional subspace L_1. We denote this probability by $g(\theta(L_1, L_2))$. To prove the theorem, we need only to show that $g(\theta)$ is a decreasing function of θ. Indeed, then $p_1 = g(\theta) > g(c\theta) = p_2$.

Let

$$\begin{aligned} &B_{\mathrm{G}(D,1)}(L, \theta_0) = \{l \in \mathrm{G}(D, 1) | \mathrm{dist}_{\mathrm{G}}(l, L) < \theta_0\}, \\ &\mathcal{L}_{\cap}(L_1, L_2) = \{l \in \mathrm{G}(D, 1) | h_{l,\theta_0}(L_1) = h_{l,\theta_0}(L_2) = 1\}, \\ &\text{and} \\ &\mathcal{L}_{\cup^c}(L_1, L_2) = \{l \in \mathrm{G}(D, 1) | h_{l,\theta_0}(L_1) = h_{l,\theta_0}(L_2) = 0\}. \end{aligned} \tag{2.10}$$

We note that,

$$\begin{aligned} &\mathcal{L}_{\cap}(L_1, L_2) = B_{\mathrm{G}(D,1)}(L_1, \theta_0) \cap B_{\mathrm{G}(D,1)}(L_2, \theta_0), \\ &\mathcal{L}_{\cup^c}(L_1, L_2) = (B_{\mathrm{G}(D,1)}(L_1, \theta_0) \cup B_{\mathrm{G}(D,1)}(L_2, \theta_0))^c, \end{aligned} \tag{2.11}$$

and

$$\begin{aligned} g(\theta(L_1, L_2)) = &1 - \mu(B_{\mathrm{G}(D,1)}(L_1, \theta_0)) \\ &- \mu(B_{\mathrm{G}(D,1)}(L_2, \theta_0)) + 2\mu(\mathcal{L}_{\cap}(L_1, L_2)). \end{aligned} \tag{2.12}$$

Since $\mu(B_{\mathrm{G}(D,1)}(L_1, \theta_0))$ and $\mu(B_{\mathrm{G}(D,1)}(L_2, \theta_0))$ in (2.12) are independent of $\theta(L_1, L_2)$, it is enough to show that $\mu(\mathcal{L}_{\cap}(L_1, L_2))$ decrease as $\theta(L_1, L_2)$ increases.

Let L_1 be a d-dimensional subspace in $\mathbb{R}^D$ and L_2, L_3 be two lines in $\mathbb{R}^D$ such that $\theta(L_1, L_2) = \alpha$ and $\theta(L_1, L_3) = \alpha + \beta$ ($0 < \beta, \alpha$ and $\alpha + \beta < \pi/6$ and $\alpha + \theta_0 < \pi/4$). Let $\{e_i\}_{i=1}^D$ be a basis of $\mathbb{R}^D$ such that

$$\begin{aligned} L_1 &= \mathrm{span}\{e_1, ..., e_d\}, \\ L_2 &= \mathrm{span}\{\cos\alpha e_1 + \sin\alpha e_{d+1}\}, \end{aligned}$$

We may rotate L_3 in a direction orthogonal to L_1 and maintain the elevation angle $\theta(L_1, L_3)$ so that L_3 is modified as follows

$$L_3 = \mathrm{span}\{\cos(\alpha + \beta)e_1 + \sin(\alpha + \beta)e_{d+1}\}.$$

Throughout the rest of the proof we express coordinates and operators w.r.t. the basis $\{e_i\}_{i=1}^D$.

Let A be the rotation of R^D which rotates L_2 to L_3 within the subspace $\mathrm{span}\{e_1, e_{d+1}\}$. We denote the image of a line l under the rotation A by $A(l)$ and note that $A(L_2) = L_3$.

Let l be the line passing through the point $(a_1, ..., a_D) \in \mathbb{S}^{D-1}$ and such that $l \in (B_{\mathrm{G}(D,1)}(L_1, \theta_0))^c \cap B_{\mathrm{G}(D,1)}(L_2, \theta_0)$. Since $l \in (B_{\mathrm{G}(D,1)}(L_1, \theta_0))^c$ and $\alpha + \theta_0 < \pi/4$

$$\sum_{i=d+1}^{D} a_i^2 > \sin\theta_0 \quad \text{and} \quad a_1 > a_{d+1}. \tag{2.13}$$

The image $A(l)$ is the line passing through $(a_1 \cos\beta - a_{d+1} \sin\beta, a_2, ..., a_d, a_1 \sin\beta + a_{d+1} \cos\beta, ..., a_D)$. The elevation angle $\theta(L_1, A(l))$ of $A(l)$ with respect to L_1 is

$$\begin{aligned}&\sin^{-1}(\sqrt{(a_1 \sin\beta + a_{d+1} \cos\beta)^2 + a_{d+2}^2 ... + a_D^2})\\ &> \sin^{-1}(\sqrt{a_{d+1}^2 + ... + a_D^2}) > \theta_0.\end{aligned}$$

. Therefore, $A(l) \in (B_{\mathrm{G}(D,1)}(L_1, \theta_0))^c \cap B_{\mathrm{G}(D,1)}(L_3, \theta_0)$. That is, $A((B_{\mathrm{G}(D,1)}(L_1, \theta_0))^c \cap B_{\mathrm{G}(D,1)}(L_2, \theta_0)) \subset (B_{\mathrm{G}(D,1)}(L_1, \theta_0))^c \cap B_{\mathrm{G}(D,1)}(L_3, \theta_0)$. Consequently,

$$\begin{aligned}&\mu((B_{\mathrm{G}(D,1)}(L_1, \theta_0))^c \cap B_{\mathrm{G}(D,1)}(L_2, \theta_0))\\ &= \mu(A((B_{\mathrm{G}(D,1)}(L_1, \theta_0))^c \cap B_{\mathrm{G}(D,1)}(L_2, \theta_0)))\\ &\leq \mu((B_{\mathrm{G}(D,1)}(L_1, \theta_0))^c \cap B_{\mathrm{G}(D,1)}(L_3, \theta_0)).\end{aligned} \tag{2.14}$$

In view of (2.11), we can rewrite (2.14) as

$$\begin{aligned}&\mu(B_{\mathrm{G}(D,1)}(L_2, \theta_0)/\mathcal{L}_{\cap}(L_1, L_2))\\ &\qquad\qquad \leq \mu(B_{\mathrm{G}(D,1)}(L_3, \theta_0)/\mathcal{L}_{\cap}(L_1, L_3))\end{aligned} \tag{2.15}$$

. Since $\mu(B_{\mathrm{G}(D,1)}(L_2, \theta_0)) = \mu(B_{\mathrm{G}(D,1)}(L_3, \theta_0))$, (2.15) implies that

$$\mu(\mathcal{L}_{\cap}(L_1, L_2)) \geq \mu(\mathcal{L}_{\cap}(L_1, L_3))$$

. That is, $\mu(\mathcal{L}_{\cap}(L_1, l))$ is a decreasing function of $\theta(L_1, l)$ for any l satisfying $\theta(L_1, l) < \min\{\pi/6, \pi/4 - \theta_0\}$. Combining this observation with 2.12 we conclude that $g(\theta(L_1, L_2))$ is a decreasing function of $\theta(L_1, L_2)$ and thus conclude the proof.

□

Proof of Theorem 3.4. We use similar notations as in the proof of Theorem 3.3. We fix $0 < \theta_0 < \pi/6$. For $L_1, L_2 \in \mathrm{G}(D, d)$, the probability

$$\mu(h_{l,\theta_0} \in \mathcal{H}_{\theta_0}(D, d) | h_{l,\theta_0}(L_1) = h_{l,\theta_0}(L_2))$$

depends only on the principal angles between L_1 and L_2. Indeed, this probability equals the RHS of (2.12) (here, L_1, $L_2 \in \mathrm{G}(D, d)$) and $\mu(\mathcal{L}_\cap(L_1, L_2))$ depends only on the relative position between the two subspaces. We denote this probability by $g(\theta_1, ..., \theta_d)$ where $\{\theta_i\}_{i=1}^d$ are the principal angles.

It is obvious that if $L_1 = L_2$, then the corresponding probability is $g(0, ..., 0) = 1$ and it obtains the maximal value among all principal angles $(\theta_1, ..., \theta_d)$. We will show that the directional derivatives of $g(\theta_1, ..., \theta_d)$ w.r.t. any direction at the origin is strictly negative. We use our estimates to obtain a lower bound on g in a ball of radius R around the origin when R is sufficiently small and an upper bound on g in the ball of radius cR and use these bounds to conclude that $p_1 > p_2$.

For convenience, we drop the requirements that $\theta_1 \geq ... \geq \theta_d$, but only assume that $0 \leq \theta_1, ..., \theta_d \leq \pi/2$. More precisely, for any $\theta_1, ..., \theta_d \in [0, \pi/2]^d$ we can parametrize L_1 (in the right coordinate system) as $L_1 = \{(x_1, ..., x_d, 0, ..., 0) | x_i \in \mathbb{R}\}$ and then $L_2 = \{(x_1 \cos\theta_1, ..., x_d \cos\theta_d, x_1 \sin\theta_1, ..., x_d \sin\theta_d, 0, ..., 0) | x_i \in \mathbb{R}\}$. This $\theta_1, ..., \theta_d$ parametrize the relative position between L_1 and L_2 even though they don't satisfy $\theta_1 \geq ... \geq \theta_d$. We note that with this convention, $g(\theta_1, ..., \theta_d)$ is invariant to permutations of its arguments.

We will verify the following two lemmas. Lemma 2.9.1 asserts that the probability $g(\theta_1, ..., \theta_d)$ indeed decreases around the origin in the coordinate directions. Lemma 2.9.2 reestablishes the connection between directional derivatives and coordinate derivatives (to use the chain rule we verify continuity of the partial derivatives). Moreover, it shows that the ratio of change in the fastest descent direction (the direction where θ_i change at the same rate) and in the slowest descent direction (the coordinate direction) is bounded by a factor of $\sqrt{d}$.

Lemma 2.9.1. *For each i, the coordinate directional derivative $\frac{\partial g}{\partial \theta_i}|_{(\theta_1,...,\theta_d)=(0,...,0)}$ is negative.*

Lemma 2.9.2. *For $s = s_1 \frac{\partial}{\partial \theta_1} + ... + s_d \frac{\partial}{\partial \theta_d}$, $\sqrt{d} || \frac{\partial g}{\partial \theta_1} ||_2 \leq || \frac{\partial g}{\partial s} ||_2 \leq || \frac{\partial g}{\partial \theta_1} ||_2$.*

The proofs of these two lemmas are given in the Appendix. Now, we give the proof of Theorem 3.4.

Let $\frac{\partial g}{\partial \theta_i} = -a$ $(a > 0)$ and $S(R) = \{(\theta_1, ..., \theta_d) | \sum_{i=1}^d \theta_i^2 = R^2\}$. Applying Taylor expansion to $g(\theta_1, ..., \theta_d)$ at the origin and Lemmas 2.9.1 and 2.9.2, we obtain that

$$\begin{aligned} &\max_{S(cR)} g(\theta_1, ..., \theta_d) \leq g(0, ..., 0) - acR + O(c^2R^2), \\ \text{and } &\min_{S(R)} g(\theta_1, ..., \theta_d) \geq g(0, ..., 0) - \sqrt{d}aR + O(R^2). \end{aligned} \tag{2.16}$$

Therefore, if $c > \sqrt{d}$ and R is sufficiently small, then,

$$\min_{S(R)} g(\theta_1, ..., \theta_d) > \max_{S(cR)} g(\theta_1, ..., \theta_d).$$

If we choose $p_1 = \min\limits_{S(R)} g(\theta_1, ..., \theta_d)$ and $p_2 = \max\limits_{S(cR)} g(\theta_1, ..., \theta_d)$, then $\mathcal{H}_{\theta_0}(D, d)$ is an (R, c, p_1, p_2)-sensitive hashing family by the definition of g and the LSH family.

□

Proof of Theorem 3.5. Fixing $0 < \theta_0 < \pi/6$, the neighborhood $B_{\mathrm{G}(D,1)}(L, \theta_0) = \{l \in \mathrm{G}(D, 1) | \mathrm{dist}_{\mathrm{G}}(l, L) < \theta_0\}$ of a line L is a hyperspherical cap (on the unit sphere).

Let L_1, L_2 and L_2' be three lines in $\mathbb{R}^D$ with $\mathrm{dist}(L_1, L_2) = \theta$ and $\mathrm{dist}(L_1, L_2') = c\theta$ for some $c, \theta > 0$. Moreover, let

$$\begin{aligned} &X_1 = B_{\mathrm{G}(D,1)}(L_1, \theta_0) \backslash B_{\mathrm{G}(D,1)}(L_2, \theta_0), \\ &X_2 = B_{\mathrm{G}(D,1)}(L_1, \theta_0) \cap B_{\mathrm{G}(D,1)}(L_2, \theta_0), \\ \text{and } &X_3 = X_2 \backslash B_{\mathrm{G}(D,1)}(L_2', \theta_0). \end{aligned} \tag{2.17}$$

Using this notation, we formulate the following lemma, which we later prove in the appendix.

Lemma 2.9.3. *Assume that $R = \alpha/\sqrt{D}$ for a fixed $\alpha > 0$ and that $cR = O(1)$. The measures $x_1 := \mu(X_1)$ and $x_3 := \mu(X_3)$ satisfy the following properties: when $D \to \infty$, $x_1, x_3 \to 0$ and $\lim\limits_{D\to\infty} x_3/x_1 > e^{\alpha^2/2}$.*

We conclude Theorem 3.5 as follows.

We denote $p_1 = \mu(h_{l,\theta_0} | h_{l,\theta_0}(L_1) = h_{l,\theta_0}(L_2)) = 1 - 2x_1$ and $p_2 = \mu(h_{l,\theta_0} | h_{l,\theta_0}(L_1) =$

$h_{l,\theta_0}(L'_2)) = 1 - 2(x_1 + x_3)$.

$$\begin{aligned}\lim_{D\to\infty} \rho &= \lim_{D\to\infty} \ln(p_1)/\ln(p_2) \\ &= \lim_{D\to\infty} \ln(1-2x_1)/\ln(1-2x_1-2x_3) \\ &= \lim_{D\to\infty} x_1/(x_1+x_3) \qquad \text{(since } x_1, x_3 \to 0\text{)} \qquad (2.18) \\ &= \lim_{D\to\infty} 1/(1+x_3/x_1) \\ &< 1/(1+e^{\alpha^2/2}). \qquad \text{(by Lemma 2.9.3)}\end{aligned}$$

□

Proof of Theorem 3.6. The GLH algorithm returns a point that is within cR distance of the query if there is a point in the dataset that is within R distance of the query. When the query is in the dataset, it is guaranteed to have a point within R distance of the query for any $R > 0$. Therefore, we can pick R arbitrarily small, and we note that p_1 approaches 1 as R approaches zero ($\ln(p_1)$ approaches zero). Moreover, we can pick c such that $cR = O(1)$. This can keep p_2 to be a fixed constant less than 1. That is, if the query is in the dataset, we are able to adjust c and R such that $\rho = \ln(p_1)/\ln(p_2) = \epsilon$ for any $\epsilon > 0$. □

2.10 Local Behavior of $g(\theta_1, ..., \theta_d)$

Throughout this section, we use the following coordinate representation.

$$\begin{aligned}&\mathbb{R}^D: && (x_1, ..., x_d, y_1, ..., y_d, z_{2d+1}, ..., z_D), \\ &L_1: && \{(x_1, x_2, ..., x_d, 0, ..., 0) | x_i \in \mathbb{R}\}, \\ &\text{and} && \qquad (2.19) \\ &L_2: && \{(x_1 \cos\theta_1, ..., x_d \cos\theta_d, x_1 \sin\theta_1, ..., \\ &&& x_d \sin\theta_d, 0, ..., 0) | x_i \in \mathbb{R}, \theta_i > 0\}.\end{aligned}$$

Proof of Lemma 2.9.1 First, we study the derivative along the coordinate direction $\frac{\partial}{\partial\theta_1}$ at the origin. This is the case where $\theta_1 = \epsilon$ and $\theta_i = 0$ for $i \neq 1$. Suppose L_1 is given in (2.19) and L_2^ϵ is given by

$$\{(x_1\cos\epsilon, x_2, ..., x_d, x_1\sin\epsilon, 0, ..., 0)|x_i \in \mathbb{R}, \epsilon > 0\}.$$

Denote $g(\epsilon, 0..., 0) = g(\epsilon)$ for short.

Let $(a_1, ..., a_{2d-2}) \in \mathbb{R}^{2d-2}$. We define the following quantities:

$A(a_1, ..., a_{2d-2})$ is subset of S^{D-1} with coordinates:

$$(x_2, ..., x_d, y_2, ..., y_d) = (a_1, ..., a_{2d-2}),$$

$$h(\epsilon, a_1, ..., a_{2d-2}) = \mu(h_{l,\theta_0} \in \mathcal{H}_{\theta_0}(D, d)$$
$$|l \in A(a_1, ..., a_{2d-2}), h_{l,\theta_0}(L_1) \neq h_{l,\theta_0}(L_2^\epsilon)),$$

and

$\text{Vol}_d(r)$ is volume of $(d-1)$-dim. sphere of radius r.

Then, we can write $g(\epsilon)$ as an integral of $h(\epsilon, a_1, ..., a_{2d-2})$ as follows:

$$g(\epsilon) = 1 - \int_{(a_1,..,a_{2d-2}) \in D^{2d-2}} h(\epsilon, a_1, .., a_{2d-2}) d\nu. \tag{2.20}$$

We observe that $\frac{\partial h(\epsilon,0,...,0)}{\partial \epsilon}$ where $a_i = 0, \forall i$. Using polar coordinates (r, θ) on (x_1, y_1)-plane. $h(\epsilon, 0, ..., 0)$ can be written in this way:

$$\frac{2}{\text{Vol}_D(1)} \int_{-\epsilon/2}^{\theta_0} d\theta \int_{\frac{\cos\theta_0}{\cos\theta}}^{\frac{\cos\theta_0}{\cos(\theta+\epsilon)}} \text{Vol}_{D-2d}(\sqrt{1-r^2}) r dr.$$

Therefore, the derivative $\frac{\partial h(\epsilon,0,...,0)}{\partial \epsilon}|_{\epsilon=0}$ is equal to

$$\frac{2}{\text{Vol}_D(1)} \int_0^{\theta_0} \text{Vol}_{D-2d}(\sqrt{1 - \frac{\cos^2\theta_0}{\cos^2\theta}}) \frac{\cos^2\theta_0 \sin\theta}{\cos^3\theta} d\theta.$$

This is bigger than zero. Since $\frac{\partial h}{\partial \epsilon}|_{\epsilon=0}$ is continuous and non-negative on $(a_1, ..., a_{2d})$, and when $(a_1, ..., a_{2d}) = 0$, $\frac{\partial h}{\partial \epsilon}|_{\epsilon=0}$ is strictly positive. We conclude,

$$\frac{\partial g(\theta_1, .., \theta_d)}{\partial \theta_1}|_{(\theta_1,...,\theta_d)=0} = \frac{\partial g(\epsilon)}{\partial \theta_1}|_{\epsilon=0}$$
$$= -\int_{(a_1,..,a_{2d}) \in D^{2d-2}} \frac{\partial h(\epsilon, a_1, .., a_{2d})}{\partial \epsilon}|_{\epsilon=0} d\nu < 0.$$

By symmetry,

$$\frac{\partial g}{\partial \theta_i} = \frac{\partial g}{\partial \theta_1} = -a < 0$$

for some fixed a and all i. □

Proof of Lemma 2.9.2 By symmetry of the function $g(\theta_1, ..., \theta_d)$, we need show the usual chain rule $\frac{\partial g}{\partial s} = s_1 \frac{\partial g}{\partial \theta_1} + ... + s_d \frac{\partial g}{\partial \theta_d}$ holds for the region $\{\theta_i \geq 0, \forall i\}$. Firstly, we show that the partial derivatives of $g(\theta_1, ..., \theta_d)$ are continuous up to the origin in the region $\{\theta_i \geq 0, \forall i\}$. Then, the chain rule follows from this fact.

Notice that $g(\theta_1, ..., \theta_d) = 1 - 2\mu(B_{\mathrm{G}(D,1)}(L, \theta_0)) + 2\mu(\mathcal{L}_\cap(L_1, L_2))$. Since the first two terms are constants, it is enough to show that the derivatives of $\mu(\mathcal{L}_\cap(L_1, L_2))$ is continuous up to the origin in the region $s_i \geq 0$. We shall prove this for a general class of functions.

Firstly, we introduce a class of rotations.

Definition 2.10.1. *Given angles $\{\theta_i\}_{i=1}^d$ $(d > 0)$, a rotation $A(\theta_1, ..., \theta_d)$ on $\mathbb{R}^D$ is defined as follows.*

For a point $\boldsymbol{x} = (x_1, ..., x_D) \in \mathbb{R}^D$, the i-th coordinate of the image $A(\theta_1, ..., \theta_d)(\boldsymbol{x})$ is equal to

$$\begin{cases} x_i \cos\theta_i, + x_{d+i} \sin\theta_i, & 1 \leq i \leq d, \\ -x_{i-d} \sin\theta_{i-d} + x_i \cos\theta_{i-d}, & d+1 \leq i \leq 2d, \\ x_i, & i \geq 2d. \end{cases}$$

In other words, $A(\theta_1, ..., \theta_d)$ rotates the first $2d$ coordinates. For a set $X \subset \mathbb{R}^D$, we denote its image under this rotation by $A(\theta_1, ..., \theta_d)(X)$.

Then, we define two set of functions.

Definition 2.10.2. *Let $\mu_{\mathbb{S}^{D-1}}$ be the uniform measure on the unit sphere $\mathbb{S}^{D-1}$ and $\mu_{\mathbb{R}^D}$ be the Lebesgue measure on $\mathbb{R}^D$. Given two smooth-boundary regions U and V on $\mathbb{S}^{D-1}$, a function of $(\theta_1, ..., \theta_d)$ is defined by*

$$G_{UV}(\theta_1, ..., \theta_d) = \mu_{\mathbb{S}^{D-1}}(U \cap A(\theta_1, ..., \theta_d)(V))$$

Moreover, denote by $C[U]$ and $C[V]$ the cones generated by connecting U and V with the origin respectively. We define

$$CG_{UV}(\theta_1, ..., \theta_d) = \mu_{\mathbb{R}^D}(C[U] \cap A(\theta_1, ..., \theta_d)(C[V]))$$

In the following, a convex polytope cone is a convex cone with vertex at the origin such that sides are hyperplanes and the base is enclosed by the unit sphere. Denote

$\Theta = (\theta_1, ..., \theta_d)$ and $\boldsymbol{e}_i$ and $\boldsymbol{v}_i$ be the i-th coordinate direction of Θ and $\mathbb{R}^D$ respectively. Let X be a convex polytope cone with sides $\{F_i\}_{i=1}^S$. Let X_Θ be the cone with sides $A(\Theta)(F_1)$ and $\{F_i\}_{i=2}^S$.

Lemma 2.10.3. *$\mu_{\mathbb{R}^D}(X_\Theta)$ is contiuously differentiable w.r.t. $\{\frac{\partial}{\partial \theta_i}\}_{i=1}^d$ in $[0, \alpha_1] \times ... \times [0, \alpha_d]$ for some positive numbers $\alpha_1, ..., \alpha_d > 0$.*

Let n^Θ be the unit normal direction of $A(\Theta)(F_1)$ and α^Θ be the elevation angle between n^Θ and the subspace spanned by $\{\boldsymbol{v}_1, \boldsymbol{v}_{d+1}\}$. Let Ω^Θ be the region on $A(\Theta)(F_1)$ enclosed by the other sides $\{F_i\}_{i=2}^S$ and the base $\mathbb{S}^{D-1}$. Specifically, $\Omega^{\mathbf{0}} = X$ when Θ is the origin $\mathbf{0}$.

We show that $\frac{\partial \mu_{\mathbb{R}^D}(X_\Theta)}{\partial \theta_1}$ is continuous.

Let $\Delta\Theta = \epsilon \boldsymbol{e}_1$. The angle $\text{Ang}(\Theta, \Delta\Theta)$ between n^Θ and $n^{\Theta+\Delta\Theta}$ is $\cos^{-1}[\cos^2\alpha^\Theta \cos\epsilon + \sin^2\alpha^\Theta]$. Let $\text{ProjNorm}(\boldsymbol{x}, \Theta, \Delta\Theta)$ be the norm of the projection of a point $\boldsymbol{x} \in A(\Theta)(F_1)$ to the plane spanned by n^Θ and $n^{\Theta+\Delta\Theta}$. Following from direct computation, $\text{ProjNorm}(\boldsymbol{x}, \Theta, \Delta\Theta)$ is equal to

$$\frac{(\cos\epsilon - 1)[n_1^\Theta x_1 + n_{d+1}^\Theta x_{d+1}] + \sin\epsilon[n_{d+1}^\Theta x_1 - n_1^\Theta x_{d+1}]}{(\sin^2\alpha^\Theta \cos^2\alpha^\Theta (1-\cos\epsilon)^2 + \cos^2\alpha^\Theta \sin^2\epsilon)^{1/2}}$$

Then, we can express the partial derivative as follows.

$$\frac{\partial \mu_{\mathbb{R}^D}(X_\Theta)}{\partial \theta_1} = \lim_{\epsilon \to 0} \int_{r=0}^1 r^2 dr \int_{\boldsymbol{x} \in \Omega^\Theta} \text{ProjNorm}(\boldsymbol{x}, \Theta, \Delta\Theta) \times \text{Ang}(\Theta, \Delta\Theta) d\boldsymbol{x} / \epsilon.$$

By applying Tyler's expansion, we have

$$\frac{\partial \mu_{\mathbb{R}^D}(X_\Theta)}{\partial \theta_1} = \int_{\boldsymbol{x} \in \Omega^\Theta} \frac{2(n_{d+1}^\Theta x_1 - n_1^\Theta x_{d+1})}{3\cos\alpha^\Theta(\sin^2\alpha^\Theta \cos^2\alpha^\Theta + 2\cos^2\alpha^\Theta)} d\boldsymbol{x}$$

Since n^Θ and α^Θ are continous as Θ approaches the origin. Moreover, the domain Ω^Θ will approach $\Omega^{\mathbf{0}}$. Therefore,

$$\lim_{\Theta \to \mathbf{0}} \frac{\partial \mu_{\mathbb{R}^D}(X_\Theta)}{\partial \theta_1} = \frac{\partial \mu_{\mathbb{R}^D}(X_\Theta)}{\partial \theta_1}\Big|_{\Theta=\mathbf{0}}.$$

This means $\mu_{\mathbb{R}^D}(X_\Theta)$ is continuously differentiable. □

Lemma 2.10.4. *if $C[U]$ and $C[V]$ are two convex polytope cones, then there exists some positive numbers $\alpha_1, ..., \alpha_d > 0$ such that $CG_{UV}(\theta_1, ..., \theta_d)$ has continuous partial derivatives in $[0, \alpha_1] \times ... \times [0, \alpha_d]$.*

The intersection $C[U] \cap A(\theta_1, ..., \theta_d)(C[V])$ is also a convex polytope cone. Its sides from U are fixed and sides from $A(\theta_1, ..., \theta_d)(C[V])$ are moving as $(\theta_1, ..., \theta_d)$ change. We can decompose the rotation of its moving sides into individual rotations of each moving side. Following Lemma 2.10.3, The intersection has continuous partial derivatives if one side is moving. After combining individual rotations, we have partial derivatives of $CG_{UV}(\theta_1, ..., \theta_d)$ are continuous in $[0, \alpha_1] \times ... \times [0, \alpha_d]$. □

For general smooth-boundary region U on $\mathbb{S}^{D-1}$, we approximate $C[U]$ by unions of polytope cones. Let $\{\mathcal{P}_i = \{X_{ij}\}_{j=1}^{N_i}\}_{i=1}^{\infty}$ be a sequence of partitions of $\mathbb{S}^{D-1}$ satisfying:

- $\forall i$, $\cup_{j=1}^{N_i} X_{ij} = \mathbb{S}^{D-1}$
- $\forall i, j$, $C[X_{ij}]$ is a polytope cone.
- if $i < k$, each piece X_{ij} of $\mathcal{P}_i$ is a union of pieces in $\mathcal{P}_k$. That is, $\mathcal{P}_k$ is a refinement of $\mathcal{P}_i$.
- $\max\limits_{1 \le j \le N_i} diam(X_{ij}) \le 1/n$ for $i = n$, $\forall n$ ($diam(X_{ij})$ is the diameter of X_{ij})

For each n, let $U^n = \bigcup\limits_{X_{nj} \subset U;\, X_{nj} \in \mathcal{P}_n} X_{nj}$. Then we have an increasing sequence $U^1 \subset ... \subset U^n \subset ... \subset U$ and $\bigcup\limits_{n=1}^{\infty} U^n = U$. Notice that $C[U^n]$ can be expressed as a finite collection of polyhedra cones.

From now on, we fix the sequence of partitions $\{\mathcal{P}_i\}_{i=1}^{\infty}$ and denote $G_{U^n V^n}$ and $CG_{U^n V^n}$ by G_{UV}^n and CG_{UV}^n respectively.

Lemma 2.10.5. *Given U, V and $1 \le i \le d$, $\frac{\partial CG_{UV}}{\partial \theta_i}$ is continuous at the origin.*

By Lemma 2.10.4, $\frac{\partial CG_{UV}^n}{\partial \theta_i}$ is continuous on $[0, \alpha_1^n] \times ... \times [0, \alpha_d^n]$ for $1 \le n \le \infty$ where α_i^j is positive $\forall i, j$.

$$\begin{aligned} \frac{\partial CG_{UV}(\Theta)}{\partial \theta_i} &= \lim_{\epsilon \to 0} \frac{CG_{UV}(\Theta + \epsilon \boldsymbol{e}_i) - CG_{UV}(\Theta)}{\epsilon} \\ &= \lim_{\epsilon \to 0} \frac{CG_{UV}^n(\Theta + \epsilon \boldsymbol{e}_i) - CG_{UV}^n(\Theta) - c\epsilon/n^{D-2}}{\epsilon} \\ &= \frac{\partial CG_{UV}^n(\Theta)}{\partial \theta_i} - c/n^{D-2} \end{aligned} \tag{2.21}$$

In the second equality, c is a bounded constant. If $\frac{\partial CG_{UV}}{\partial \theta_i}$ is not continuous at the origin, then there exists $\epsilon > 0$ such that $\forall \delta > 0$, there exist points $|\boldsymbol{a} - \boldsymbol{b}| < \delta$ s.t. $|\frac{\partial CG_{UV}(b)}{\partial \theta_i} - \frac{\partial CG_{UV}(a)}{\partial \theta_i}| > \epsilon$.

On the other hand, we can pick $N > 0$ so that $c/N^{D-2} \leq \epsilon/2$. Moreover, we can pick $\delta > 0$ so that if $|\boldsymbol{a} - \boldsymbol{b}| < \delta$, then $|\frac{\partial CG^N_{UV}(b)}{\partial \theta_i} - \frac{\partial CG^N_{UV}(a)}{\partial \theta_i}| \leq \epsilon/2$ since $\frac{\partial CG^N_{UV}}{\partial \theta_i}$ is continuous. By 2.21, we have $|\frac{\partial CG_{UV}(b)}{\partial \theta_i} - \frac{\partial CG_{UV}(a)}{\partial \theta_i}| < \epsilon$ for $|\boldsymbol{a} - \boldsymbol{b}| < \delta$. This contradiction asserts that $\frac{\partial CG_{UV}}{\partial \theta_i}$ is continuous at the origin. □

Lemma 2.10.6. *if CG_{UV} has continuous partial derivatives, then G_{UV} also has continuous partial derivatives.*

We have the following relation:

$$CG_{UV}(\theta_1, ..., \theta_d) = \int_{r=0}^{1} r^{D-1} G_{UV}(\theta_1, ..., \theta_d) dr.$$

It is easy to see the lemma holds. □

By above lemmas, it is easy to see that G_{UV} has continuous partial derivatives. And $g(\theta_1, ..., \theta_d) = G_{UU}(\theta_1, ..., \theta_d)$ with U and $V = A(\theta_1, ..., \theta_d)(U)$ be neighborhoods of L_1 and $L_2 = A(\theta_1, ..., \theta_d)(L_1)$ respectively. Thus, $g(\theta_1, ..., \theta_d)$ has continuous partial derivatives and the chain rule holds for it.

Using the above result, we can now prove Lemma 2.9.2.

By chain rule, $\frac{\partial g}{\partial s} = |s_1| \frac{\partial g}{\partial \theta_1} + ... + |s_d| \frac{\partial g}{\partial \theta_d}$. Since $\frac{\partial g}{\partial \theta_i} = \frac{\partial g}{\partial \theta_j}$, $\frac{\partial g}{\partial s} = (|s_1| + ... + |s_d|) \frac{\partial g}{\partial \theta_1}$. Note $s_1^2 + ... + s_d^2 = 1$. By Cauchy-Schwarz inequality, $1 \leq |s_1| + ... + |s_d| \leq \sqrt{d}$. Thus, the inequality for directional derivatives follows. □

2.10.1 Hyperspherical Area As D Approaches Infinity

In this section, we use probability measure on $\mathbb{S}^{D-1}$ such that $\mu(\mathbb{S}^{D-1}) = 1$.

Lemma 2.10.7. *For any fixed $0 < \theta_1 < \theta_0 < \pi/6$, both $\mu(B_{\mathrm{G}(D,1)}(L, \theta_0))$ and the ratio of $\mu(B_{\mathrm{G}(D,1)}(L, \theta_1))/\mu(B_{\mathrm{G}(D,1)}(L, \theta_0))$ approaches zero, as the dimension D approaches infinity.*

The usual volumn of hyperspherical cap is $\pi^{(D-2)/2}/\Gamma(D/2)\int_0^\theta \sin^{D-1}(t)dt$. So,

$$\begin{aligned}
&\mu(B_{\mathrm{G}(D,1)}(L,\theta_0))\\
&\quad= \mathrm{Vol}(B_{\mathrm{G}(D,1)}(L,\theta_0))/\,\mathrm{Vol}(\mathbb{S}^{D-1})\\
&\quad= \int_0^{\theta_1} \sin^{D-1}(t)dt \Big/ (2\pi)\\
&\quad\to 0, \text{ as } D \text{ approaches infinity.}
\end{aligned} \tag{2.22}$$

$$\begin{aligned}
&\mu(B_{\mathrm{G}(D,1)}(L,\theta_1))/\mu(B_{\mathrm{G}(D,1)}(L,\theta_0))\\
&\quad= \mathrm{Vol}(B_{\mathrm{G}(D,1)}(L,\theta_1))/\,\mathrm{Vol}(B_{\mathrm{G}(D,1)}(L,\theta_0))\\
&\quad= \int_0^{\theta_1} \sin^{D-1}(t)dt \Big/ \int_0^{\theta_0} \sin^{D-1}(t)dt\\
&\quad< \int_0^{\theta_1} \sin^{D-1}(t)dt \Big/ (\sin^{D-1}(\frac{\theta_0+\theta_1}{2}) * \frac{\theta_0-\theta_1}{2})\\
&\quad< \frac{2}{\theta_0-\theta_1}\int_0^{\theta_1} [\sin(t)/\sin(\frac{\theta_0+\theta_1}{2})]^{D-1}dt\\
&\quad\to 0, \text{ as } D \text{ approaches infinity.}
\end{aligned} \tag{2.23}$$

Integrals in (2.22),(2.23) approach zero because θ_0, θ_1 are fixed and the integrant approaches zero as D approaches infinity.

□

Proof of Lemma 2.9.3. We use the same notation as in Proof of Theorem 3.5. In addition, we denote $B_{\mathrm{G}(D,1)}(L_1,\theta_0) \cap B_{\mathrm{G}(D,1)}(L_2',\theta_0)$ by X_4 and $\mu(X_4)$ by x_4 and $\mu(B_{\mathrm{G}(D,1)}(L_1,\theta_0))$ by x. It is easy to see $B_{\mathrm{G}(D,1)}(L_1,\theta_0) = X_1 \cup X_2$ and $X_2 = X_3 \cup X_4$.

Since X_1, X_3 are subsets of the hyperspherical cap $\mu(B_{\mathrm{G}(D,1)}(L,\theta_0))$. Lemma 2.10.7 implies that $\mu(B_{\mathrm{G}(D,1)}(L,\theta_0))$ approaches zero. So, x_1 and x_3 also approach zero as D approaches infinity.

Now, we show $\lim\limits_{D\to\infty} x_3/x_1 > e^{\alpha^2/2}$.

Let L_3 be the line passing through the middle point of the great circle connecting L_1 and L_2. Then, X_2 contains the hyperspherical cap $B_{\mathrm{G}(D,1)}(L_3,\theta_0-\frac{\theta}{2})$. this implies $x_3 + x_4 > \mu(B_{\mathrm{G}(D,1)}(L_3,\theta_0-\frac{\theta}{2}))$.

Moreover, since X_4 is contained in a hyperspherical cap with smaller angle than θ_0, $\lim\limits_{D\to\infty} x_4/x = 0$ by Lemma 2.10.7.

First, we compute

$$\begin{aligned}
&\lim_{D\to\infty} \int_0^{\theta_0} \sin^{D-1}(t)dt \Big/ \int_0^{\theta_0-\frac{\theta}{2}} \sin^{D-1}(t)dt \\
&= 1 + \lim_{D\to\infty} \int_{\theta_0-\frac{\theta}{2}}^{\theta_0} \sin^{D-1}(t)dt \Big/ \int_0^{\theta_0-\frac{\theta}{2}} \sin^{D-1}(t)dt \\
&< 1 + \lim_{D\to\infty} (\sin^D(\theta_0) * \frac{\theta}{2})/(\sin^D(\theta_0-\theta) * \frac{\theta}{2}) \\
&= 1 + \lim_{D\to\infty} (\sin(\theta_0)/\sin(\theta_0-\theta))^D \qquad (2.24) \\
&= 1 + \lim_{D\to\infty} \cos^D(\theta) \\
&= 1 + \lim_{D\to\infty} (1-\theta^2/2+O(\theta^4))^D \\
&= 1 + \lim_{D\to\infty} (1-(\alpha^2/2)/D+O(D^2))^D \\
&= 1 + e^{-\alpha^2/2}.
\end{aligned}$$

Then,

$$\begin{aligned}
&\lim_{D\to\infty} x_3/x_1 > \lim_{D\to\infty} x_3/(x-x_3) \\
&= \lim_{D\to\infty} (x/x_3-1)^{-1} \\
&= \lim_{D\to\infty} (x/(x_3+x_4)-1)^{-1} \\
&\geq \lim_{D\to\infty} \left(x \Big/ \mu(B_{\mathrm{G}(D,1)}(L_3,\theta_0-\frac{\theta}{2}))-1 \right)^{-1} \\
&= \lim_{D\to\infty} \left(\int_0^{\theta_0} \sin^{D-1}(t)dt \Big/ \int_0^{\theta_0-\frac{\theta}{2}} \sin^{D-1}(t)dt - 1 \right)^{-1} \\
&> e^{\alpha^2/2}. \qquad \text{(by (2.24))}
\end{aligned}$$

□

Chapter 3

Part II: Riemannian Multi-Manifold Modeling

[1]

3.1 Introduction

Many modern data sets are of moderate or high dimension, but manifest intrinsically low-dimensional structures. A natural quantitative framework for studying such common data sets is multi-manifold modeling (MMM) or its special case of hybrid-linear modeling (HLM). In this MMM framework a given dataset is modeled as a union of submanifolds (whereas HLM considers union of subspaces). When proposing a valid algorithm for MMM, one assumes an underlying dataset that can be modeled as mixture of submanifolds and tries to prove under some conditions that the proposed algorithm can cluster the dataset according to the submanifolds. This framework has been extensively studied and applied for datasets embedded in the Euclidean space or the sphere [23, 24, 25, 26, 27, 28, 29, 30].

Nevertheless, there is an overwhelming number of application domains, where information is extracted from datasets that lie on Riemannian manifolds, such as the

[1] A portion of Part II appearing in Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volumn 38. ©2015 by the authors X. Wang, K. Slavakis and G. Lerman.

Grassmannian, the sphere, the orthogonal group, or the manifold of symmetric positive (semi)definite [P(S)D] matrices. For example, auto-regressive moving average (ARMA) models are utilized to extract low-rank linear subspaces (points on the Grassmannian) for identifying spatio-temporal dynamics in video sequences [31]. Similarly, convolving patches of images by Gabor filters yields covariance matrices (points on the PD manifold) that can capture effectively texture patterns in images [32]. Nevertheless, current MMM strategies are not sufficiently accurate for handling data in more general Riemannian spaces.

The purpose of this paper is to develop theory and algorithms for the MMM problem in more general Riemannian spaces that are relevant to important applications.

Related Work. Recent advances in parsimonious data representations and their important implications in dimensionality reduction techniques have effected the development of non-standard spectral-clustering schemes that result in state-of-the-art results in modern applications [24, 33, 34, 35, 36, 37, 38, 39]. Such schemes rely on the assumption that data exhibit low-dimensional structures, such as unions of low-dimensional linear subspaces or submanifolds embedded in Euclidean spaces.

Several algorithms for clustering on manifolds are generalizations of well-known schemes developed originally for Euclidean spaces. For example, [40] extended the classical K-means algorithm from Euclidean spaces to Grassmannians, and illustrated an application to nonnegative matrix factorization. [41] capitalized on the Riemannian distance of SO(3) to design an efficient mean-shift (MS) algorithm for multiple 3D rigid motion estimation. [42], as well as [25], extended further the MS algorithm to general analytic manifolds including Grassmannians, Stiefel manifolds, and matrix Lie groups. [43] showed promising results by using the geodesic distance of product manifolds in clustering of human expressions, gestures, and actions in videos. [44] solved the image segmentation problem, after recasting it as a matrix clustering problem, via probability distributions on symmetric PD matrices. [35] extended spectral clustering and nonlinear dimensionality reduction techniques to Riemannian manifolds. These previous works are quite successful when the convex hulls of individual clusters are well-separated, but they often fail when clusters intersect or are closely located.

HLM and MMM accommodate low-dimensional data structures by unions of subspaces or submanifolds, respectively, but are restricted to manifolds embedded in either a Euclidean space or the sphere. Many strategies have been suggested for solving the HLM problem, known also as subspace clustering. These strategies include methods inspired by energy minimization [45, 19, 46, 47, 48, 49, 50], algebraic methods [51, 52, 53, 54, 55, 56, 57], statistical methods [58, 59], and spectral-type methods with various types of affinities representing subspace-related information [33, 60, 38, 61, 39]. Recent tutorial papers on HLM are [62] and [63]. Some theoretical guarantees for particular HLM algorithms appear in [64, 65, 66, 67]. There are fewer strategies for the MMM problem, which is also known as manifold clustering. They include higher-order spectral clustering [23], spectral methods based on local PCA [24, 36, 68, 27, 30], sparse-coding-based spectral clustering in a Euclidean space [26] and its modification to the sphere by [69] (the sparse coding encodes local subspace approximation), energy minimization strategies [70], methods based on manifold learning algorithms [71, 72], and methods based on clustering dimension or local density [73, 74, 75]. Notwithstanding, only higher-order spectral clustering and spectral local PCA are theoretically guaranteed [23, 24].

In a different context, [76] suggested multiscale strategies for signals taking values in Riemannian manifolds, in particular, the sphere, the orthogonal group, the Grassmannian, and the PD manifold. Even though [76] addresses a completely different problem, its basic principle is similar in spirit to ours and can be described as follows. Local analysis is performed in the tangent spaces, where the exponential and logarithm maps are used to transform data between local manifold neighborhoods and local tangent space neighborhoods. Information from all local neighborhoods is then integrated to infer global properties.

Contributions. Despite the popularity of manifold learning, the associated literature lacks generic schemes for clustering low-dimensional data embedded in non-Euclidean spaces. Furthermore, even in the Euclidean setting only few algorithms for MMM or HLM are theoretically guaranteed. To this end, this paper aims at filling this gap and provides an MMM approach in non-Euclidean setting with some theoretical guarantees even when the clusters intersect. In order to avoid nontrivial theoretical obstacles, the theory assumes that the underlying submanifolds are geodesic and refer to it as *multi-geodesic modeling* (MGM). Clearly, this modeling paradigm is a direct generalization of HLM from Euclidean spaces to Riemannian manifolds. A more practical and robust variant of the theoretical algorithm is also developed, and its superior performance over state-of-the-art clustering techniques is exhibited by extensive validation on synthetic and real datasets. We remark that in practice we require that the logarithm map of M can be computed efficiently and we show that this assumption does not restrict the wide applicability of this work.

We believe that it is possible to extend the theoretical foundations of this work to deal with general submanifolds by using local geodesic submanifolds (in analogy to [24]). However, this will significantly increase the complexity of our proof, which is already not simple. Nevertheless, the proposed method directly applies to the more general setting (without theoretical guarantees) since geodesics are only used in local neighborhoods and not globally. Furthermore, our numerical experiments show that the proposed method works well in real practical scenarios that deviate from the theoretical model.

On a more technical level, the paper is distinguished from previous works in multi-manifold modeling in its careful incorporation of "directional information," e.g., local tangent spaces and geodesics. This is done for two purposes: (i) To distinguish submanifolds at intersections; (ii) to filter out neighboring points that belong to clusters different than the cluster of the query point. In such a way, the proposed algorithm allows for neighborhoods to include points from different clusters, while previous multi-manifold algorithms (e.g., [26]) need careful choice of neighborhood radii to avoid points belonging to other clusters.

3.2 Theoretical Preliminaries

We formulate the theoretical problem of MGM and review preliminary background of Riemannian geometry, which is necessary to follow this work.

3.2.1 Multi-Geodesic Modeling (MGM)

MGM assumes that each point in a given dataset $X = \{x_i\}_{i=1}^N$ lies in the tubular neighborhood of some unknown geodesic submanifold S_k, $1 \leq k \leq K$, of a Riemannian manifold, M.[2] The goal is to cluster the dataset X into K groups $X_1, \ldots, X_K \subset M$ such that points in X_k are associated with the submanifold S_k. Note that if M is a Euclidean space, geodesic submanifolds are subspaces and MGM boils down to HLM, or equivalently, subspace clustering [34, 62, 39].

For theoretical purposes, we assume the following data model, which we refer to as uniform MGM: The data points are i.i.d. sampled w.r.t. the uniform distribution on a fixed tubular neighborhood of $\cup_{k=1}^K S_k$. We denote the radius of the tubular neighborhood by τ and refer to it as the noise level. Figure 3.1 illustrates data generated from uniform MGM with two underlying submanifolds ($K = 2$).

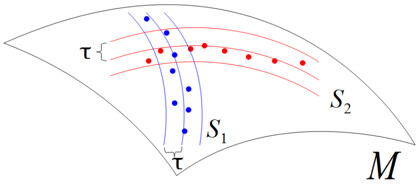


Figure 3.1: Illustration of data generated from a uniform MGM when $K = 2$.

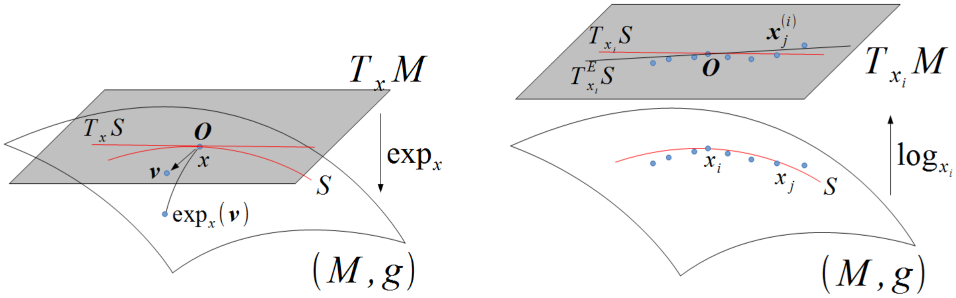
The MGM problem only serves our theoretical justification. The numerical experiments show that the proposed algorithm works well under a more general MMM setting. Such a setting may include more general submanifolds (not necessarily geodesic), non-uniform sampling and different kinds and levels of noise.

3.2.2 Basics of Riemannian Geometry

This section reviews basic concepts from Riemannian geometry; for extended and accessible review of the topic we recommend the textbook by [77]. Let (M, g) be a D-dimensional Riemannian manifold with a metric tensor g. A geodesic between $x, y \in M$ is a curve in M whose length is locally minimized among all curves connecting x and y. Let $\text{dist}_g(x, y)$ be the Riemannian distance between x and y on M. If $T_x M$ denotes the

[2] The tubular neighborhood with radius $\tau > 0$ of S_k in M (with metric tensor g and induced distance dist_g) is $S_k^\tau = \{x \in M : \text{dist}_g(x, s) < \tau \text{ for some } s \in S_k\}$.

tangent space of M at x, then T_xS stands for the tangent subspace of a d-dimensional geodesic submanifold S at x. As shown in Figure 3.2a, T_xS is a linear subspace of T_xM. The exponential map $\exp_x$ maps a tangent vector $\mathbf{v} \in T_xM$ to a point $\exp_x(\mathbf{v}) \in M$, which provides local coordinates around x. By definition, the geodesic submanifold S is the image of T_xS under $\exp_x$ (cf., Definition 3.5.1). The functional inverse of $\exp_x$ is the logarithm map $\log_x$ from M to T_xM, which maps x to the origin $\mathbf{O}$ of T_xM. Let $\mathbf{x}_j^{(i)}$ denote the image of a data point x_j in $T_{x_i}M$ by the logarithm map at x_i; that is, $\mathbf{x}_j^{(i)} = \log_{x_i}(x_j)$.



(a) Tangent space and exponential map

(b) Logarithm map and estimated tangent space

Figure 3.2: Demonstration of the exponential and logarithm maps as well as the tangent and estimated subspaces.

Figure 3.2a shows the tangent space and the exponential map of a manifold (M, g) at a point $x \in M$. Note that the tangent subspace T_xS is a pre-image of S under the exponential map. Figure 3.2b shows the logarithm map $\log_{x_i}$ w.r.t. $x_i \in S$ and the images by $\log_{x_i}$ of data points in a local neighborhood of x_i, in particular, $\mathbf{x}_j^{(i)}$, the image of x_j. Note the difference between $T_{x_i}S$, which is the image of S under $\log_{x_i}$, and the subspace $T_{x_i}^E S$ estimated by the images of the data points in the local neighborhood.

3.3 Solutions for the MGM (or MMM) Problem in M

We suggest solutions for the MMM problem in M with theoretical guarantees supporting one of these solutions when restricting the problem to MGM. Section 3.3.1 defines two key quantities for quantifying directional information: Estimated local tangent subspaces and geodesic angles. Section 3.3.2 presents the two solutions and discusses their properties.

3.3.1 Directional Information

The Estimated Local Tangent Subspace $T^E_{x_i}S$. Figure 3.2b demonstrates the main quantity defined here ($T^E_{x_i}S$) as well as related concepts and definitions. It assumes a dataset $X = \{x_j\}_{j=1}^N \subset M$ generated by uniform MGM with a single geodesic submanifold S. The dataset is thus contained in a tubular neighborhood of a d-dimensional geodesic submanifold S. Since S is geodesic, for any $1 \leq i \leq N$ the set $\{\mathbf{x}_j^{(i)}\}_{j=1}^N$ of images by the logarithm map is contained in a tubular neighborhood of the d-dimensional subspace $T_{x_i}S$ (possibly with a different radius than τ).

Since the true tangent subspace $T_{x_i}S$ is unknown, an estimation of it, $T^E_{x_i}S$, is needed. Let $B(x_i, r) \subset M$ be the neighborhood of x_i with a fixed radius $r > 0$. Let

$$J(x, r) := \{j : x_j \in B(x, r) \cap X\}. \tag{3.1}$$

Moreover, let $\mathbf{C}_{x_j}$ denote the local sample covariance matrix of the dataset $\{\mathbf{x}_j^{(i)}\}_{j \in J(x_i,r)}$ on $T_{x_i}M$, and $\|\mathbf{C}_{x_j}\|$ the spectral norm of $\mathbf{C}_{x_j}$, i.e., its maximum eigenvalue. Since $\{\mathbf{x}_j^{(i)}\}_{j=1}^N$ is in a tubular neighborhood of a d-dimensional subspace, estimates of the intrinsic dimension d of the local tangent subspace, which is also the dimension of S, can be formed by bottom eigenvalues of $\mathbf{C}_{x_j}$ (cf., [24]). We adopt this strategy of dimension estimation and define the estimated local tangent subspace, $T^E_{x_i}S$, as the span in $T_{x_i}M$ of the top eigenvectors of $\mathbf{C}_{x_j}$. In theory, the number of top eigenvectors is the number of eigenvalues of $\mathbf{C}_{x_i}$ that exceed $\eta\|\mathbf{C}_{x_j}\|$ for some fixed $0 < \eta < 1$ (see Theorem 3.3.1 and its proof for the choice of η). In practice, the number of top eigenvectors is the number of top eigenvalues $\mathbf{C}_{x_i}$ until the largest gap occurs.

Empirical Geodesic Angles. Let $l(x_i, x_j)$ be the shortest geodesic (global length minimizer) connecting x_i and x_j in (M, g). Let $\mathbf{v}_{ij} \in T_{x_i}M$ be the tangent vector of $l(x_i, x_j)$ at x_i. In other words, $\mathbf{v}_{ij}$ shows the direction at x_i of the shortest path from x_i to x_j. Given a dataset $X = \{x_j\}_{j=1}^N$, the empirical geodesic angle θ_{ij} is the elevation angle (cf., (9) of [78]) between the vector $\mathbf{v}_{ij}$ and the subspace $T^E_{x_i}S$ in the Euclidean space $T_{x_i}M$.

3.3.2 Proposed Solutions

In Section 3.3.2, we propose a theoretical solution for data sampled according to uniform MGM. We start with its basic motivation, then describe the proposed algorithm and at last formulate its theoretical guarantees. In Section 3.3.2, we propose a practical algorithm. At last, Section 3.3.2 discusses the numerical complexity of both algorithms.

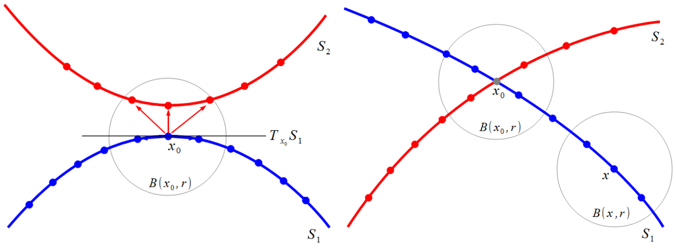
Algorithm 3: Theoretical Geodesic Clustering with Tangent information (TGCT)

The proposed solution for the MGM-clustering task applies spectral clustering with carefully chosen weights. Specifically, a similarity graph is constructed whose vertices are data points and whose edges represent the similarity between data points. The challenge is to construct a graph such that two points are locally connected only when they come from the same cluster. This way spectral clustering will recover exactly the underlying clusters.

For the sake of illustration, let us assume only two underlying geodesic submanifolds S_1 and S_2. We also assume that the data was sampled from $S_1 \cup S_2$ according to uniform MGM. Given a point $x_0 \in S_1$ one wishes to connect to it the points from the same submanifold within a local neighborhood $B(x_0, r)$ for some $r > 0$. Clearly, it is not realistic to assume that all points in $B(x_0, r)$ are from the same submanifold of x_0 (due to nearness and intersection of clusters as demonstrated in Figures 3.3a and 3.3b).

We first assume no intersection at x_0 as demonstrated in Figure 3.3a. In order to be able to identify the points in $B(x_0, r)$ from the same submanifold of x_0, we use local tangent information at x_0. If $x \in B(x_0, r)$ belongs to S_2, then the geodesic $l(x_0, x)$ has a large angle with the tangent space $T_{x_0}S_1$ at x_0. On the other hand, if such x belongs to S_1, then the geodesic has an angle close to zero. Therefore, thresholding the empirical geodesic angles may become beneficial for eliminating neighboring points belonging to a different submanifold (cf., Figure 3.3a).

In Figures 3.3a and 3.3b, points lie on two submanifolds S_1 and S_2. In Figure 3.3a, a local neighborhood, which is a disk of radius r, around the point x_0 is observed and the goal is to exclude the points from S_2 in $B(x_0, r)$. This can be done by thresholding the angles between geodesics and the tangent subspace $T_{x_0}S_1$. Indeed, the angles



(a) Angle filtering

(b) Intersection

Figure 3.3: Geometric observations

w.r.t. points from S_1 are close to zero and the angles w.r.t. points from S_2 are sufficiently large. In Figure 3.3b, a point x_0 is in $S_1 \cup S_2$ and an arbitrary point x sufficiently far from it. The goal is to assure that x is not connected to x_0. This can be done by comparing local estimated dimensions. The estimated dimension in $B(x_0, r)$ is $\dim(S_1) + \dim(S_2)$, while the estimated dimension in $B(x, r)$ is $\dim(S_1)$. Due to the dimension difference, the intersection is disconnected from the two submanifolds.

If x_0 is at or near the intersection, it is hard to estimate correctly the tangent spaces of each submanifold and the geodesic angles may not be reliable. Instead, one may compare the dimensions of estimated local tangent subspaces. The estimated dimensions of local neighborhoods of data points, which are close to intersections, are larger than the estimated dimensions of local neighborhoods of data points further away from intersections (cf., Figure 3.3b). The algorithm thus connects x_0 to other neighboring points only when their “local dimensions” (linear-algebraic dimension of the estimated local tangent) are the same. In this way, the intersection will not be connected with the other clusters.

The dimension difference criterion, together with the angle filtering procedure, guarantee that there is no false connection between different clusters (the rigorous argument is established in the proof of Theorem 3.3.1). We use these two simple ideas and the common spectral-clustering procedure to form the Theoretical Geodesic Clustering with Tangent information (TGCT) in Algorithm 3.

Algorithm 3 Theoretical Geodesic Clustering with Tangent information (TGCT)

```
Input: Number of clusters: $K \geq 2$, a dataset $X$ of $N$ points, a neighborhood radius
    $r$, a projection threshold $\eta$ for estimating tangent subspaces, a distance threshold $\sigma_d$
    and an angle threshold $\sigma_a$.
Output: Index set $\{\mathrm{Id}_i\}_{i=1}^N$ such that $\mathrm{Id}_i \in \{1, \ldots, K\}$ is the cluster label assigned to
    $x_i$
    Steps:
    • Compute the following geometric quantities around each point:
    for $i = 1, \ldots, N$ do
        ∘ For $j \in J(x_i, r)$ (c.f., (3.1)), compute $\mathbf{x}_j^{(i)} = \log_{x_i}(x_j)$
        ∘ Compute the sample covariance matrix $\mathbf{C}_{x_i}$ of $\{\mathbf{x}_j^{(i)}\}_{j \in J(x_i, r)}$
        ∘ Compute the eigenvectors of $\mathbf{C}_{x_i}$ whose eigenvalues exceed $\eta \cdot \|\mathbf{C}_{x_i}\|$ (their span
        is $T_{x_i}^E S$)
        ∘ For all $j = 1, \ldots, N$, compute the empirical geodesic angles $\theta_{ij}$ (see Section 3.3.1)
    end for
    • Form the following $N \times N$ affinity matrix $\mathbf{W}$:

        $$\mathbf{W}_{ij} = \mathbf{1}_{\mathrm{dist}_g(x_i, x_j) < \sigma_d} \mathbf{1}_{\dim(T_{x_i}^E S) = \dim(T_{x_j}^E S)} \mathbf{1}_{(\theta_{ij} + \theta_{ji}) < \sigma_a}$$

    • Apply spectral clustering to the affinity matrix $\mathbf{W}$ to determine the output $\{\mathrm{Id}_i\}_{i=1}^N$
```

The following theorem asserts that TGCT achieves correct clustering with high probability. Its proof is in Section 3.5. Its statement relies on the constants $\{C_i\}_{i=0}^6$ and C_0', which are clarified in the proof and depend only on the underlying geometry of the generative model. For simplicity, the theorem assumes that there are only two geodesic submanifolds and that they are of the same dimension. However, it can be extended to K geodesic submanifolds of different dimensions.

Theorem 3.3.1. *Consider two smooth compact d-dimensional geodesic submanifolds, S_1 and S_2, of a Riemannian manifold and let X be a dataset generated according to uniform MGM w.r.t. $S_1 \cup S_2$ with noise level τ. If the positive parameters of the TGCT* *algorithm, r, σ_d, σ_a and η, satisfy the inequalities*

$$\eta < C_2^{-\frac{d+2}{2}}, \sigma_d < C_4^{-\frac{1}{2}},\ r > \tau/C_5,\ r < \min(\eta, \sigma_d, \sigma_a)/C_1 \quad \textit{and} \tag{3.2}$$

$$\sigma_a < \min(\sin^{-1}(r\sqrt{1 - C_2\eta^{\frac{2}{d+2}}}/(2\sigma_d)) - C_3\eta^{\frac{d}{d+2}} - C_3 r, \pi/6),$$

then with probability at least $1 - C_0 N \exp[-Nr^{d+2}/C_0']$, the TGCT algorithm can cluster correctly a sufficiently large subset of X, whose relative fraction (over X) has expectation at least $1 - C_6(r + \tau)^{d - \dim(S_1 \cap S_2)}$.

Algorithm 4: Geodesic Clustering with Tangent information (GCT)

A practical version of the TGCT algorithm, which we refer to as Geodesic Clustering with Tangent information (GCT), is described in Algorithm 4. This is the algorithm implemented for the experiments in Section 3.4 and its choice of parameters is clarified in Section 3.7.2. GCT differs from TGCT in three different ways. First, hard thresholds in TGCT are replaced by soft ones, which are more flexible. Second, the dimension indicator function is dropped from the affinity matrix W. Indeed, numerical experiments indicate that the algorithm works properly without the dimension indicator function, whenever there is only a small portion of points near the intersection. This numerical observation makes sense since the dimension indicator is only used in theory to avoid connecting intersection points to points not in intersection. At last, pairwise distances are replaced by weights resulting from sparsity-cognizant optimization tasks. Sparse coding takes advantage of the low-dimensional structure of submanifolds and produces larger weights for points coming from the same submanifold [26].

Algorithm 4 Geodesic Clustering with Tangent information (GCT)

```
Input: Number of clusters: $K \geq 2$, a dataset $X$ of $N$ points, a neighborhood radius $r$,
    a distance threshold $\sigma_d$ (default: $\sigma_d = 1$) and an angle threshold $\sigma_a$ (default $\sigma_a = 1$)
Output: Index set $\{\mathrm{Id}_i\}_{i=1}^N$ such that $\mathrm{Id}_i \in \{1, \ldots, K\}$ is the cluster label assigned to
    $x_i$
  Steps:
  for $i = 1, \ldots, N$ do
    ∘ For $j \in J(x_i, r)$, compute $\mathbf{x}_j^{(i)} = \log_{x_i}(x_j)$
    ∘ Compute the weights $\{\mathbf{S}_{ij}\}_{j \in J(x_i,r)}$ that minimize
        $\|\mathbf{x}_i^{(i)} - \sum_{\substack{j \in J(x_i,r) \\ j \neq i}} \mathbf{S}_{ij} \mathbf{x}_j^{(i)}\|_2^2 + \sum_{\substack{j \in J(x_i,r) \\ j \neq i}} e^{\|\mathbf{x}_i^{(i)} - \mathbf{x}_j^{(i)}\|_2 / \sigma_d} |\mathbf{S}_{ij}|$    (3.3)
      among all $\{\mathbf{S}_{ij}\}_{j \in J(x_i,r)}$ such that $\mathbf{S}_{ii} = 0$ and $\sum_{\substack{j \in J(x_i,r) \\ j \neq i}} \mathbf{S}_{ij} = 1$
    ∘ Complete these weights as follows: $\mathbf{S}_{ij} = 0$ for $j \notin J(x_i, r)$
    ∘ Compute the sample covariance matrix $\mathbf{C}_{x_i}$ of $\{\mathbf{x}_j^{(i)}\}_{j \in J(x_i,r)}$
    ∘ Find the largest gap between eigenvalues $\lambda_m$ and $\lambda_{m+1}$ of $\mathbf{C}_{x_i}$ and compute the
      top $m$ eigenvectors of $\mathbf{C}_{x_i}$ (their span is $T_{x_i}^E S$)
    ∘ For all $j = 1, \ldots, N$, compute the empirical geodesic angles $\theta_{ij}$ (see Section 3.3.1)
  end for
  • Form the following $N \times N$ affinity matrix $\mathbf{W}$:
        $\mathbf{W}_{ij} = e^{|\mathbf{S}_{ij}| + |\mathbf{S}_{ji}|} e^{-(\theta_{ij} + \theta_{ji})/\sigma_a}$    (3.4)
  • Apply spectral clustering to the affinity matrix $W$ to determine the output $\{\mathrm{Id}_i\}_{i=1}^N$
```

Algorithm 4 solves a sparse coding task in (3.3). The penalty used is non-standard since the codes $|\mathbf{S}_{ij}|$ are multiplied by $e^{\|\mathbf{x}_i^{(i)} - \mathbf{x}_j^{(i)}\|_2 / \sigma_d}$ (where in [69], these latter terms are all 1). These weights were chosen to increase the effect of nearby points (in addition to their sparsity). In particular, it avoids sparse representations via far-away points that are unrelated to the local manifold structure (see further explanation in Figure 3.4). Similarly to [69], the clustering weights in (3.4) exponentiate the sparse-coding weights.

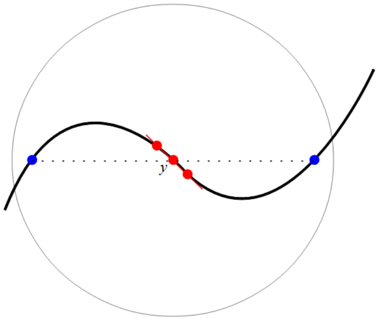


Figure 3.4: Illustration of the need for weighted sparse optimization in (3.3). The non-weighted sparse optimization may fail to detect the local structure at y in the manifold setting. The term $e^{\|\mathbf{x}_i^{(i)}-\mathbf{x}_j^{(i)}\|_2/\sigma_d}$ is used to avoid assigning large weights to the far-away blue points.

Computational Complexity of GCT and TGCT

We briefly discuss the computational complexity of GCT and TGCT, while leaving many technical details to Appendices 3.8 and 3.9. The computational complexity of GCT is

$$\mathcal{O}(N^2(\mathrm{CR}+\mathrm{CL}+D)+kN\log(N)+ND+Nk^3),$$

where k bounds the number of nearest neighbors in a neighborhood (typically $k=30$ by the choice of parameters), CR is the cost of computing the Riemannian distances between any two points and CL is the cost of computing the logarithm map of a given point w.r.t. another point. Furthermore, once CL was computed, CR= $\mathcal{O}(D)$. The complexity of CL depends on the Riemannian manifold M. If $M=\mathbb{S}^D$, then CL= $\mathcal{O}(D)$. If M is the space of symmetric PD matrices and $\dim(M)=D$, then CL=$\mathcal{O}(D^{1.5})$. If M is the Grassmannian, $\dim(M)=D$ and d is chosen to be of the same order as the dimension of the subspaces in M, then CL=$d(d+D/d)^2$. In all applications of Riemannian multi-manifold modeling we are aware of M is known and it is one of these examples. For more general or unknown M, estimation of the logarithm map is discussed in [79] (this estimation is rather slow).

It is possible to reduce the total computational cost under some assumptions. In particular, in theory, it is possible to implement TGCT (or more precisely an approximate variant of it) for the sphere or the Grassmannian with computational complexity of order

$$\mathcal{O}(N^{1+\rho}\mathrm{CR} + (k+1)N\log(N) + kN(\mathrm{CL}+D) + Nk^3),$$

where $\rho > 0$ is near zero.

3.4 Numerical Experiments

To assess performance on both synthetic and real datasets, the GCT algorithm is compared with the following algorithms: Sparse manifold clustering (SMC) [26, 69], which is adapted here for clustering within a Riemannian manifold and still referred to as SMC, spectral clustering with Riemannian metric (SCR) of [35], and embedded K-means (EKM). The three methods and choices of parameters for all four methods are reviewed in Appendix 3.7.2.

The ground truth labeling is given in each experiment. To measure the accuracy of each method, the assigned labels are first permuted to have the maximal match with the ground truth labels. The clustering rate is computed for that permuted labels as follows:

$$\text{clustering rate} = \frac{\text{\# of points whose group labels are the same as ground truth labels}}{\text{\# of total points}}.$$

3.4.1 Experiments with Synthetic Datasets

Six datasets were generated. Dataset I and II are from the Grassmannian $\mathrm{G}(6,2)$, datasets III and IV are from 3×3 symmetric positive-definite (PD) matrices, and datasets V and VI are from the sphere $\mathbb{S}^2$. Each dataset contains 260 points generated from two "parallel" or intersecting submanifolds (130 points on each) and cropped by white Gaussian noise. The exact constructions are described below.

Datasets I and II The first two datasets are on the Grassmannian $\mathrm{G}(6,2)$. In dataset I, 130 pairs of subspaces are drawn from the following non-intersecting submanifolds:

$$\mathbf{x}_1 = \mathrm{span}\{(\cos(\theta), 0, \sin(\theta), 0, 0, 0) + \frac{1}{40}\boldsymbol{\epsilon}_{1\times 6}, (0, \cos(\theta), 0, \sin(\theta), 0, 0) + \frac{1}{40}\boldsymbol{\epsilon}_{1\times 6}\},$$
$$\mathbf{x}_2 = \mathrm{span}\{(\cos(\phi), 0, \sin(\phi), 0, 0.5, 0) + \frac{1}{40}\boldsymbol{\epsilon}_{1\times 6}, (0, \cos(\phi), 0, \sin(\phi), 0.5, 0) + \frac{1}{40}\boldsymbol{\epsilon}_{1\times 6}\},$$

where θ, ϕ are equidistantly drawn from $[-\pi/3, \pi/3]$ and the noise vector $\boldsymbol{\epsilon}_{1\times 6}$ comprises i.i.d. normal random variables $\mathcal{N}(0,1)$.

In dataset II, 130 pairs of subspaces lie around two intersecting submanifolds as follows:

$$\begin{aligned}\mathbf{x}_1 &= \operatorname{span}\{(\cos(\theta), 0, \sin(\theta), 0, 0, 0) + \frac{1}{40}\boldsymbol{\epsilon}_{1\times 6}, (0, \cos(\theta), 0, \sin(\theta), 0, 0) + \frac{1}{40}\boldsymbol{\epsilon}_{1\times 6}\},\\ \mathbf{x}_2 &= \operatorname{span}\{(\cos(\phi), 0, 0, 0, \sin(\phi), 0) + \frac{1}{40}\boldsymbol{\epsilon}_{1\times 6}, (0, \cos(\phi), 0, 0, 0, \sin(\phi)) + \frac{1}{40}\boldsymbol{\epsilon}_{1\times 6}\},\end{aligned}$$

where θ, ϕ are equidistantly drawn from $[-\pi/3, \pi/3]$ and the noise vector $\boldsymbol{\epsilon}_{1\times 6}$ comprises, again, i.i.d. normal random variables $\mathcal{N}(0,1)$.

Datasets III and IV The next two datasets are contained in the manifold of 3 × 3 symmetric PD matrices.

In dataset III, 130 pairs of matrices of two intersecting groups are generated from the model

$$\begin{aligned}\mathbf{A}_1 &= \begin{pmatrix} 4 & 4\cos(\theta+\pi/4) & 4\sin(\theta+\pi/4) \\ 4\cos(\theta+\pi/4) & 4 & 0 \\ 4\sin(\theta+\pi/4) & 0 & 4 \end{pmatrix} + \boldsymbol{\epsilon}_{3\times 3}/40,\\ \mathbf{A}_2 &= \begin{pmatrix} 4 & 0 & 4\cos(\theta-\pi/4) \\ 0 & 4 & 4\sin(\theta-\pi/4) \\ 4\cos(\theta-\pi/4) & 4\sin(\theta+\pi/4) & 4 \end{pmatrix} + \boldsymbol{\epsilon}_{3\times 3}/40,\end{aligned} \tag{3.5}$$

where θ is equidistantly drawn from $[0, \pi]$ and $\boldsymbol{\epsilon}_{3\times 3}$ is a symmetric matrix whose entries are i.i.d. normal random variables with distribution $\mathcal{N}(0,1)$.

In dataset IV, 130 pairs of matrices of two non-intersecting groups are generated from the model

$$\mathbf{A}_1 = \begin{pmatrix} 10\alpha & 0 & 0 \\ 0 & 10\alpha & 0 \\ 0 & 0 & 10\alpha \end{pmatrix} + \boldsymbol{\epsilon}_{3\times 3}/40, \quad \mathbf{A}_2 = \begin{pmatrix} 10\beta & 0 & 0 \\ 0 & 10\beta^2 & 0 \\ 0 & 0 & 10\beta^3 \end{pmatrix} + \boldsymbol{\epsilon}_{3\times 3}/40,$$

where α, β are equidistantly drawn from $[0.5, 1]$ respectively and $\boldsymbol{\epsilon}_{3\times 3}$ is a symmetric matrix whose entries are i.i.d. normal random variables with distribution $\mathcal{N}(0,1)$.

Datasets V and VI Two datasets are constructed on the unit sphere $\mathbb{S}^2$ of the 3-dimensional Euclidean space. Dataset V comprises of vectors lying around the following two parallel arcs:

$$\begin{aligned}\mathbf{x}_1 &= [\cos(\theta), \sin(\theta), 0] + \boldsymbol{\epsilon}_{1\times 3},\\ \mathbf{x}_2 &= [\sqrt{0.97}\cos(\phi), \sqrt{0.97}\sin(\phi), \sqrt{0.03}] + \boldsymbol{\epsilon}_{1\times 3},\end{aligned}$$

where θ, ϕ are equidistantly drawn from $[0, \pi/2]$. To ensure membership in $\mathbb{S}^2$, vectors generated by $\mathbf{T}_1$ and $\mathbf{T}_2$ are normalized to unit length. On the other hand, dataset VI considers the following two intersecting arcs:

$$\begin{aligned}\mathbf{x}_1 &= [\cos(\theta + \pi/4), \sin(\theta + \pi/4), 0] + \frac{1}{40}\boldsymbol{\epsilon}_{1\times 3},\\ \mathbf{x}_2 &= [0, \cos(\phi - \pi/4), \sin(\phi - \pi/4)] + \frac{1}{40}\boldsymbol{\epsilon}_{1\times 3}.\end{aligned}$$

Numerical Results

Each one of the six datasets is generated according to the postulated models above, and the experiment is repeated 30 times. Table 3.1 shows the average clustering rate for each method. GCT, SMC and SCR are all based on the spectral clustering scheme. However, when a dataset has low-dimensional structures, GCT's unique procedure of filtering neighboring points ensures that it yields superior performance over the other methods. This is because both SMC and SCR are sensitive to the local scale σ, and require each neighborhood not to contain points from different groups. This becomes clear by the results on datasets I, IV, and V of non-intersecting submanifolds. SMC only works well in dataset I, where most of the neighborhoods $B(x_0, r)$ contain only points from the same cluster, while neighborhoods $B(x_0, r)$ in datasets IV and V often contain points from different ones. Embedded K-means generally requires that the intrinsic means of different clusters are located far from each other. Its performance is not as good as GCT when different groups have low-dimensional structures.

3.4.2 Robustness to Noise and Running Time

Section 3.4.1 illustrated GCT's superior performance over the competing SMC, SCR, and EKM on a variety of manifolds. This section further investigates GCT's robustness to noise and computational cost pertaining to running time. In summary, GCT is shown

Methods	Set I	Set II	Set III	Set IV	Set V	Set VI
GCT	**1.00**	**0.98**	**0.98**	**0.95**	**0.98**	**0.96**
SMC	0.97	0.66	0.88	0.80	0.55	0.69
SCR	0.51	0.66	0.84	0.80	0.50	0.53
EKM	0.50	0.50	0.67	0.50	0.50	0.67

Table 3.1: Average clustering rates on the six synthetic datasets of Section 3.4.1.

to be far more robust than SMC in the presence of noise, at the price of a small increase of running time.

Robustness to Noise

The proposed tangent filtering scheme enables GCT to successfully eliminate neighboring points that originate from different groups. As such, it exhibits robustness in the presence of noise and/or whenever different groups are close or even intersecting. On the other hand, SMC appears to be sensitive to noise due to its sole dependence on sparse weights. Figures 3.5 and 3.6 demonstrate the performance of GCT, SMC, SCR, and EKM on the Grassmannian and the sphere for various noise levels (standard deviations of Gaussian noise).

The datasets in Figure 3.5 are generated on the Grassmannian according to the model of dataset II in Section 3.4.1 but with different noise levels (in Section 3.4.1 the noise level was 0.025). Both SMC and SCR appear to be volatile over different datasets, with their best performance never exceeding 0.75 clustering rate. It is worth noticing that EKM shows poor clustering accuracy. On the contrary, GCT exhibits remarkable robustness to noise, achieving clustering rates above 0.9 even when the standard deviation of the noise approaches 0.1.

GCT's robustness to noise is also demonstrated in Figure 3.6, where datasets are generated on the unit sphere according to the model of the dataset VI, but with different noise levels. SMC appears to be volatile also in this setting; it collapses when the standard deviation of noise exceeds 0.05, since its affinity matrix precludes spectral clustering from identifying eigenvalues with sufficient accuracy (see further explanation

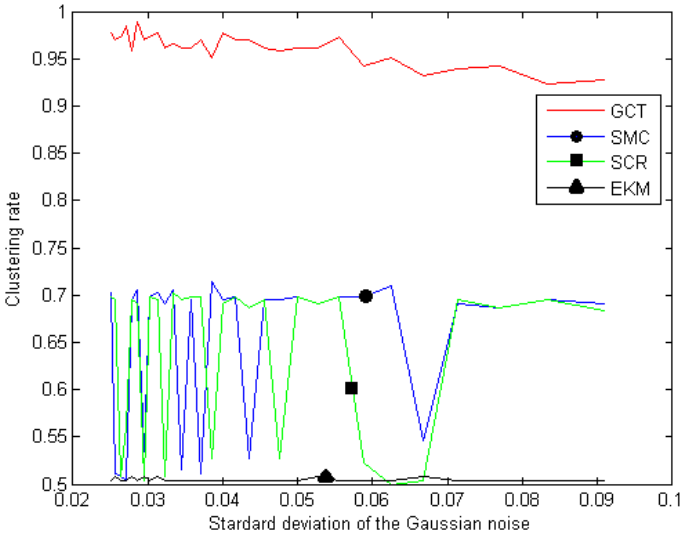


Figure 3.5: Performance of clustering methods on the Grassmannian for various noise levels. Datasets are generated according to the model of dataset II, but with an increasing standard deviation of the noise.

on the collapse of SMC at the end of Section 3.7.2).

Running time

This section demonstrates that GCT outperforms SMC at the price of a small increase in computational complexity. Similarly to any other manifold clustering algorithm, computations have to be performed per local neighborhood, where local linear structures are leveraged to increase clustering accuracy. The overall complexity scales quadratically w.r.t. the number of data-points due to the last step of Algorithm 4, which amounts to spectral clustering of the $N \times N$ affinity matrix $\mathbf{W}$. Both the optimization task of (3.3) and the computation of a few principal eigenvectors of the covariance matrix $\mathbf{C}_{x_i}$ in Algorithm 4 do not contribute much to the complexity since operations are performed on a small number of points in the neighborhood $J(x_i, r)$. The computational complexity of GCT is detailed in Appendix 3.9. It is also noteworthy that GCT can be fully parallelized since computations per neighborhood are independent. Nevertheless, such a route is not followed in this section.

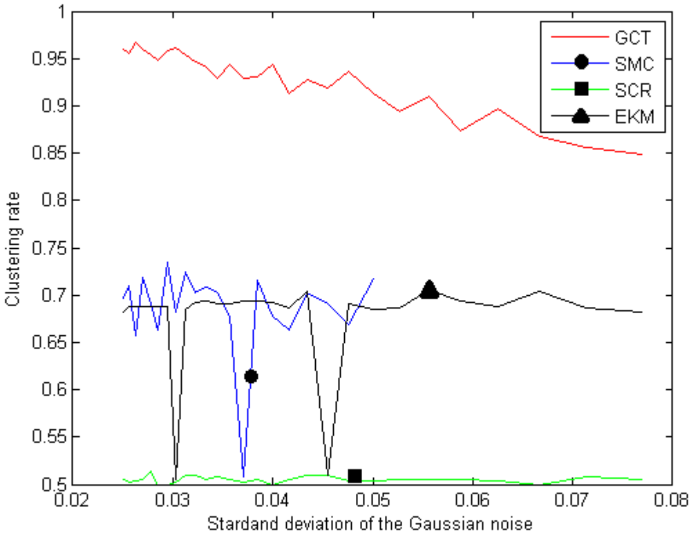


Figure 3.6: Performance of clustering methods on the sphere for various noise levels. Datasets are generated according to the model of dataset VI, but with an increasing standard deviation of the noise.

Running-time ratio	G(6, 2)	$PD_{3\times3}$	$\mathbb{S}^2$
GCT/SMC	1.06	1.05	1.11

Table 3.2: Ratio of running times of GCT and SMC for instances of the synthetic datasets I, IV and VI

Compared with SMC, GCT has one additional component: identifying tangent spaces through local covariance matrices—a task that entails local calculation of a few principal eigenvectors. Nevertheless, it is shown in Appendix 3.9.1 that for k neighbors it can be calculated with $\mathcal{O}(D + k^3)$ operations.

The ratios of running times between GCT and SMC for all three types of manifolds are illustrated in Table 3.2. It can be readily verified that the extra step of identifying tangent spaces in GCT increases running time by less than 11% of the one for SMC.

Ratios of running times were also investigated for increasing ambient dimensions of the sphere. More precisely, dataset VI of Section 3.4.1, which lies in S^2, was embedded via a random orthonormal matrix into the unit sphere $\mathbb{S}^D$, where D ranged from 100 to $3,000$. Figure 3.7 shows the ratios of the running time of GCT over that of SMC as a function of D. We observe that the extra cost of computing the eigendecomposition in GCT is mostly less than 20% of SMC, and never exceeds 30%, even when the ambient dimension is as large as $3,000$.

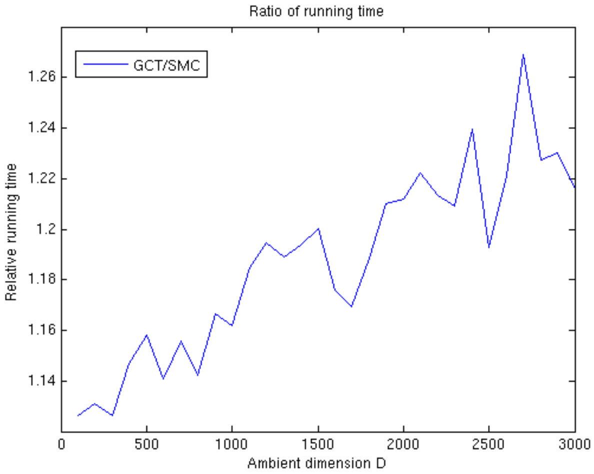


Figure 3.7: Relative running times of GCT w.r.t. SMC as the ambient dimension increases. With dimensions D ranging from 100 to $3,000$, dataset VI of Section 3.4.1 was embedded via a random orthonormal matrix into the unit sphere $\mathbb{S}^D$. Dataset VI of Section 3.4.1 matrix with two random

3.4.3 Synthetic Brain Fibers Segmentation

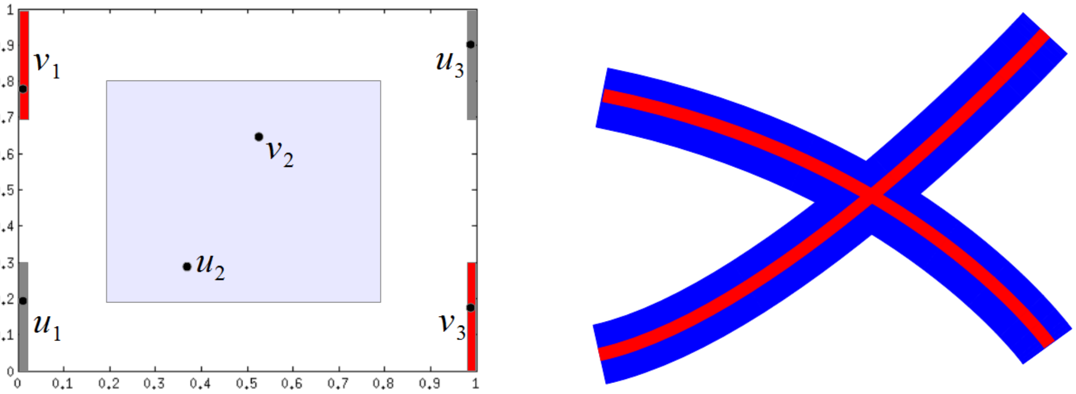
[69] cast the problem of segmenting diffusion magnetic resonance imaging (DMRI) data of different fiber tracts as a clustering problem on $\mathbb{S}^D$. The crux of the methodology lies on the transformation of diffusion images, associated with different views of the same object, into orientation distribution functions (ODFs), which are nothing but probability density functions on $\mathbb{S}^2$. The discretized ODF (dODF) is a probability mass function (pmf) $\mathbf{f} : (\mathbb{S}^2)^{D+1} \to \mathbb{R}_+^{D+1} : (\mathbf{s}_1, \ldots, \mathbf{s}_{D+1}) \mapsto \mathbf{f}(\mathbf{s}_1, \ldots, \mathbf{s}_{D+1}) := [f_1(\mathbf{s}_1), \ldots, f_{D+1}(\mathbf{s}_{D+1})]^T$, with $\sum_{i=1}^{D+1} f_i(\mathbf{s}_i) = 1$, that describes the water diffusion

pattern at a corresponding location of the object's image according to the viewing directions $\{\mathbf{s}_i\}_{i=1}^{D+1}$. Given $\{\mathbf{s}_i\}_{i=1}^{D+1}$ and a fixed location, the *square-root* (SR)dODF is the vector $\sqrt{\mathbf{f}}(\mathbf{s}_1, \ldots, \mathbf{s}_{D+1}) := [\sqrt{f_1(\mathbf{s}_1)}, \ldots, \sqrt{f_{D+1}(\mathbf{s}_{D+1})}]^T$, which lies on the sphere $\mathbb{S}^D$ since $\mathbf{f}$ is a pmf. In this way, pixels of diffusion images of the same object at a given location are mapped into an element of $\mathbb{S}^D$. [69] assume that each fiber tract is mapped into a submanifold of $\mathbb{S}^D$ and thus try to identify different fiber tracts by multi-manifold modeling on $\mathbb{S}^D$.

As suggested in [69], to differentiate pixels with similar diffusion patterns but located far from each other in an image, one has to incorporate pixel spatial information in the segmentation algorithm. Therefore, for GCT, SMC and SCR, the similarity entry $\mathbf{W}_{ij}$ of two pixels $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^2$ is modified as

$$\mathbf{W}_{ij}^{\text{new}} = \mathbf{W}_{ij} \cdot e^{-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/\sigma},$$

where $\mathbf{W}$ is the similarity matrix before modification (e.g., for GCT, it is described in Algorithm 2), $\sigma = 0.1$ and $\|\mathbf{x}_i - \mathbf{x}_j\|_2$ is the Euclidean distance between two pixels. For EKM, where no spectral clustering is employed, the dODF is simply augmented with the spatial coordinates of $\mathbf{x}_i$ and $\mathbf{x}_j$.



(a) Randomly sampled 6 points from the colored regions of the $[0, 1] \times [0, 1]$ domain.

(b) A configuration of two intersecting fibers generated according to points in Figure 3.8a

Figure 3.8: Demonstration of fiber generation. Two fibers are generated in Figure 3.8b by fitting two cubic splines to $\{\mathbf{u}_i\}_{i=1}^3$ and $\{\mathbf{v}_i\}_{i=1}^3$ in Figure 3.8a, respectively.

Following [69], we consider here the problem of segmenting or clustering two 2D

Methods	SNR=40	SNR=30	SNR=20	SNR=10
GCT	**0.80 ± 0.12**	**0.82 ± 0.12**	**0.78 ± 0.14**	**0.80 ± 0.13**
SMC	0.73 ± 0.14	0.73 ± 0.13	0.70 ± 0.13	0.67 ± 0.13
SCR	0.66 ± 0.11	0.66 ± 0.11	0.68 ± 0.11	0.66 ± 0.11
EKM	0.59 ± 0.08	0.58 ± 0.08	0.61 ± 0.08	0.59 ± 0.08

Table 3.3: Mean ± standard deviation of accuracy rates for 100 experiments on clustering synthetic brain fibers.

synthetic fiber tracts in the $[0,1] \times [0,1]$ domain. To generate the fibers, six points $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ are randomly chosen in the regions of Figure 3.8a. Two cubic splines passing through $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ and $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$, respectively, are set to be the center of the fibers (cf., thin curves (red) in Figure 3.8b). Fibers are defined as the curved bands around the splines with bandwidth 0.12 (cf., thick region (blue) in Figure 3.8b).

Given a pair of such fibers, the next step is to map each pixel (e.g., both red and blue ones in Figure 3.8b) to a point (SRdODF) in $\mathbb{S}^D$. To this end, the software code provided by [80] is used to generate SRdODFs on $\mathbb{S}^{100}$, where diffusion images $\{S_n\}_{n=1}^G$ at $G = 70$ gradient directions, with baseline image $S_0 = 100$ and $b = 4,000\text{s/mm}^2$, are considered. The dimensionality of the generated SRdODFs corresponds to 100 directions. Moreover, Gaussian noise $\mathcal{N}(0, \sigma^2)$ was added in the ODF-generation mechanism, resulting in a signal-to-noise ratio SNR $= S_0/\sigma$ (more details on the construction can be found in [69]). Typical noise levels for real-data brain images are considered: SNR $= 10, 20, 30$, and 40 (i.e., $\sigma = 10, 5, 10/3, 2.5$).

Once SRdODFs are formed, clustering is carried out on the Riemannian manifold $\mathbb{S}^D$. This in turn provides a segmentation of pixels according to different fiber tracts. A total number of 100 pairs of synthetic brain fibers are randomly generated, and clustering is performed for each pair. Table 3.3 reports the mean ± standard deviation of the clustering accuracy rates. Results clearly suggest that GCT outperforms the other three clustering methods. For the case of SNR $= 10$, Figure 3.9 plots sample distributions of accuracy rates and shows that GCT demonstrates the highest probability of achieving almost accurate clustering among competing schemes. In Figure 3.9, each bar shows the number of experiments whose rates fall within one of the ten intervals of length 0.05 in the partition (in each interval, four bars from left to right correspond to four algorithms GCT, SMC, SCR and EKM respectively). For example, since the tallest bar within the $[0.95, 1]$ range is the first bar (corresponds to GCT), GCT is the most likely method to achieve almost accurate clustering. On the contrary, the fourth bar (brown) is the tallest one between the range of 0.5 and 0.55, meaning that a clustering rate within $[0.5, 0.55]$ is the most likely one to be achieved for EKM over 100 experiments.

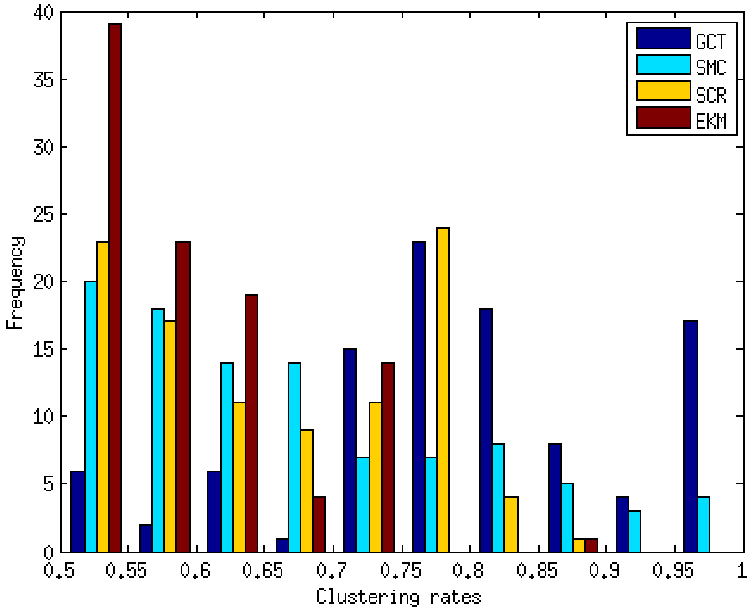


Figure 3.9: Histogram of clustering rates for the noise level SNR = 10 over a total number of 100 experiments.

3.4.4 Experiments with Real Data

In this section, GCT performance is assessed on real datasets. Scenarios where data within each cluster have submanifold structures are demonstrated.

Stylized Application: Texture Clustering

We cluster local covariance matrices obtained from various transformations of images of the Brodatz database [81] where the goal is to be able to distinguish between the different images independently of the transformation.

The Brodatz database contains 112 images of 640 × 640 pixels with different textures (e.g., brick wall, beach sand, grass) captured under uniform lighting and in frontview position. We apply three simple deformations to these images, which mimic real settings: different lighting conditions, stretching (obtained by shearing) and different viewpoints (obtained by affine transformation). Figure 3.10 shows sample images in the Brodatz database and their deformations. The first row shows the 6 original images; in the second row, each image contains a unique texture but different regions of it have different lighting; the third row shows the horizontal-shifted (distorted) images of an image; the fourth row shows affine-transformed (change of viewpoints) images of an image.

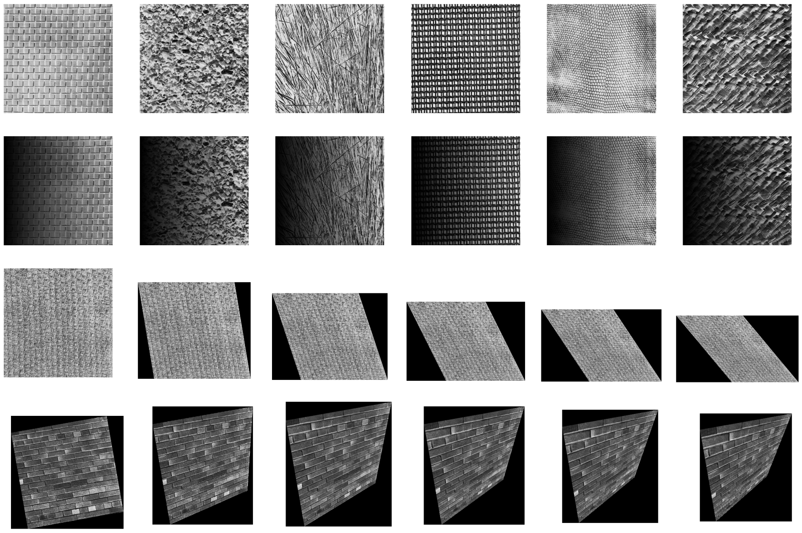


Figure 3.10: Sample images in the Brodatz database and their deformations.

[32] show that region covariances generated by Gabor filters effectively represent texture patterns in a region (patch). Given a patch of size 60 × 60, a Gabor filter of size 11×11 with 8 parameters is used to extract 2,500 feature vectors of length 8. This set of feature vectors is then used to compute an 8×8 covariance matrix for the specific patch.

Three clustering tests, one for each type of deformation, are carried out. In each test, 300 transformed patches are generated equally from 3 different textures and the region covariance is computed for each patch. Then clustering algorithms are applied on the dataset of 300 region covariances belonging to 3 texture patterns. The way to generate transformed patches is described below.

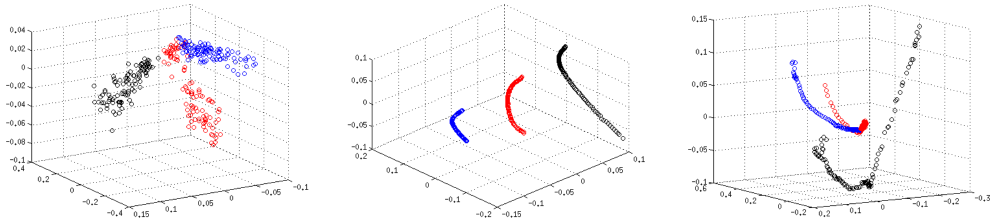
I. Lighting transformation: A single lighting transformation (demonstrated in Figure 3.10) is applied to three randomly drawn images from the Brodatz database and 100 patches of size 60×60 are randomly picked from each of the 3 transformed images.

II. Horizontal shearing: Three randomly drawn images are horizontally sheared by 100 different angles to get 3 sequences of 100 shifted images. From each shifted image, a patch of size 60×60 is randomly picked.

III. Affine transformation: Three randomly drawn images are affine transformed to create 3 sequences of 100 affine-transformed images. From each transformed image, a patch of size 60×60 is randomly picked.

Figure 3.11 plots the projection of the embedded datasets generated by the above procedure onto their top three principal components (the embedding to Euclidean spaces is done by direct vectorization of the covariance matrices). For 3 sample images, a dataset of 300 covariance matrices is computed for each transformation type. The 8×8 covariance matrices are identified as vectors in $\mathbb{R}^{64}$. The figure demonstrates the underlying structure of 3 manifolds for the data generated with each kind of transformation. The submanifold structure in each cluster can be easily observed.

The procedure of generating the data is repeated 30 times for each type of transformation. GCT as well as the other three clustering methods are applied to these datasets, and the average clustering rates are reported in Table 3.4. GCT exhibits the best performance for all datasets and for all types of transforms.



(a) Lighting transformation (b) Horizontal shearing (c) Affine transformation

Figure 3.11: Projection of the covariance matrices of local patches of the transformed 3 images onto their top 3 principal directions.

Methods	GCT	SMC	SCR	EKM
Lighting transformation	**0.73**	0.53	0.68	0.67
Horizontal shifting	**0.95**	0.61	0.85	0.76
Affine tranformation	**0.83**	0.53	0.82	0.76

Table 3.4: Average clustering rates for each method over 30 datasets.

Clustering Dynamic Patterns.

Spatio-temporal data such as dynamic textures and videos of human actions can often be approximated by linear dynamical models [82, 31]. In particular, by leveraging the autoregressive and moving average (ARMA) model, we experiment here with two spatio-temporal databases: Dyntex++ and Ballet. Following [31], we employ the ARMA model to associate local spatio-temporal patches with linear subspaces of the same dimension. We then apply manifold clustering on the Grassmannian in order to distinguish between different textures and actions in the Dyntex++ and Ballet database respectively.

ARMA Model. The premise of ARMA modeling is based on the assumption that the spatio-temporal dataset under study is governed by a small number of latent variables whose temporal variations obey a linear rule. More specifically, if $\mathbf{f}(t) \in \mathbb{R}^p$ is the observation vector at time t (in our case, it is the vectorized image frame of a video sequence), then

$$\begin{aligned} \mathbf{f}(t) &= \mathbf{C}\mathbf{z}(t) + \boldsymbol{\epsilon}_1(t) \qquad && \boldsymbol{\epsilon}_1(t) \sim N(\mathbf{0}, \boldsymbol{\Sigma}_1) \\ \mathbf{z}(t+1) &= \mathbf{A}\mathbf{z}(t) + \boldsymbol{\epsilon}_2(t) \qquad && \boldsymbol{\epsilon}_2(t) \sim N(\mathbf{0}, \boldsymbol{\Sigma}_2) \end{aligned} \tag{3.6}$$

where $\mathbf{z}(t) \in \mathbb{R}^\ell$, $\ell \leq p$, is the vector of latent variables, $\mathbf{C} \in \mathbb{R}^{p\times\ell}$ is the observation matrix, $\mathbf{A} \in \mathbb{R}^{\ell\times\ell}$ is the transition matrix, and $\boldsymbol{\epsilon}_1(t) \in \mathbb{R}^p$ and $\boldsymbol{\epsilon}_2(t) \in \mathbb{R}^\ell$ are i.i.d. sampled vector-values r.vs. obeying the Gaussian distributions $\mathcal{N}(0, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}(0, \boldsymbol{\Sigma}_2)$, respectively.

We next explain the idea of [31] to associate subspaces with spatio-temporal data. Given data $\{\mathbf{f}(t)\}_{t=\tau_1}^{\tau_2}$, the ARMA parameters $\mathbf{A}$ and $\mathbf{C}$ can be estimated according to the procedure in [31]. Moreover, by arbitrarily choosing $\mathbf{z}(0)$, it can be verified that for any $m \in \mathbb{N}$,

$$\mathbb{E}\begin{bmatrix} \mathbf{f}(\tau_1) \\ \mathbf{f}(\tau_1+1) \\ \vdots \\ \mathbf{f}(\tau_1+m-1) \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{m-1} \end{bmatrix} \mathbf{z}(\tau_1).$$

We then set $\mathbf{V} := [\mathbf{C}^T, (\mathbf{CA})^T, ..., (\mathbf{CA}^{m-1})^T]^T \in \mathbb{R}^{mp\times\ell}$, which is known as the mth order observability matrix. If the observability matrix is of full column rank, which was the case in all of the conducted experiments, the column space of $\mathbf{V}$ is a ℓ-dimensional linear subspace of $\mathbb{R}^{pm}$. In other words, the ARMA model estimated from data $\{\mathbf{f}(t)\}_{t=\tau_1}^{\tau_2}$, $\tau_1 \leq \tau_2$, gives rise to a point on the Grassmannian $\mathrm{G}(mp, \ell)$. For a fixed dataset $\{\mathbf{f}(t)\}_{t=1}^{\tau}$, different choices of (τ_1, τ_2), s.t. $\tau_1, \tau_2 \leq \tau$, and several local regions within the image give rise to different estimates of $\mathbf{A}$ and $\mathbf{C}$ and thus to different points in $\mathrm{G}(mp, \ell)$.

Dynamic textures. The Dyntex++ database [83] contains 3600 dynamic textures videos of size $50 \times 50 \times 50$, which are divided into 36 categories. It is a hard-to-cluster database due to its low resolution. Three videos were randomly chosen, each one from a distinct category from the available 36 ones.

Per video sequence, 50 patches of size $40 \times 40 \times 20$ are randomly chosen. Each frame of the patch is vectorized resulting into patches of size 1600×20. To reduce the size to 30×20, a (Gaussian) random (linear) projection operator is applied to each patch. As a result, each patch is reduced to the set $\{\mathbf{f}(t)\}_{t=\tau_1}^{\tau_1+20} \subset \mathbb{R}^{30}$. We fix $\ell = 3$ and $m = 3$ and use each such set $\{\mathbf{f}(t)\}_{t=\tau_1}^{\tau_1+20}$ to estimate the underlying ARMA model. Consequently, 150 points on $\mathrm{G}(90,3)$ are generated, 50 per video category.

We expect that points in $\mathrm{G}(90,3)$ of the same cluster lie near a submanifold of $\mathrm{G}(90,3)$. This is due to the repeated pattern of textures in space and time (they often look like a shifted version of each other in space and time). To visualize the submanifold structure, we isometrically embedded $\mathrm{G}(90,3)$ into a Euclidean space [84], so that subspaces are mapped to Euclidean points. We then projected the latter points on their top 3 principal components. Figure 3.13a demonstrates this projection as well as the submanifold structure within each cluster.

Ballet database. The Ballet database [85] contains 44 videos of 8 actions from a ballet instruction DVD. The frames of all videos are of size 301×301 and their lengths vary and are larger than 100. Different performers have different attire and speed. Three videos, each one associated with a different action, were randomly chosen.

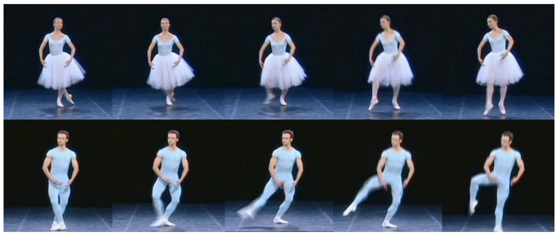


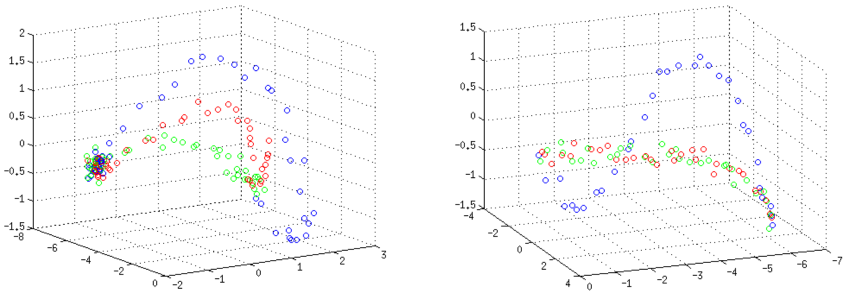
Figure 3.12: Two samples of Ballet video sequences: the first and second rows are from videos demonstrating actions of hopping and leg-swinging, respectively.

Spatio-temporal patches are generated by selecting 10 consecutive frames of size 301×301 from each one of the following overlapping time intervals: $\{1,\ldots,10\}$, $\{4,\ldots,13\}$, $\{7,\ldots,16\}$, $\ldots$, $\{91,\ldots,100\}$. In this way, for each of the three videos, 31 spatio-temporal patches of size $301 \times 301 \times 10$ are generated. As in the case of the Dyntex++ database, video patches are vectorized and downsized to spatio-temporal patches of size 30×10. Following the previous ARMA modeling approach, we set $\ell = 3$ and $m = 3$ and associate each such patch with a subspace in $\mathrm{G}(90,3)$. Consequently, 93 subspaces (31 per cluster) in the Grassmannian $\mathrm{G}(90,3)$ are generated. Figure 3.13b visualizes the 3D representation of the subspaces created from three random videos. Their intersection represents still motion.

Methods	GCT	SMC	SCR	EKM
Dyntex++	**0.85**	0.69	0.77	0.42
Ballet	**0.81**	0.76	0.68	0.47

Table 3.5: Average clustering accuracy rates for the Dyntex++ and Ballet datasets.

The procedure described above (for generating data by randomly choosing 3 videos from the Dyntex++ and Ballet databases and applying clustering methods on $\mathrm{G}(90,3)$) is repeated 30 times. The average clustering accuracy rates are reported in Table 3.5. GCT achieves the highest rates on both datasets.



(a) Dyntex++

(b) Ballet

Figure 3.13: Projection onto top 3 principal components of the two embedded datasets (the embedding into Euclidean spaces is according to [84]). A submanifold structure for each cluster is clearly depicted.

3.5 Proof of Theorem 3.3.1

The idea of the proof is as follows. After excluding points sampled near the possibly nonempty intersection of submanifolds, we form a graph whose vertices are the points of the remaining set and whose edges are determined by $\mathbf{W}$. The proof then establishes that the resulting graph has two connected components, which correspond to the two different submanifolds S_1 and S_2. Spectral clustering can exactly cluster such a graph with appropriate choice of its tuning parameter σ, which can be specified by self-tuning mechanism [86]. This claim follows from [87] and its unpublished supplemental material.

The basic strategy of the proof and its organization are described as follows. Section 3.5.1 presents additional notation used in the proof. Section 3.5.2 reminds the reader the underlying model of the proof (with additional details). Section 3.5.3 eliminates undesirable events of negligible probability (it clarifies $1 - C_0 N \exp[-Nr^{d+2}/C_0']$ in the statement of the theorem).

The rest of the proof (described in Sections 3.5.4-3.5.8) is briefly sketched as follows. For simplicity, we first assume no noise, i.e., $\tau = 0$. We define a "sufficiently large" set X^* (and its subsets X_1^* and X_2^*) by the following formula (which uses the notation $X_1 = S_1 \cap X$ and $X_2 = S_2 \cap X$):

$$X_1^* = \{x \in X_1 | B(x,r) \cap X_2 = \emptyset\},\ X_2^* = \{x \in X_2 | B(x,r) \cap X_1 = \emptyset\} \text{ and } X^* = X_1^* \cup X_2^*. \tag{3.7}$$

In the first part of the proof (see Section 3.5.4), we show that the graphs of X_1^* and X_2^* (with weights $\mathbf{W}$) are respectively connected. If we can show that the graphs of X_1^* and X_2^* are disconnected from each other, then the proof can be concluded. To this end, the subsequent auxiliary sets $\hat{X}_1$ and $\hat{X}_2$ will be instrumental in the proof. We fix a constant δ (to be specified later in (3.34)), which depends on r, η and the angles of intersection of S_1 and S_2, and define

$$\hat{X}_1 = \{x \in X_1 | \operatorname{dist}_g(x, S_2) \geq \delta\},\ \hat{X}_2 = \{x \in X_2 | \operatorname{dist}_g(x, S_1) \geq \delta\} \text{ and } \hat{X} = \hat{X}_1 \cup \hat{X}_2. \tag{3.8}$$

We will verify that $X_1^* \subset \hat{X}_1$ and $X_2^* \subset \hat{X}_2$. In fact, it will be a consequence of the second part of the proof. This part shows that the graph of $\hat{X}^c$ is disconnected from the graph of X_1^* as well as graph of X_2^*. Therefore, X_1^* and X_2^* cannot be connected via points in $\hat{X}^c$. At last, we show that they also cannot be connected within $\hat{X}$. That is, we show in the third part of the proof (Section 3.5.6) that the graphs of $\hat{X}_1$ and $\hat{X}_2$ are disconnected from each other. These three parts imply that the graphs of X_1^* and X_2^* form two connected components within X^*. By definition, X_1^* and X_2^* are identified with S_1 and S_2 respectively. To conclude the proof (for the noiseless case), we estimate the measure of the set X^{*c}, which was excluded. More precisely, we consider the measure of the set $X_{S_1 \cap S_2} \supset X^{*c}$, which we define as follows

$$X_{S_1 \cap S_2} = \{x \in X_1 | \operatorname{dist}_g(x, S_2) < r\} \cup \{x \in X_2 | \operatorname{dist}_g(x, S_1) < r\}. \tag{3.9}$$

This measure estimate and the conclusion of the proof (to the noiseless case) are established in Section 3.5.7. Section 3.5.8 discusses the generalization of the proof to the noisy case.

Various ideas of the proof follow [24], which considered multi-manifold modeling in Euclidean spaces. Some of the arguments in the proof of [24] even apply to general metric spaces, in particular, to Riemannian manifolds. We thus tried to maintain the notation of [24].

However, the algorithm construction and the main theoretical analysis of [24] are valid only when the dataset X lies in a Euclidean space and it is nontrivial to extend them to a Riemannian manifold. Indeed, the basic idea of [24] is to compare local covariance matrices and use this comparison to infer the relation between the corresponding data points, over which those matrices were generated. However, comparing local covariance matrices in the case where the ambient space is a Riemannian manifold is not straightforward as in Euclidean spaces. This is due to the fact that local covariance matrices are computed at different tangent spaces with different coordinate systems. Instead we show that it is sufficient to compare the "local directional information" (i.e., empirical geodesic angles) and "local dimension". Both of these quantities are derived from the local covariance matrices but are independent of the change of coordinates. Furthermore, the Riemannian setting requires local application of the nonlinear logarithm map, which distorts the uniform assumption within the manifold. Therefore, special care must be taken in using the logarithm map.

3.5.1 Notation

We provide additional notation to the one in Section 3.2.2. Readers are referred to [77] for a complete introduction to Riemannian geometry.

Let $B(x, r)$ and $B_x(\mathbf{0}, r)$ denote the r-neighborhoods of x and $\mathbf{0}$ in M and T_xM respectively. They are related by the exponential map, Φ_x, as follows: $B(x, r) = \Phi_x(B_x(\mathbf{0}, r))$. We refer to the coordinates obtained in the tangent space by the exponential map Φ as normal coordinates. Using normal coordinates, $B_x(\mathbf{0}, r) \subset T_xM$

is endowed with the Riemannian metric dist_g and measure μ_g. On the other hand, the tangent space T_xM can also be identified with $\mathbb{R}^D$ by choosing an orthonormal basis. This provides Euclidean metric dist_E and measure μ_E on T_xM, in particular, on $B_x(\mathbf{0}, r)$. There is a simple relation between μ_E and μ_g [77]:

$$\mu_g(d\mathbf{y}) = \mu_E(d\mathbf{y}) + \mathcal{O}(r^2)d\mathbf{y} \qquad \text{for} \quad \mathbf{y} \in B_x(\mathbf{0}, r). \tag{3.10}$$

Figure 3.14 highlights the difference between dist_E and dist_g. It shows the tangent space $T_n\mathbb{S}^2$ of the north pole, n, of $\mathbb{S}^2$ and the straight line (blue) connecting $\Phi_n^{-1}(x)$ and $\Phi_n^{-1}(y)$ in T_n; it is the shortest path w.r.t. dist_E. On the other hand, the shortest path w.r.t. dist_g is clearly the equator (the geodesic connecting x and y), which is the arc (black) on $T_n\mathbb{S}^2$; it is different than the straight line. In fact, only lines in $T_n\mathbb{S}^2$ connecting the origin and other points on $T_n\mathbb{S}^2$ correspond to geodesics on $\mathbb{S}^2$ for a general metric. As a consequence, the measures μ_g and μ_E induced by dist_g and dist_E are also different.

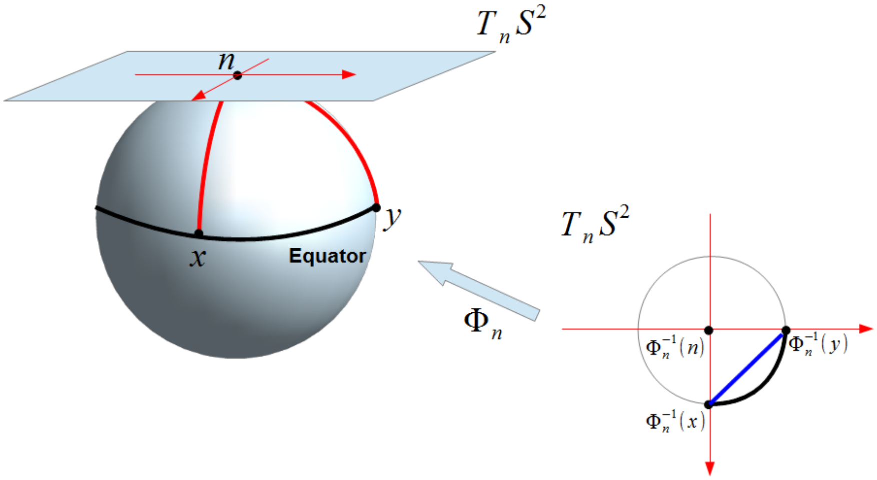


Figure 3.14: Difference between the metrics dist_g and dist_E on $T_n\mathbb{S}^2$. The arc in $T_n\mathbb{S}^2$ is a geodesic under the metric g. The line segment in $T_n\mathbb{S}^2$ is a geodesic under the Euclidean metric dist_E.

Given a submanifold $S \subset M$ (or a τ-tubular neighborhood S^τ of S), the metric tensor on S (or S^τ) inherited from g induces a measure μ_{gS} on S (or S^τ), which is called the uniform measure on S (or S^τ). For simplicity we assume throughout most of the proof that $\tau = 0$ and thus mainly discuss the measure μ_{gS} on S. In Section 3.5.8 we generalize the proof to the noisy case and thus discuss μ_{gS} on S^τ. The push-forward measure of μ_{gS} by Φ_x^{-1} is a measure on T_xM, which is again denoted by μ_{gS}. By definition, the support of the push-forward measure μ_{gS} is $\Phi_x^{-1}(S)$, which is a submanifold of $T_xM \equiv \mathbb{R}^D$. The Euclidean metric dist_E similarly induces another measure μ_{ES}, which is supported on $\Phi_x^{-1}(S)$.

For a measure μ on T_xM and a subset $H \subset T_xM$ of positive such measure, the expected covariance matrix $\mathbb{E}_\mu \mathbf{C}_H$ is defined by

$$\mathbb{E}_\mu \mathbf{C}_H = \frac{1}{\mu(H)} \int_{\mathbf{y}\in H} \mathbf{y}\mathbf{y}^T \mu(d\mathbf{y}) - \frac{1}{(\mu(H))^2} \int_{\mathbf{y}\in H} \mathbf{y}\mu(d\mathbf{y}) \cdot \int_{\mathbf{y}\in H} \mathbf{y}^T \mu(d\mathbf{y}). \tag{3.11}$$

For the two compact submanifolds of the model, S_1 and S_2, we denote $S = S_1 \cup S_2$ and define the following two measures w.r.t. S: $\mu_{gS} = \mu_{gS_1} + \mu_{gS_2}$ and $\mu_{ES} = \mu_{ES_1} + \mu_{ES_2}$. The covariance matrices w.r.t. μ_{gS} and μ_{ES} are denoted by $\mathbb{E}_{\mu_{gS}} \mathbf{C}_H$ and $\mathbb{E}_{\mu_{ES}} \mathbf{C}_H$, respectively. For simplicity, when $H = B_x(\mathbf{0}, r) \subset T_xM$, we denote them by $\mathbb{E}_{\mu_{gS}} \mathbf{C}_x$ and $\mathbb{E}_{\mu_{ES}} \mathbf{C}_x$. For $H = \Phi_z^{-1}(B(x,r)) \subset T_zM$, we denote them by $\mathbb{E}_{\mu_{gS}} \mathbf{C}_x^z$ and $\mathbb{E}_{\mu_{ES}} \mathbf{C}_x^z$.

If a dataset $X \in M$ is given, let $\mathbf{C}_{x_0}$ denote the sample covariance of the data $\Phi_{x_0}^{-1}(B(x_0,r) \cap X)$ on $T_{x_0}M$ and $\mathbf{C}_{x_0}^z$ denote the sample covariance of $\Phi_z^{-1}(B(x_0,r) \cap X)$ on T_zM. Let $\theta_{\min}(T_zS_1, T_zS_2)$ denote the minimal nonzero principal angle between the subspaces $T_zS_1, T_zS_2 \subset T_zM$[3] and let

$$\theta_0(S_1, S_2) = \inf_{z \in S_1 \cap S_2} \theta_{\min}(T_zS_1, T_zS_2). \tag{3.12}$$

For $x \in S_1 \cap S_2$, let $\theta_{\max}(T_xS_1, T_xS_2)$ denote the largest principal angle between T_xS_1 and T_xS_2 and let

$$\theta_{\max}(S_1, S_2) = \min_{x \in S_1 \cap S_2} \theta_{\max}(T_xS_1, T_xS_2). \tag{3.13}$$

Recall that the notation

$$Q_1(r) = Q_2(r) + \mathcal{O}(r^n)$$

means that there is a constant C independent of r such that

$$|Q_1(r) - Q_2(r)| \leq Cr^n. \tag{3.14}$$

[3] We only use $\theta_{\min}(T_zS_1, T_zS_2)$ when there is a nonzero principal angle. It is thus well-defined.

If $Q_1(r)$ and $Q_2(r)$ are matrices, then (3.14) applies to their entries. If $x_i, x_j \in M$, we denote by $l'(x_i, x_j)$ the tangent vector of $l(x_i, x_j)$ at x_i (it was denoted by $\mathbf{v}_{ij}$ in Section 3.2.2). We denote the empirical geodesic angle between x and y by $\theta_{x,y}$ (where for data points x_i, x_j, $\theta_{x_i,x_j} = \theta_{ij}$). Lastly, for a matrix $\mathbf{C}$, $\lambda_k(\mathbf{C})$ stands for the kth largest eigenvalue of $\mathbf{C}$.

3.5.2 A Generative Multi-Geodesic Model

We review in more details the generative model for two geodesic submanifolds (see Section 3.2.1). We first state the definition of geodesic submanifolds.

Definition 3.5.1. *For a Riemannian manifold M, a submanifold S is called a geodesic submanifold if $\forall x, y \in S$, the shortest geodesic connecting x and y in M is also contained in S.*

Let S_1, S_2 be two compact geodesic submanifolds of dimension d in a Riemannian manifold (M, g) and recall that $S = S_1 \cup S_2$. Let S_1^τ, S_2^τ and S^τ denote τ-tubular neighborhoods of S_1, S_2 and S respectively. For example, $S_1^\tau = \{x \in M \ : \mathrm{dist}_g(x, S_1) \leq \tau\}$, where $\mathrm{dist}_g(x, S_1) := \min_{y \in S_1} \mathrm{dist}_g(x, y)$. The dataset X of size N is i.i.d. sampled from the normalized version of μ_{gS} (by $\mu_{gS}(S^\tau)$) on S^τ. We recall the notation: $X_1 = S_1 \cap X$ and $X_2 = S_2 \cap X$. Fixing a point x_i, then $\mathbf{x}_j^{(i)}$ is i.i.d. sampled from the normalized push-forward μ_{gS} on $T_{x_i}M$.

3.5.3 Local Concentration with High Probability

We verify here the concentration of the local covariance matrices and the existence of sufficiently large samples in local neighborhoods from the same submanifold. We follow [24][4] and define on the probability space S^N (i.e., $(S_1 \cup S_2) \times \cdots \times (S_1 \cup S_2)$), the following events Ω_1 and Ω_2:

$$\Omega_1 = \bigcap_{k=1}^{2} \{X = (x_1, \ldots, x_N) \in S^N : \#\{i : x_i \in S_k \cap B(y, r/C_\Omega)\} > nr^d/C_7, \forall y \in S_k\}, \tag{3.15}$$

$$\Omega_2 = \{X = (x_1, \ldots, x_N) \in S^N : \|\mathbf{C}_{x_i} - \mathbb{E}_{\mu_{gS}} \mathbf{C}_{x_i}\| \leq r^3,\ i = 1, \ldots, N\}, \tag{3.16}$$

[4] For simplicity, we set the parameter t of [24] to be equal to r

where C_Ω and C_7 are specified in [24] (C_Ω depends on d and θ_0 (defined in (3.12)) and C_7 depends on the covering number of S) and $\mathbf{C}_{x_i}$ is the sample covariance of images $\{\mathbf{x}_j^{(i)}\}_{j \in J(x_i,r)}$ on $T_{x_i}M$. We note that Ω_1 is the set of datasets of N samples, where each dataset satisfies the following condition: for any point in S_i ($i = 1, 2$ is fixed), there are enough samples that also belong to S_i (their fraction is proportional to r^d). The set Ω_2 is the set of datasets of N samples with sufficient concentration of local covariance matrices. The following theorem of [24, page 35] ensures that the event $\Omega = \Omega_1 \cap \Omega_2$ is large. It uses the constant $C_0 = 4d + 2C_7$ and an absolute constant C_0'.

Theorem 3.5.2. *Let $\Omega = \Omega_1 \cap \Omega_2$. Then,*

$$\mathbb{P}(\Omega^c) \leq C_0 \cdot N e^{-Nr^{d+2}/C_0'}.$$

In view of this theorem, we assume in the rest of the proof that

$$X \in \Omega. \tag{3.17}$$

3.5.4 Ensuring Connectedness of X_1^* and X_2^*

The following proposition establishes WLOG the connectedness of the graph of the set X_1^* (defined in (3.7)). It uses a constant C_1, which is clarified in the proof and depends on geometric properties of S_1 and S_2 and their angle of intersection.

Proposition 3.5.3. *There exists a constant $C_1 > 1$ such that if*

$$r < \frac{\min(\eta, \sigma_a, \sigma_d)}{C_1}, \tag{3.18}$$

then the graph with nodes at X_1^* and edges given by $\mathbf{W}$ is connected.

Proof of Proposition 3.5.3

Three different constants C_8, C_9 and C_{10} appear in the proof. As clarified below, they depend on geometric properties of S_1 and S_2 and their angle of intersection. The constant C_1 is then determined by these constants as follows: $C_1 = \max(\{C_i\}_{i=8}^{10})$.

The proof is divided into three parts. The first one shows that $\mathbf{1}_{\dim(T^E_{x_i}S)=\dim(T^E_{x_j}S)} = 1$ for all x_i, x_j in X_1^* if $r < \eta/C_8$. The second one shows that $\mathbf{1}_{(\theta_{ij}+\theta_{ji})<\sigma_a} = 1$ for all x_i, x_j in X_1^* if $\sigma_a \geq C_9 r$. The last one uses an argument of [24, page 38]. It claims that the graph with nodes at X_1^* and weights given by the indicator function $\mathbf{1}_{\mathrm{dist}_g(x_i,x_j)<\sigma_d}$ is connected if $r \leq \sigma_d/C_{10}$.

Part I: We prove the following lemma, which clearly implies that $\mathbf{1}_{\dim(T^E_{x_i}S)=\dim(T^E_{x_j}S)} = 1$ for $x_i, x_j \in X_1^*$.

Lemma 3.5.4. *There exists a constant $C_8 > 1$ such that if $x_0 \in X_1^*$, $r < \eta/C_8$ and $0 < \eta < 1$, then*

$$\dim(T^E_{x_0}S) = \dim(T_{x_0}S). \tag{3.19}$$

Proof. Recall that $\mathbf{C}_{x_0}$ denotes the sample covariance of the transformed data $\Phi_x^{-1}(X) \cap B_{x_0}(\mathbf{0}, r)$. We denote $H = B_{x_0}(\mathbf{0}, r) \cap T_{x_0}S$ and note that

$$\begin{aligned}
&\mathbb{E}_{\mu_{gS}}\mathbf{C}_{x_0} = \frac{1}{\mu_{gS}(H)}\int_H \mathbf{y}\mathbf{y}^T \mu_S(d\mathbf{y}) - \frac{1}{(\mu_{gS}(H))^2}\int_H \mathbf{y}\mu_{gS}(d\mathbf{y}) \cdot \int_H \mathbf{y}^T \mu_S(d\mathbf{y}) \\
&= \frac{1}{\mu_{ES}(H)}\int_H \mathbf{y}\mathbf{y}^T \mu_{IS}(d\mathbf{y}) - \frac{1}{(\mu_{ES}(H))^2}\int_H \mathbf{y}\mu_{ES}(d\mathbf{y}) \cdot \int_H \mathbf{y}^T \mu_{IS}(d\mathbf{y}) + \mathcal{O}(r^4) \\
&= \mathbb{E}_{\mu_{ES}}\mathbf{C}_{x_0} + \mathcal{O}(r^4).
\end{aligned} \tag{3.20}$$

The first and third equalities of (3.20) follow from the definition of the expected covariance. The second equality of (3.20) follows from (3.10) and the fact that $\|\mathbf{y}\| \le r$. A slight generalization of Lemma 11 of [24] implies that

$$\mathbb{E}_{\mu_{ES}}\mathbf{C}_{x_0} = \frac{r^2}{d+2}\mathbf{P}_{T_{x_0}S}, \tag{3.21}$$

where $\mathbf{P}_{T_{x_0}S}$ is the orthogonal projector onto $T_{x_0}S$ in $T_{x_0}M$. Equation (3.20) and (3.21) imply that

$$\|\mathbb{E}_{\mu_{gS}}\mathbf{C}_{x_0} - \frac{r^2}{d+2}\mathbf{P}_{T_{x_0}S}\| < C_S r^4, \tag{3.22}$$

where $C_S > 0$ is a constant depending on the Riemannian metric g (arising due to (3.10)). Using this constant C_S, we define

$$C_8 = 2(d+2)(C_S+1). \tag{3.23}$$

We note that $C_8 > 1$. Combining this observation with the following two assumptions: $r < \eta/C_8$ and $0 < \eta < 1$, we conclude that $r < 1$.

Combining the triangle inequality, (3.17), (3.22) and the fact that $r < 1$, we conclude that

$$\|\mathbf{C}_{x_0} - \frac{r^2}{d+2}\mathbf{P}_{T_{x_0}S}\| \le \|\mathbf{C}_{x_0} - \mathbb{E}_{\mu_{gS}}\mathbf{C}_{x_0}\| + \|\mathbb{E}_{\mu_{gS}}\mathbf{C}_{x_0} - \frac{r^2}{d+2}\mathbf{P}_{T_{x_0}S}\| < (C_S+1)r^3. \tag{3.24}$$

The application of both Weyl's inequality [88] and (3.24) results in the following lower bound of $\lambda_1(\mathbf{C}_{x_0})$ and upper bound of $\lambda_{d+1}(\mathbf{C}_{x_0})$:

$$\lambda_{d+1}(\mathbf{C}_{x_0}) < (C_S+1)r^3 \quad \text{and} \quad \lambda_1(\mathbf{C}_{x_0}) > \frac{r^2}{d+2} - (C_S+1)r^3. \tag{3.25}$$

It follows from (3.23), (3.25) and elementary algebraic manipulations that

$$\begin{aligned} \frac{\lambda_{d+1}(\mathbf{C}_{x_0})}{\lambda_1(\mathbf{C}_{x_0})} &< \frac{(C_S+1)r^3}{\frac{r^2}{d+2} - (C_S+1)r^3} = \frac{C_S+1}{1/(r(d+2)) - (C_S+1)} \\ &< \frac{C_S+1}{C_8/(d\eta+2\eta) - (C_S+1)} = \frac{\eta}{2-\eta} < \eta. \end{aligned} \tag{3.26}$$

Equation (3.19) thus follows from (3.26) and the thresholding of eigenvalues by $\eta\|\mathbf{C}_{x_0}\|$ in Algorithm 1. □

Part II: Next, we prove that $\mathbf{1}_{(\theta_{ij}+\theta_{ji})<\sigma_a} = 1$ if $\sigma_a \geq C_9 r$.

Lemma 3.5.5. *There exists a constant $C_9 > 1$ such that if $x, y \in X_1^*$ and*

$$\sigma_a \geq C_9 r, \tag{3.27}$$

then $\mathbf{1}_{(\theta_{x,y}+\theta_{y,x})<\sigma_a} = 1$.

Proof. We define

$$C_9 = \sqrt{2}(C_S+1)(d+2)\pi, \tag{3.28}$$

where C_S is the constant introduced in (3.22). We show that for $x, y \in X_1^*$:

$$\theta_{x,y} < \frac{C_9}{2} r, \tag{3.29}$$

which immediately implies the lemma.

In order to prove (3.29), we first apply the Davis-Kahan Theorem [89] and (3.19) and then apply (3.24) to obtain the following bound on the distance between the subspaces $T_x^E S$ and $T_x S$ (which are spanned by the top d eigenvectors of $\mathbf{C}_x$ and $\frac{r^2}{d+2}\mathbf{P}_{T_x S}$, respectively; this observation uses (3.19)):

$$\|\mathbf{P}_{T_x^E S} - \mathbf{P}_{T_x S}\| < \frac{\sqrt{2}\|\mathbf{C}_x - \frac{r^2}{d+2}\mathbf{P}_{T_x S}\|}{\frac{r^2}{d+2}} < \sqrt{2}(C_S+1)(d+2)r. \tag{3.30}$$

We remark that in applying the Davis-Kahan Theorem we made use of the following basic calculation of Δ, the dth spectral gap of $\frac{r^2}{d+2}\mathbf{P}_{T_xS}$: $\Delta = \lambda_d(\frac{r^2}{d+2}\mathbf{P}_{T_xS}) - \lambda_{d+1}(\frac{r^2}{d+2}\mathbf{P}_{T_xS}) = \frac{r^2}{d+2}$.

Next, we recall that $\theta_{\max}(T_x^E S, T_x S)$ denotes the largest principal angle between $T_x^E S$ and $T_x S$. We note that Lemma 15 of [24] (whose application requires (3.19)), (3.30), (3.28) and Jordan's inequality (lower bounding the sin function by $2/\pi$) imply that

$$\theta_{\max}(T_x^E S, T_x S) = \sin^{-1}(\|\mathbf{P}_{T_x^E S} - \mathbf{P}_{T_x S}\|) < \sin^{-1}(\sqrt{2}(C_S + 1)(d+2)r) < \frac{C_9}{2} r. \tag{3.31}$$

Since $\theta_{x,y}$ is the angle between $l'(x,y) \in T_x S$ and $T_x^E S$, (3.29) follows from (3.31). We can then conclude that if $\sigma_a \geq C_9 r$, then $\mathbf{1}_{(\theta_{ij}+\theta_{ji})<\sigma_a} = 1$ for all x_i, x_j in X_1^*.

□

Part III: By the construction of the affinity matrix $\mathbf{W}$ and Lemmata 3.5.4 and 3.5.5, the connectivity between points $x_i, x_j \in X_1^*$ is solely determined by the indicator function $\mathbf{1}_{\mathrm{dist}_g(x_i,x_j)<\sigma_d}$. It is obvious that if $\sigma_d > 4r$ then the graph with nodes in X_1 and weights $\mathbf{1}_{\mathrm{dist}_g(x_i,x_j)<\sigma_d}$ is connected (this can be done by finite covering of S_1 with balls of radius r). It follows from [24, pages 38-39] that the graph with nodes in X_1^* is also connected if

$$r \leq \sigma_d / C_{10}. \tag{3.32}$$

There is one component in the argument of [24, page 38] that requires careful adaptation to the Riemannian case. It is related to the determination of the constant C_{10}. This constant is set to be $(3C' + 9)^{-1}$ (see [24, page 39]). In the Euclidean case, C' is guaranteed by Lemma 18 of [24]. The adaptation of this Lemma to the Riemannian case can be stated in the following lemma (it uses θ_0, which was defined in (3.12)).

Lemma 3.5.6. *Let (M, g) be a Riemannian manifold and S_1, S_2 be two compact geodesic submanifolds of dimension d such that $\theta_0(S_1, S_2) > 0$. Then there is a constant C' such that*

$$\mathrm{dist}_g(x, S_1 \cap S_2) \leq C' \max\{\mathrm{dist}_g(x, S_1), \mathrm{dist}_g(x, S_2)\} \quad \forall x \in S_1 \cup S_2.$$

We prove Lemma 3.5.6 in Appendix 3.10.1. The proof implies that C' is determined by the geometric properties of S_1 and S_2 and the angle $\theta_0(S_1, S_2)$.

3.5.5 Disconnectedness Between $\hat{X}^c$ and X_1^* (or X_2^*)

We show here that the points in $\hat{X}^c$ (where $\hat{X}$ is defined in (3.8)) are not connected to the points of X^*. In Section 3.5.4, we showed that the estimated dimensions of local neighborhoods of points in X^* equal d. In this section, we show that the estimated dimensions of local neighborhoods of points in $\hat{X}^c$ are larger than d. Since $\mathbf{1}_{\dim(T^E_{x_i}S)=\dim(T^E_{x_j}S)}$ is a multiplicative term of $\mathbf{W}$, we conclude that $\hat{X}^c$ is disconnected from X^*. The following main proposition of this section implies that WLOG the estimated tangent dimension at $\hat{X}^c \cap X_1$ is at least $d+1$ (it uses the angle $\theta_{\max}(S_1, S_2)$ defined in (3.13)).

Proposition 3.5.7. *There exists a constant $C_2 > 1$ depending only on d and $\theta_{\max}(S_1, S_2)$ such that if $r < \eta$,*

$$\eta < C_2^{-\frac{d+2}{2}}, \tag{3.33}$$

$$\delta := r\sqrt{1 - C_2 \eta^{\frac{2}{d+2}}} \tag{3.34}$$

and

$$x \in \hat{X}^c \cap X_1, \text{ that is, } \operatorname{dist}_g(x, S_2) < \delta, \tag{3.35}$$

then

$$\frac{\lambda_{d+1}(\mathbf{C}_x)}{\lambda_1(\mathbf{C}_x)} > \eta.$$

Proof. Let us first sketch the idea of the proof. It is easier to estimate the local covariance matrices when the two manifolds are subspaces (see Lemma 21 of [24]). However, for $x \in \hat{X}^c \cap X_1$, the logarithm map of S into T_xM does not result in two subspaces (see Figure 3.15). On the other hand, for z, the projection of x onto $S_1 \cap S_2$, the logarithm map of S into T_zM results in two subspaces, where the local covariance can be estimated more easily. Some difficulties arise due to the application of the logarithm map and the change of tangent spaces. In particular, the ball $B(x, r)$ becomes irregular in the domain T_zM.

We recall that $\mu_{gS} = \mu_{gS_1} + \mu_{gS_2}$ and $\mu_{ES} = \mu_{ES_1} + \mu_{ES_2}$. We arbitrarily fix $x_0 \in S_1$ such that $\operatorname{dist}_g(x_0, S_2) < r$. We note that Lemma 3.5.6 implies that

$$\operatorname{dist}_g(x_0, S_1 \cap S_2) \leq Cr. \tag{3.36}$$

Let

$$z = \operatorname{argmin}_{y \in S_1 \cap S_2} \operatorname{dist}_g(x_0, y),$$

where if argmin is not uniquely defined, then z is arbitrarily chosen among all minimizers. It follows from (3.36) that $\operatorname{dist}_g(x_0, z) \leq Cr$ and from this and the triangle inequality, it follows that

$$B(x_0, r) \subset B(z, (C+1)r). \tag{3.37}$$

Recall that Φ_{x_0} and Φ_z denote the normal coordinate charts around x_0 and z respectively (see Figure 3.15); it is sufficient to restrict them to $B(x_0, r)$ and $B(z, (C+1)r)$ respectively. When using the chart Φ_z, S_1 and S_2 correspond to two subspaces in T_zM, which we denote by L_1 and L_2 respectively. On the other hand, when using the chart Φ_{x_0}, S_2 corresponds to a manifold in $T_{x_0}M$, whereas S_1 still corresponds to a subspace. It follows from (3.37) and the invertibility of Φ_z that the composition map $\phi = \Phi_z^{-1} \circ \Phi_{x_0}$ embeds $B_{x_0}(\mathbf{0}, r)$ into $B_z(\mathbf{0}, (C+1)r)$ as shown in Figure 3.15. Recall that $\mathbf{C}_{x_0}$ denotes

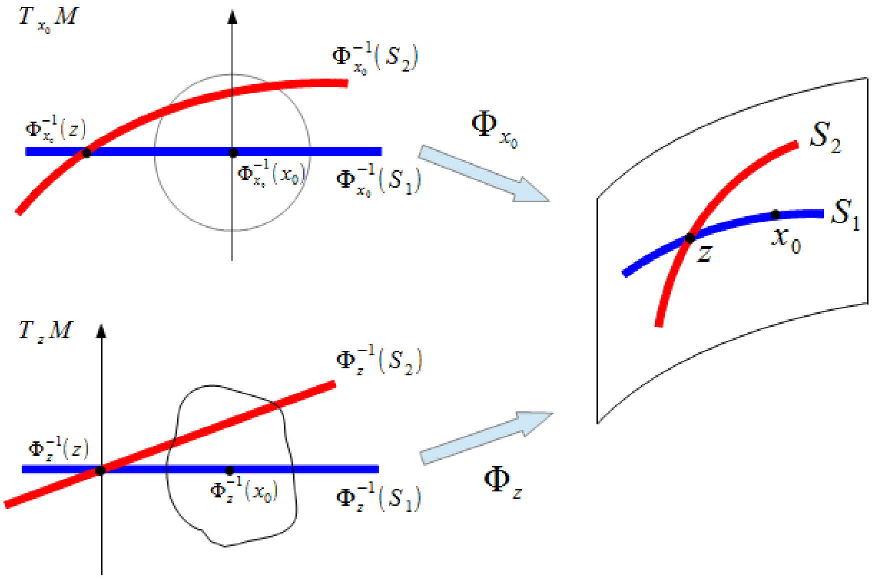


Figure 3.15: The transition map between normal coordinates of $T_{x_0}M$ and T_zM. Notice the regular ball $B_{x_0}(\mathbf{0}, r)$ in $T_{x_0}M$ is mapped to the irregular region in T_zM because the exponential maps Φ_{x_0} and Φ_z are nonlinear.

the sample covariance of the data $\Phi_{x_0}^{-1}(B(x_0, r) \cap X)$ in $T_{x_0}M$ and $\mathbf{C}_{x_0}^z$ denotes the sample covariance of the data $\Phi_z^{-1}(B(x_0, r) \cap X)$ in T_zM. Using the notation $O(D)$ for the set of orthogonal $D \times D$ matrices, we claim that

$$\exists \mathbf{R} \in O(D) \quad \text{s.t.} \quad \mathbf{R} \left(\mathbb{E}_{\mu_{gS}} \mathbf{C}_{x_0} \right) \mathbf{R}^T = \mathbb{E}_{\mu_{gS}} \mathbf{C}_{x_0}^z + \mathcal{O}(r^3). \tag{3.38}$$

The technical proof of (3.38) is in Appendix 3.10.2.

We estimate $\mathbb{E}_{\mu_{gS}} \mathbf{C}_{x_0}^z$ as follows. Let $H = \Phi_z^{-1}(B(x_0, r) \cap (S_1 \cup S_2))$ and $H' = B_I(\Phi_z^{-1}(x_0), r) \cap (L_1 \cup L_2)$ (see Figure 3.16), where $B_I(\Phi_z^{-1}(x_0), r)$ is the r-ball with center $\Phi_z^{-1}(x_0)$ in T_zM, which uses the Euclidean distance dist_E.

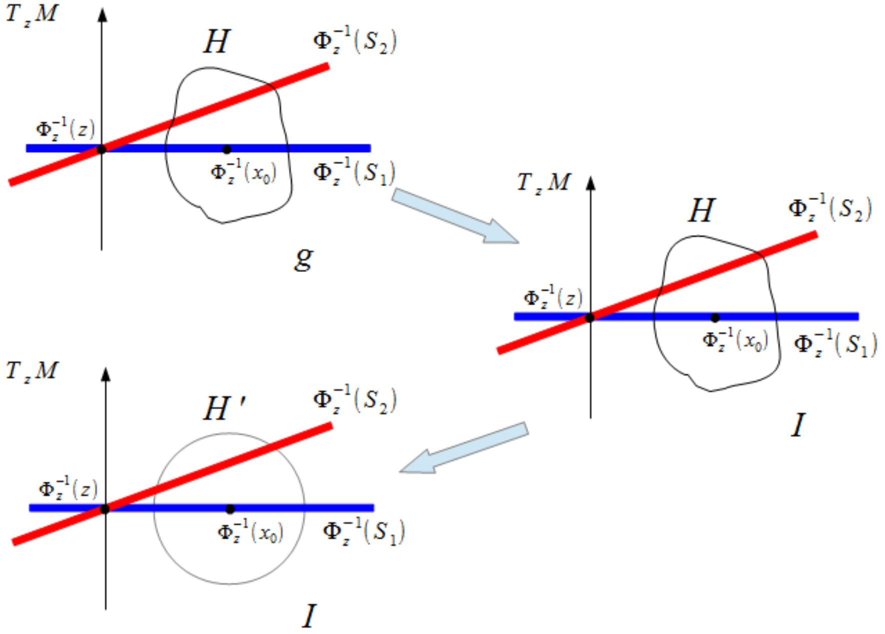


Figure 3.16: Change of domain and metric

The rest of the proof requires the following two technical observations

$$\mu_{ES}((H \setminus H') \cup (H' \setminus H)) = \mathcal{O}(r)\mu_{ES}(H). \tag{3.39}$$

and

$$\mathbb{E}_{\mu_{gS}} \mathbf{C}_{x_0}^z = \mathbb{E}_{\mu_{ES}} \mathbf{C}_{x_0}^z + \mathcal{O}(r^3) = \mathbb{E}_{\mu_{ES}} \mathbf{C}_{H'} + \mathcal{O}(r^3). \tag{3.40}$$

We prove (3.39) in Appendix 3.10.3. The first equality of (3.40) follows from the definition of the expected covariance (see (3.11)), (3.10) and the fact that $\|\mathbf{y}\| \leq (C+1)r$. The second equality of (3.40) follows from the definition of the expected covariance (see (3.11)), (3.39) and the fact that $\|\mathbf{y}\| \leq (C+1)r$.

It follows from (3.16), (3.17), (3.38), (3.40) and the triangle inequality that

$$\begin{aligned}\|\mathbf{R}\mathbf{C}_{x_0}\mathbf{R}^T - \mathbb{E}_{\mu_{ES}}\mathbf{C}_{H'}\| \leq & \|\mathbf{R}\mathbf{C}_{x_0}\mathbf{R}^T - \mathbf{R}\mathbb{E}_{\mu_{gS}}\mathbf{C}_{x_0}\mathbf{R}^T\| + \|\mathbf{R}\mathbb{E}_{\mu_{gS}}\mathbf{C}_{x_0}\mathbf{R}^T - \mathbb{E}_{\mu_{ES}}\mathbf{C}^z_{x_0}\| + \\ & \|\mathbb{E}_{\mu_{ES}}\mathbf{C}^z_{x_0} - \mathbb{E}_{\mu_{ES}}\mathbf{C}_{H'}\| \leq r^3 + \mathcal{O}(r^3) + \mathcal{O}(r^3) \leq C'_S r^3\end{aligned} \tag{3.41}$$

for a constant $C'_S > 0$.

The combination of (3.41), Weyl's inequality [88] for $\mathbf{R}(\mathbf{C}_{x_0})\mathbf{R}^T$ and $\mathbb{E}_{\mu_{ES}}\mathbf{C}_{H'}$, and the fact that $\mathbf{R}\mathbf{C}_{x_0}\mathbf{R}^T$ and $\mathbf{C}_{x_0}$ have the same eigenvalues implies that

$$\lambda_{d+1}(\mathbf{C}_{x_0}) \geq \lambda_{d+1}(\mathbb{E}_{\mu_{ES}}\mathbf{C}_{H'}) - C'_S r^3, \quad \lambda_1(\mathbf{C}_{x_0}) \leq \lambda_1(\mathbb{E}_{\mu_{ES}}\mathbf{C}_{H'}) + C'_S r^3. \tag{3.42}$$

Notice that $\theta_{\max}(S_1, S_2) \leq \theta_{\max}(L_1, L_2)$ by definition. Applying (3.42) and Lemma 21 of [24] to $\mathbb{E}_{\mu_{ES}}\mathbf{C}_{H'}$, where $\theta_{\max}(L_1, L_2)$ is replaced by $\theta_{\max}(S_1, S_2)$ and proper scaling is used, results in

$$\begin{aligned}\frac{\lambda_{d+1}(\mathbf{C}_{x_0})}{\lambda_1(\mathbf{C}_{x_0})} &\geq \frac{\frac{1}{8(d+2)}(1-\cos\theta_{\max}(S_1,S_2))^2(1-(\mathrm{dist}_g(x_0,S_2)/r)^2)_+^{d/2+1} - C'_S r^3}{1/(d+2) + (\mathrm{dist}_g(x_0,S_2)/r)(1-(\mathrm{dist}_g(x_0,S_2)/r)^2)_+^{d/2} + C'_S r^3} \\ &\geq \frac{\frac{1}{8(d+2)}(1-\cos\theta_{\max}(S_1,S_2))^2(1-(\mathrm{dist}_g(x_0,S_2)/r)^2)^{d/2+1} - C'_S r^3}{1/(d+2) + 1 + C'_S r^3}.\end{aligned} \tag{3.43}$$

We remark that the second inequality of (3.43) is derived by applying the bound: $\mathrm{dist}_g(x, S_2) < r$. In order to satisfy $\frac{\lambda_{d+1}(\mathbf{C}_{x_0})}{\lambda_1(\mathbf{C}_{x_0})} > \eta$, we require that

$$\frac{\frac{1}{8(d+2)}(1-\cos\theta_{\max}(S_1,S_2))^2(1-(\mathrm{dist}_g(x_0,S_2)/r)^2)^{d/2+1} - C'_S r^3}{1/(d+2) + 1 + C'_S r^3} > \eta. \tag{3.44}$$

Since $r < \eta < 1$, we replace $C'_S r^3$ with $C'_S \eta$ in the numerator of (3.44) and $C'_S r^3$ with C'_S in the denominator of (3.44) and slightly simplify the inequality to obtain the following stronger requirement:

$$\frac{(1-\cos\theta_{\max}(S_1,S_2))^2}{8(d+2)}(1-(\mathrm{dist}_g(x_0,S_2)/r)^2)^{d/2+1} > \left(\frac{1}{d+2} + 1 + 2C'_S\right)\eta. \tag{3.45}$$

Finally, setting

$$C_2 = \left(\frac{8d + 24 + 16(d+2)C'_S}{(1 - \cos\theta_{\max}(S_1, S_2))^2}\right)^{\frac{2}{d+2}} \tag{3.46}$$

we can rewrite (3.45) as follows

$$(1 - (\mathrm{dist}_g(x_0, S_2)/r)^2)^{\frac{d+2}{2}} > C_2^{\frac{d+2}{2}}\eta. \tag{3.47}$$

We immediately conclude (3.47) (and consequently the lemma) from (3.34) and (3.35). □

We end this section with an immediate corollary of Proposition 3.5.7, which is crucial in order to follow the proof.

Corollary 3.5.8. *The following relations are satisfied:*

$$X_1^* \subset \hat{X}_1 \quad \textit{and} \quad X_2^* \subset \hat{X}_2.$$

Proof. It follows from Lemma 3.5.4 and Proposition 3.5.7 that $X_1^* \cap \hat{X}^c = \emptyset$. Therefore $X_1^* \subset \hat{X}_1$. Similarly, $X_2^* \subset \hat{X}_2$. □

3.5.6 The Disconnectedness of X_1^* and X_2^*

We show here that the graphs with nodes at X_1^* and X_2^* are disconnected. The idea is to show that the function $\mathbf{1}_{\mathrm{dist}_g(x_i,x_j)<\sigma_d}\mathbf{1}_{\theta_{ij}+\theta_{ji}<\sigma_a}$ (and thus the weight $\mathbf{W}$) is zero between two points in $\hat{X}_1 \supset X_1^*$ and $\hat{X}_2 \supset X_2^*$ for appropriate choice of constants. This and Proposition 3.5.7 imply that the graphs associated with X_1^* and X_2^* are disconnected. We first establish a lower bound on the empirical geodesic angle in Lemma 3.5.9 and then conclude that there is no direct connection between the sets $\hat{X}_1$ and $\hat{X}_2$ in Corollary 3.5.10.

Lemma 3.5.9. *There exist constants $C_3 > 0$ and $C_4 > 0$ such that if $x_1 \in \hat{X}_1$, $x_2 \in \hat{X}_2$,*

$$\mathrm{dist}_g(x_1, x_2) < \sigma_d, \tag{3.48}$$

$$\textit{and} \ \ \sigma_d < C_4^{-\frac{1}{2}}, \tag{3.49}$$

then the angle between the estimated tangent subspace $T^E_{x_1}S_1$ and the line segment $l^{(1)}_{12}$, which connects the origin and $\mathbf{x}^{(1)}_2$ (the image of x_2 by $\log_{x_1}$) in $T_{x_1}M$ is bounded below as follows:

$$\angle(l^{(1)}_{12}, T^E_{x_1}S_1) > \min(\sin^{-1}(\delta/2\sigma_d) - C_3\eta^{d/(d+2)} - C_3r, \pi/6). \tag{3.50}$$

Proof. The proof develops various geometric estimates that eventually conclude (3.50). Let

$$\mathbf{x}_3 = \operatorname{argmin}_{\mathbf{x}\in T_{x_1}S_1} \operatorname{dist}_g(\mathbf{x}, \mathbf{x}^{(1)}_2) \quad \text{and} \quad \mathbf{x}_4 = \operatorname{argmin}_{\mathbf{x}\in T_{x_1}S_1} \operatorname{dist}_E(\mathbf{x}, \mathbf{x}^{(1)}_2),$$

where dist_E is defined with respect to the normal coordinate chart in T_xM (see Figure 3.17).

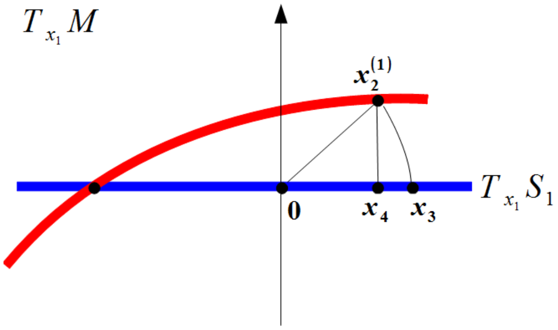


Figure 3.17: The normal coordinate chart at x_1

We note that by definition $\mathbf{x}_4$ is the projection of $\mathbf{x}^{(1)}_2$ onto $T_{x_1}S_1$ and thus

$$\operatorname{dist}_E(\mathbf{x}_4, \mathbf{0}) < \operatorname{dist}_g(\mathbf{x}^{(1)}_2, \mathbf{0}). \tag{3.51}$$

Combining (3.51) with the fact that dist_E and dist_g are the same on lines through the origin in $T_{x_1}M$ and then applying (3.48), we obtain that

$$\operatorname{dist}_g(\mathbf{x}_4, \mathbf{0}) < \operatorname{dist}_g(\mathbf{x}^{(1)}_2, \mathbf{0}) < \sigma_d. \tag{3.52}$$

Furthermore, combining the following two facts: $\mathbf{x}_3$ is a minimizer of $\operatorname{dist}_g(\cdot, \mathbf{x}^{(1)}_2) \in T_{x_1}S_1$ and $x_2 \in \hat{X}_2$, we obtain that

$$\delta \le \operatorname{dist}_g(x_2, \Phi_{x_1}(\mathbf{x}_3)) = \operatorname{dist}_g(\mathbf{x}^{(1)}_2, \mathbf{x}_3) < \operatorname{dist}_g(\mathbf{x}^{(1)}_2, \mathbf{x}_4). \tag{3.53}$$

We prove in Appendix 3.10.4 that there exists a constant $C_4 > 0$, which depends only on the Riemannian manifold M, such that

$$\forall R > 0,\ \mathbf{x}, \mathbf{y} \in B_{x_1}(\mathbf{0}, R), \quad |\operatorname{dist}_E(\mathbf{x}, \mathbf{y}) - \operatorname{dist}_g(\mathbf{x}, \mathbf{y})| < C_4 R^2 \operatorname{dist}_E(\mathbf{x}, \mathbf{y}). \tag{3.54}$$

Applying (3.54) (with $R = \sigma_d$) first and (3.53) next we obtain that

$$\begin{aligned} \operatorname{dist}_E(\mathbf{x}_2^{(1)}, \mathbf{x}_4) &> \operatorname{dist}_g(\mathbf{x}_2^{(1)}, \mathbf{x}_4) - C_4 \sigma_d^2 \operatorname{dist}_E(\mathbf{x}_2^{(1)}, \mathbf{x}_4) \\ &> \delta - C_4 \sigma_d^2 \operatorname{dist}_E(\mathbf{x}_2^{(1)}, \mathbf{x}_4) \end{aligned} \tag{3.55}$$

and consequently

$$\operatorname{dist}_E(\mathbf{x}_2^{(1)}, \mathbf{x}_4) > \frac{\delta}{1 + C_4 \sigma_d^2}. \tag{3.56}$$

It follows from (3.49) and (3.56) that

$$\sin(\angle(l_{12}^{(1)}, T_{x_1} S_1)) = \frac{\operatorname{dist}_E(\mathbf{x}_2^{(1)}, \mathbf{x}_4)}{\operatorname{dist}_E(\mathbf{x}_2^{(1)}, \mathbf{0})} > \frac{\delta}{\sigma_d + C_4 \sigma_d^3} > \delta / 2\sigma_d. \tag{3.57}$$

Our proof concludes from (3.57) and the following two claims:

$$\sin(\theta_{\max}(T_{x_1}^E S_1, T_{x_1} S_1)) \leq C_3' \eta^{d/(d+2)} + C_3' r \tag{3.58}$$

and

$$\angle(l_{12}^{(1)}, T_{x_1}^E S_1) \geq \min(\angle(l_{12}^{(1)}, T_{x_1} S_1) - \frac{2\pi\sqrt{d}}{3} \sin(\theta_{\max}(T_{x_1}^E S_1, T_{x_1} S_1)), \pi/6). \tag{3.59}$$

Inequalities (3.58) and (3.59) are verified in Appendices 3.10.5 and 3.10.6 respectively, where we also carefully analyze how the constant C_3' depends on the underlying Riemannian manifold (see (3.102)). Combining (3.57), (3.58) and (3.59), we conclude (3.50) by letting $C_3 = \dfrac{2\pi\sqrt{d}}{3} C_3'$.

□

The desired disconnectedness of X_1^* and X_2^* immediately follows from Lemma 3.5.9 in the following way:

Corollary 3.5.10. *The graphs with nodes at X_1^* and X_2^* respectively and weights in* $\mathbf{W}$ *are disconnected if the angle threshold σ_a is chosen such that*

$$\sigma_a < \min(\sin^{-1}(\delta / 2\sigma_d) - C_3 \eta^{d/(d+2)} - C_3 r, \pi/6) \tag{3.60}$$

and the distance threshold σ_d satisfies (3.49).

Proof. When σ_a and σ_d satisfy (3.60) and (3.49) respectively, Lemma 3.5.9 implies that if $x_i \in \hat{X}_1$ and $x_j \in \hat{X}_2$, then $\mathbf{1}_{\text{dist}_g(x_i,x_j)<\sigma_d}\mathbf{1}_{\theta_{ij}+\theta_{ji}<\sigma_a} = 0$. In other words, there is no direct connection between X_1^* and X_2^* through $\hat{X}$. On the other hand, Lemma 3.5.4 and Proposition 3.5.7 imply that X_1^* and X_2^* cannot be connected through points in $\hat{X}^c$ (since points in X^* and $\hat{X}^c$ have different local estimated dimensions). We thus conclude that X_1^* and X_2^* are disconnected. □

3.5.7 Conclusion of Theorem 3.3.1 for the Noiseless Multi-Geodesic Model

Due to Theorem 3.5.2 we replace X with $X \cap \Omega$ and obtain a statement for X with probability at least $1 - C_0 \cdot Ne^{-Nr^{d+2}/C_0'}$. Proposition 3.5.3 and Corollary 3.5.10 imply that (with probability at least $1-C_0 \cdot Ne^{-Nr^{d+2}/C_0'}$) X^* has two connected components. They require that the parameters of TGCT satisfy (3.18), (3.49) and (3.60). Additional requirement is specified in (3.33) (in Proposition 3.5.7 which implies Corollary 3.5.10). We also note that the requirement $r < \eta < 1$, which also appears in some of the auxiliary lemmata, follows from (3.18), (3.33) and the fact that $C_1 > 1$ and $C_2 > 1$. These requirements, i.e., (3.18), (3.33), (3.49) and (3.60), are sufficient and equivalent to (3.2) when $\tau = 0$.

Next, we explain why one can choose parameters that satisfy these requirements at the end of this section. The only problem is to make sure that the last inequality of (3.2) (equivalently, (3.60)) is satisfied. Given a sufficiently small $r > 0$ satisfying (3.18), we let $\sigma_d = \alpha r$ for some fixed $\alpha > 0$. The RHS of (3.60) tends to $\min(\sin^{-1}(\frac{1}{2\alpha}), \pi/6)$ as r and η approach zero. We note that the lower bound of σ_a is $C_1 r$. Therefore, if r and η are sufficiently small so that $\min(\sin^{-1}(1/2\alpha), \pi/6)/2$ is lower than the RHS of (3.60) and $C_1 r < \min(\sin^{-1}(1/2\alpha), \pi/6)/2$, then σ_a can be chosen from the interval $[C_1 r, \min(\sin^{-1}(1/2\alpha), \pi/6)/2]$.

In order to conclude the proof in this case we upper bound the expected portion of points $\#X^{*c}/\#X$, where $\#X^{*c}$ and $\#X$ denote he cardinality of X^{*c} and X respectively. For this purpose we use the set $X_{S_1 \cap S_2} \supset X^{*c}$, which was defined in (3.9) in the following way:

$$\mathbb{E}\left(\frac{\#X^{*c}}{\#X}\right) \leq \mathbb{E}\left(\frac{\#X_{S_1\cap S_2}}{\#X}\right) = \frac{\mu_{gS}(\{x \in S_1 | \operatorname{dist}_g(x, S_2) < r\})}{\mu_{gS}(S)} + \tag{3.61}$$
$$\frac{\mu_{gS}(\{x \in S_2 | \operatorname{dist}_g(x, S_1) < r\})}{\mu_{gS}(S)} \leq \frac{\mu_{gS}(\{x \in S_1 \cup S_2 | \operatorname{dist}_g(x, S_1 \cap S_2) \leq C'r\})}{\mu_{gS}(S)}$$
$$\leq C_6 r^{d-\dim(S_1\cap S_2)}.$$

The first equality of (3.61) follows from the fact that the dataset X is i.i.d. sampled from μ_{gS}. The second inequality of (3.61) follows from Lemma 3.5.6. The last one follows from Theorem 1.3 in [90], where C_6 is a constant depending only on the geometry of the underlying generative model (e.g., the mean curvature and volume of $S_1 \cap S_2$).

3.5.8 Conclusion of Theorem 3.3.1 for the Noisy Multi-Geodesic Model

The above analysis also applies when the generative multi-geodesic model has noise level τ and τ is sufficiently smaller than r, that is,

$$\tau < C_5 r, \tag{3.62}$$

where $C_5 \ll 1$. Indeed, in this case the estimates of tangent spaces and geodesics are sufficiently close to the estimates without noise. The only difference is that the last bound in (3.61) has to be replaced with $C_6(r+\tau)^{d-\dim(S_1\cap S_2)}$. This requires though a sufficiently small noise level (set by C_5). Precise bound on τ is not trivial. Furthermore, the analysis employed here is not optimal. We can thus only claim in theory robustness to very small levels of noise, whereas robustness to higher levels of noise is studied in the experiments.

3.6 Conclusions

Aiming at efficiently organizing data embedded in a non-Euclidean space according to low-dimensional structures, the present paper studied multi-manifold modeling in such spaces. The paper solves this clustering (or modeling) problem by proposing the novel GCT algorithm. GCT thoroughly exploits the geometry of the data to build a similarity matrix that can effectively cluster the data (via spectral clustering) even when the underlying submanifolds intersect or have different dimensions. In particular, it introduces the novel idea in non-Euclidean multi-manifold modeling of using directional information from local tangent spaces to avoid neighboring points of clusters different than that of the query point. Theoretical guarantees for successful clustering were established for a variant of GCT, namely TGCT for the MGM setting, which is a non-Euclidean generalization of the widely-used framework of hybrid-linear modeling. Unlike TGCT, GCT combined directional information from local tangent spaces with sparse coding, which aims to improve the clustering result by the use of more succinct representations of the underlying low-dimensional structures and by increasing robustness to corruption. Geodesic information is only used locally and thus in practice the algorithm can fit well to MMM and not just MGM. Validated against state-of-the-art existing methods for the non-Euclidean setting, GCT exhibited notable performance in clustering accuracy. More specifically, the paper tested GCT on synthetic and real data of deformed images clustering, action identification in video sequences, brain fiber segmentation in medical imaging and dynamic texture clustering.

3.7 Competing Clustering Algorithms and Their Implementation Details

Section 3.7.1 reviews the competing methods of GCT (in the Riemannian setting) and Section 3.7.2 describes the implementation of both GCT and the competing algorithms, in particular, the choice of all parameters.

3.7.1 Review of Competing Algorithms

The first competing algorithm is sparse manifold clustering (SMC). This algorithm was first suggested by [26] for clustering submanifolds embedded in Euclidean spaces and later modified by [69] for clustering submanifolds of the sphere. We adapt it to the current setting of clustering submanifolds of a Riemannian manifold and still refer to it as SMC. Its basic idea is as follows: For each data-point x, a local neighborhood is mapped to the tangent space $T_x M$ by the logarithm map and a sparse coding task is solved in $T_x M$ to provide weights for the spectral-clustering similarity matrix.

The second competing algorithm is spectral clustering with Riemannian metric

(SCR) by [35]. It applies spectral clustering with the weight matrix W whose entries are $\mathbf{W}_{ij} = e^{-\operatorname{dist}_g^2(x_i,x_j)/(2\sigma^2)}$ (see page 4 of [35]). That is, it replaces the usual Euclidean metric in standard spectral clustering with the Riemannian one.

The third competing scheme is the embedded K-means. It embeds the given dataset, which lies on a Riemannian manifold, into a Euclidean spaces (as explained next) and then applies the classical K-means to the embedded dataset. In the experiments, Grassmannian manifolds are embedded by a well-known isometric embedding into Euclidean space [84]; the manifolds of symmetric $n \times n$ PD matrices are embedded by vectorizing their elements into elements of $\mathbb{R}^{\binom{n+1}{2}}$; and data in the sphere $\mathbb{S}^D$ is already embedded in $\mathbb{R}^{D+1}$.

3.7.2 Implementation Details for All Algorithms

GCT follows the scheme of Algorithm 4. For all algorithms, the number K of clusters was known in all experiments The input parameters of GCT are set as follows: The neighborhood radius r at a point x is chosen to be the average distance of x to its nth nearest point over all x, where $n \in \{15, 16, \ldots, 30\}$; the distance and angle thresholds σ_d and σ_a are set to 1 in all experiments (we did not notice a big difference of the results when their values are changed). The dimension of the local tangent space is determined by the largest gap of eigenvalues of each local covariance matrix (more precisely, it is the number of eigenvalues until this gap).

Since there are no online available codes for SMC, SCR and EKM, we wrote our own implementations and will post them (as well as our implementation of GCT) on the supplemental webpage when the paper is accepted for publication. The spectral clustering code in GCT, SMC and SCR, as well as the K-means code in EKM are taken from the implementations of [91]. To make a faithful comparison, the input parameter r of SMC is the same as GCT (in particular, we use the radius of neighborhood and not the number of neighbors). SMC also implicitly sets $\sigma_d = 1$. There are no other parameters for SMC. We remark that [26] formed the weight matrix $\mathbf{W}$ as follows: $\mathbf{W}_{ij} = |\mathbf{S}_{ij}| + |\mathbf{S}_{ji}|$, where $|\mathbf{S}_{ij}|$ and $|\mathbf{S}_{ji}|$ are the sparse coefficients. However, this weight was unstable in some experiments and above a certain level of noise SMC often collapsed in some of the random repetition of the experiments. In such cases, we used instead (for all repetitive experiments for the same data set) the weights $\mathbf{W}_{ij} = \exp\left(|\mathbf{S}_{ij}| + |\mathbf{S}_{ji}|\right)$

suggested in [69] (which are similar to the ones of GCT). In the case of no collapse with the former weights, we tried both weights and noticed that the weights $\mathbf{W}_{ij} = |\mathbf{S}_{ij}|+|\mathbf{S}_{ji}|$ always yielded more accurate results for SMC; we thus used them then even though they can give an advantage over GCT, which uses exponential weights. Overall, the weight $\mathbf{W}_{ij} = |\mathbf{S}_{ij}| + |\mathbf{S}_{ji}|$ was used in the synthetic datasets II-VI of Section 3.4.1. The exponential weight was used in the rest of the experiments, that is, in synthetic dataset I and in the real or stylized applications. It was also used for dataset VI in Figure 3.6 under noise levels mostly higher than the 0.025 noise level used in Section 3.4.1. The collapse phenomenon is evident in Figure 3.6 for noise levels above 0.05.

The SCR algorithm has only one parameter σ_d which is set to 1 (similarly to the analogous parameter of GCT). EKM has no input parameters.

3.8 Computation of Logarithm Maps and Distances

We discuss the complexity of computing logarithm maps for Grassmannians, symmetric PD matrices and spheres. We remark though that it is possible to compute the logarithm maps for data sampled from more general Riemannian manifolds and without knowledge of the manifold, but at a significantly slower rate [79]. We also show that once the logarithm map is computed, then in all these cases the computation of the geodesic distances is of lower or equal order.

A fast way to compute the logarithm map of the Grassmannian $\mathrm{G}(p, \ell)$ (whose dimension is $D = \ell(p - \ell)$) is provided in [92]. It requires a $p \times \ell$ matrix L, with orthogonal columns, and a $p \times p$ orthonormal matrix R for each subspace, where the subspace is spanned by the columns of L, with L comprising the first k columns of R. Given two pairs (L_1, R_1) and (L_2,R_2) for two subspaces, one needs to compute $\log_{L_1}(L_2)$. This computation, which is clarified in [92], includes the singular value decomposition of $L_1^T L_2$ and $R_1^T L_2$. In total, the complexity is $\mathcal{O}(p^2\ell)$, or equivalently, $\mathcal{O}((D/\ell + \ell)^2\ell)$ (since $D = \ell(p - \ell)$).

For the set of $p \times p$ symmetric PD matrices (whose dimension is $D = p(p+1)/2$), [93] computes the logarithm $\log_{M_1}(M_2)$ of any such matrices M_1 and M_2 by first finding the Cholesky decomposition $M_1 = GG^T$ and then computing $\log_{M_1}(M_2) = G \log(GM_2G)G$, where the latter log is the matrix logarithm. The complexities of all major operations (i.e., Cholesky decomposition, the matrix logarithm and the matrix multiplication) are $\mathcal{O}(p^3)$. Therefore, the total complexity is also of order $\mathcal{O}(p^3)$, or equivalently, $\mathcal{O}(D^{1.5})$ (since the dimension of the set of symmetric PD matrices is $D = p(p+1)/2$).

The formula for finding the logarithm map on $\mathbb{S}^D$ is (see [69])

$$\log_{x_i}(x_j) = \frac{x_j - (x_i^T x_j)x_i}{\sqrt{1 - (x_i^T x_j)^2}} \cos^{-1}(x_i^T x_j),$$

where $x_i^T x_j$ is the (Euclidean) dot-vector product. Since it involves inner products and basic operations (also coordinatewise), it takes $\mathcal{O}(D)$ operations to compute it.

For $x_1, x_2 \in M$, $\operatorname{dist}_g(x_1, x_2) = \| \log_{x_1}(x_2) \|_2$. Once we have the image $\log_{x_1}(x_2)$ (which is a vector in the tangent space), the Riemannian distance is computed as the Euclidean norm of the image vector, which involves a computation of order $\mathcal{O}(D)$. Since the algorithm already computes the logarithm maps, the additional cost for computing the geodesic distances are of lower order than the logarithm maps in all 3 cases.

3.9 Computational complexity of GCT and TGCT

The computational complexity of GCT is examined per data-point x_i. It involves the computation of Riemannian distances and the logarithm map, which depends on the Riemannian manifold M (see estimates in Section 3.8). The complexity of computing the Riemannian distance between x_i and x_j and the logarithm map for x_j w.r.t. x_i are denoted by CR and CL respectively (their computational complexity for the cases of the sphere, Grassmannian and PD matrices were discussed in Appendix 3.8). A major part of GCT occurs in the r-neighborhood of x_i (WLOG), where r was defined as the average distance to the 30th nearest point from the associated data-point. To facilitate the analysis of computational complexity, we use instead of r the parameter k of k-nearest-neighbors (k-NN) around x_i. Due to the choice of r, we assume that $k \sim 30$.

The complexity for computing the k-NN of x_i is $\mathcal{O}(N \cdot \mathrm{CR} + k \log(N))$, where $\mathcal{O}(N \cdot \mathrm{CR})$ refers to the complexity of computing $N - 1$ distances, and $\mathcal{O}(k \log(N))$ refers to the effort of identifying the k smallest ones. The second step of Algorithm 4 is to solve the sparse optimization task in (3.3). Notice that due to $\| \cdot \|_2$, only the inner products of data-points are necessary to form the loss function in (3.3), which entails a complexity of order $\mathcal{O}(D)$. Given that only k-NN are involved in (3.3) and that their inner products are required to form the loss, (3.3) is a small scale convex optimization task that can be solved efficiently by any off-the-shelf solver such as the popular alternating direction method of multipliers [94, 95] or the Douglas-Rachford algorithm [96]. The third step of Algorithm 4 is to find the top eigenvectors of the sample covariance matrix defined by the k neighbors of x_i. As shown in Section 3.9.1 below the complexity of this step is $\mathcal{O}(D + k^3)$. Finally, to compute geodesic angles, $\mathcal{O}(N \cdot \mathrm{CL} + ND)$ operations are necessary. Considering all N data-points, the total complexity for the main loop of GCT is $\mathcal{O}(N^2(\mathrm{CR} + \mathrm{CL} + D) + kN\log(N) + ND + Nk^3)$. After the main loop, spectral clustering is invoked on the $N \times N$ affinity matrix $\mathbf{W}$. The main computational burden is to identify K eigenvectors of an $N \times N$ matrix, which entails complexity of order $\mathcal{O}(KN^2)$ (K is the number of clusters). In summary, the complexity of GCT is $\mathcal{O}(N^2(\mathrm{CR} + \mathrm{CL} + D + K) + kN\log(N) + ND + Nk^3)$.

Note that in TGCT, the weights of non-neighboring points are set equal to zero, and geodesic angles are computed only for neighboring points, reducing thus the complexity of this step to $\mathcal{O}(N)$. Moreover, the affinity matrix is sparse in TGCT, effecting thus a potential decrease in the complexity of spectral clustering to the order of6 $\mathcal{O}(N\log N)$ [97, 98]. Therefore, TGCT's complexity becomes $\mathcal{O}(N^2\mathrm{CR} + (k + 1)N\log(N) + kN(\mathrm{CL} + D) + Nk^3)$. The only step that contributes to N^2 in TGCT comes from k-NN. This complexity can be reduced by approximate nearest search. For example, for both the Sphere and the Grassmannian, [99] established an $\mathcal{O}(N^\rho)$ algorithm for approximate nearest neighbor search, where $\rho > 0$ is a sufficiently small parameter. Therefore the total complexity of TGCT for these special cases can be of order $\mathcal{O}(N^{1+\rho}\mathrm{CR} + (k + 1)N\log(N) + kN(\mathrm{CL} + D) + Nk^3)$ (this includes also the preprocessing for the approximate nearest neighbors algorithm).

3.9.1 An Algebraic Trick for Fast Computation of the Tangent Subspace

Consider the $D \times k$ data matrix $\mathbf{X}$ at a specific neighborhood with k points. We need to identify a few principal eigenvectors of the $D \times D$ covariance matrix $\mathbf{X}\mathbf{X}^T$. One can avoid such a costly direct computation (when D is large) by leveraging the following elementary facts from linear algebra: (i) If $(\lambda, \mathbf{v})$ is an eigenvalue-eigenvector pair of $\mathbf{X}^T\mathbf{X}$, then $(\lambda, \mathbf{X}\mathbf{v})$ is an eigenvalue-eigenvector pair of $\mathbf{X}\mathbf{X}^T$, and (ii) $\operatorname{rank}(\mathbf{X}^T\mathbf{X}) = \operatorname{rank}(\mathbf{X}\mathbf{X}^T)$. These facts suggest that the spectra of $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$ coincide, and thus it is sufficient to compute the eigendecomposition of the much smaller $k \times k$ matrix $\mathbf{X}^T\mathbf{X}$, with complexity $\mathcal{O}(k^3)$, which renders the overall cost of eigendecomposition equal to $\mathcal{O}(D + k^3)$, including, for example, the cost of computing $\mathbf{X}\mathbf{v}$.

3.10 Supplementary Details for the Proof of Theorem 3.3.1

3.10.1 Proof of Lemma 3.5.6

Suppose on the contrary that such a constant does not exist. Then there is a sequence $\{x_n\}_{n=1}^{\infty} \subset S_1 \cup S_2$ such that

$$\operatorname{dist}_g(x_n, S_1 \cap S_2) \geq n \max\{\operatorname{dist}_g(x, S_1), \operatorname{dist}_g(x, S_2)\}. \tag{3.63}$$

By picking a subsequence if necessary, assume WLOG that $\{x_n\}_{n=1}^{\infty} \subset S_1$. Since S_1 is compact, there is always a convergent subsequence. Therefore, one may assume that $\{x_n\}_{n=1}^{\infty} \subset S_1$ is also convergent. We show that it converges to a point $z \in S_1 \cap S_2$.

Since $S_1 \cup S_2$ and $S_1 \cap S_2$ are compact, $\operatorname{dist}_g(x_n, S_1 \cap S_2)$ is bounded. Equation (3.63) implies that $\operatorname{dist}_g(x_n, S_2) \to 0$ as n approaches infinity. Suppose $\{x_n\}_{n=1}^{\infty}$ converges to a point $y \notin S_1 \cap S_2$. Then $\operatorname{dist}_g(x_n, S_2) \to \operatorname{dist}_g(y, S_2) > 0$ since $y \notin S_2$. This is a contradiction.

Now that $\{x_n\}_{n=1}^{\infty}$ converges to $z \in S_1 \cap S_2$, one may assume $\{x_n\}_{n=1}^{\infty}$ is in the normal coordinate chart Φ_z of $B(z, r)$ for some fixed $r > 0$. Denote $\mathbf{y}_n = \Phi_z^{-1}(x_n)$, $L_1 = \Phi_z^{-1}(S_1)$ and $L_2 = \Phi_z^{-1}(S_2)$. Since both S_1 and S_2 are geodesic submanifolds, L_1 and L_2 are two subspaces in T_zM. The sequence $\{\mathbf{y}_n\}_{n=1}^{\infty} \subset L_1$ approaches the origin. Lemma 17 of [24] states that

$$\operatorname{dist}_E(\mathbf{y}_n, L_1 \cap L_2) \leq \frac{\operatorname{dist}_E(\mathbf{y}_n, L_2)}{\sin \theta_{\min}(L_1, L_2)},$$

where $\theta_{\min}(L_1, L_2)$ is the minimal nonzero principal angle between L_1 and L_2. Let H be a subset of $B_z(\mathbf{0}, r)$ and arbitrarily fix a point $\mathbf{u} \in H$. It follows from (3.54) (applied with $R = \mathcal{O}(r)$) that

$$\operatorname{dist}_E(\mathbf{y}_n, \mathbf{u})(1 - \mathcal{O}(r^2)) < \operatorname{dist}_g(\mathbf{y}_n, \mathbf{u}) < \operatorname{dist}_E(\mathbf{y}_n, \mathbf{u})(1 + \mathcal{O}(r^2)).$$

Since the term $\mathcal{O}(r^2)$ depends only on the metric g, not on $\mathbf{y}_n$ or $\mathbf{u}$, it is easy to see that

$$\operatorname{dist}_E(\mathbf{y}_n, H)(1 - \mathcal{O}(r^2)) < \operatorname{dist}_g(\mathbf{y}_n, H) < \operatorname{dist}_E(\mathbf{y}_n, H)(1 + \mathcal{O}(r^2)). \tag{3.64}$$

If we let $H = L_1 \cap L_2$ then (3.64) implies that

$$\operatorname{dist}_g(\mathbf{y}_n, L_1 \cap L_2) \leq \frac{(1 + \mathcal{O}(r^2)) \operatorname{dist}_g(\mathbf{y}_n, L_2)}{(1 - \mathcal{O}(r^2)) \sin \theta_{\min}(L_1, L_2)}.$$

This is equivalent to

$$\operatorname{dist}_g(x_n, S_1 \cap S_2) \leq \frac{(1 + \mathcal{O}(r^2)) \operatorname{dist}_g(x_n, S_2)}{(1 - \mathcal{O}(r^2)) \sin \theta_{\min}(L_1, L_2)} < \frac{2}{\sin \theta_0} \operatorname{dist}_g(x_n, S_2)$$

for a fixed small r. This contradicts (3.63).

3.10.2 Proof of (3.38)

The measures μ_{x_0} and μ_z are used to denote the induced measures on $\Phi_{x_0}^{-1}(B(x_0, r) \cap (S_1 \cup S_2))$ and $\Phi_z^{-1}(B(x_0, r) \cap (S_1 \cup S_2))$ by $\mu_{gS_1} + \mu_{gS_2}$. Let $H = \Phi_{x_0}^{-1}(B(x_0, r) \cap (S_1 \cup S_2))$ and $\phi_{x_0} = \Phi_z^{-1} \circ \Phi_{x_0}$ be the transition map. Note that

$$\begin{aligned}
&\mathbb{E}_{\mu_{gS}} \mathbf{C}_{x_0}^z = \mathbb{E}_{\mu_z}((\mathbf{y} - \mathbb{E}_{\mu_z}\mathbf{y}) \cdot (\mathbf{y} - \mathbb{E}_{\mu_z}\mathbf{y})^T) \\
&= \frac{1}{\mu_z(\phi_{x_0}(H))^3} \int_{\mathbf{y} \in \phi_{x_0}(H)} \left(\int_{\mathbf{u} \in \phi_{x_0}(H)} (\mathbf{y} - \mathbf{u}) \mu_z(d\mathbf{u}) \cdot \int_{\mathbf{u} \in \phi_{x_0}(H)} (\mathbf{y} - \mathbf{u})^T \mu_z(d\mathbf{u}) \right) \mu_z(d\mathbf{y}).
\end{aligned} \tag{3.65}$$

Let $\mathbf{y} = \phi_{x_0}(\mathbf{x})$ and $\mathbf{u} = \phi_{x_0}(\mathbf{v})$. We note that $\mathbf{x}, \mathbf{v} \in B(\mathbf{0}, r)$ and $\mathbf{y}, \mathbf{u} \in B(\mathbf{0}, (C'+1)r)$. It follows from the triangle inequality, double application of (3.54) (first with $R = (C'+1)r$ and next with $R = r$), the elementary bound $\operatorname{dist}_E(\mathbf{r}, \mathbf{s}) \leq 2\operatorname{diam}(M)$, where $\mathbf{r}, \mathbf{s}$ are images by the logarithm map of points in M and $\operatorname{diam}(M)$ is the diameter of M and the identity $l_g(\mathbf{y}, \mathbf{u}) = l_g(\mathbf{x}, \mathbf{v})$ (which holds since ϕ_{x_0} preserves the Riemannian distance) that

$$\begin{aligned}
&|\|\mathbf{y} - \mathbf{u}\|_2 - \|\mathbf{x} - \mathbf{v}\|_2| = |\|\mathbf{y} - \mathbf{u}\|_2 - l_g(\mathbf{y}, \mathbf{u}) + l_g(\mathbf{y}, \mathbf{u}) - \|\mathbf{x} - \mathbf{v}\|_2| \qquad (3.66)\\
&\leq |\|\mathbf{y} - \mathbf{u}\|_2 - l_g(\mathbf{y}, \mathbf{u})\| + \|l_g(\mathbf{x}, \mathbf{v}) - \|\mathbf{x} - \mathbf{v}\|_2| \leq 2C_4\operatorname{diam}(M)[(C'+1)^2 + 1]r^2.
\end{aligned}$$

Applying Taylor's expansion to $\mathbf{y} = \phi_{x_0}(\mathbf{x})$, and using the fact that $\|\mathbf{x}\|_2 \leq r$, we note that

$$\|\mathbf{y} - \mathbf{b}_{x_0} - \mathbf{A}_{x_0}\mathbf{x}\|_2 \leq C_S'' r^2, \tag{3.67}$$

where $\mathbf{b}_{x_0}$ and $\mathbf{A}_{x_0}$ depend only on x_0 and C''_S is a constant depending on the Riemannian metric g. Applying the triangle inequality, (3.66) and (3.67) (first with $\mathbf{y} = \phi_{x_0}(\mathbf{x})$ and next with $\mathbf{u} = \phi_{x_0}(\mathbf{v})$ instead of $\mathbf{y}$) we conclude that for all $\mathbf{x}, \mathbf{v} \in B_{x_0}(\mathbf{0}, r)$

$$\begin{aligned} & |\|\mathbf{A}_{x_0}(\mathbf{x}-\mathbf{v})\|_2 - \|\mathbf{x}-\mathbf{v}\|_2| \qquad (3.68)\\ & \leq |\|\mathbf{y}-\mathbf{u}\|_2 - \|\mathbf{x}-\mathbf{v}\|_2| + \|\mathbf{y}-\mathbf{b}_{x_0} - \mathbf{A}_{x_0}\mathbf{x}\|_2 + \|\mathbf{u}-\mathbf{b}_{x_0} - \mathbf{A}_{x_0}\mathbf{v}\|_2 \\ & \leq [2C_4 \mathrm{diam}(M)((C'+1)^2+1) + 2C''_S]r^2. \end{aligned}$$

In particular, suppose $\|\mathbf{x}-\mathbf{v}\|_2 = r$, then (3.68) implies that for any unit-length vectors $\mathbf{w} \in \mathbb{R}^D$ ($\mathbb{R}^D$ is identified with T_{x_0})

$$|\|\mathbf{A}_{x_0}\mathbf{w}\|_2 - 1| \leq [2C_4 \mathrm{diam}(M)((C'+1)^2+1) + 2C''_S]r. \qquad (3.69)$$

We prove below in Appendix 3.10.2 that there exists an orthogonal matrix $\mathbf{R}_{x_0}$ such that

$$\mathbf{A}_{x_0} = \mathbf{R}_{x_0} + \mathcal{O}(r). \qquad (3.70)$$

This leads to

$$\mathbf{y} = \mathbf{b}_{x_0} + \mathbf{R}_{x_0}\mathbf{x} + \mathcal{O}(r^2) \quad \text{and} \quad \mathbf{u} = \mathbf{b}_{x_0} + \mathbf{R}_{x_0}\mathbf{v} + \mathcal{O}(r^2).$$

Consequently,

$$\mathbf{y} - \mathbf{u} = \mathbf{R}_{x_0}(\mathbf{x}-\mathbf{v}) + \mathcal{O}(r^2). \qquad (3.71)$$

We also note that since μ_z and μ_{x_0} are induced from μ, then

$$\mu_z(\phi_{x_0}(H)) = \mu_{x_0}(H) \qquad (3.72)$$

At last, (3.38) is concluded by applying (3.65) (first with $\mathbf{y}$ and $\mathbf{u}$ and next with $\mathbf{x}$ and $\mathbf{v}$ while using appropriate change of variables), (3.71) and (3.72).

Proof of (3.70)

We show that if $\mathbf{A}$ is an $D \times D$ matrix such that $|\|\mathbf{A}\mathbf{w}\|_2 - 1| \leq Cr$ for all unit-length vectors $\mathbf{w} \in \mathbb{R}^D$ and a fixed constant $C > 0$, then there exists an orthogonal matrix $\mathbf{R}$ such that $\mathbf{A} = \mathbf{R} + \mathcal{O}(r)$. In other words, the ijth entries of $\mathbf{A}$ and $\mathbf{R}$ satisfy

$$|\mathbf{A}_{ij} - \mathbf{R}_{ij}| \leq f(C, D)r \qquad (3.73)$$

for a bounded function f (we only show below that the RHS of (3.73) is bounded by a constant times r, but it is not hard to see that this constant depends on C and D; this dependence is used later in (3.97) in order to provide a clearer idea of the constant C_S''').

By performing Gram-Schmidt orthogonalization on rows, the matrix $\mathbf{A}$ can be written as a product of an upper triangular matrix $\mathbf{U}$ and an orthogonal matrix $\mathbf{R}$ (this is the $\mathbf{RQ}$ decomposition of $\mathbf{A}$, but with $\mathbf{U}$ and $\mathbf{R}$ used instead of $\mathbf{R}$ and $\mathbf{Q}$ respectively). Since $\mathbf{R}$ preserves the length of vectors, the condition on $\mathbf{A}$ becomes

$$||\|\mathbf{U}\mathbf{w}\|_2 - 1| \leq Cr, \tag{3.74}$$

for all unit-length vectors $\mathbf{w}$. It is enough to show that up to a change of sign of the rows of $\mathbf{R}$: $\mathbf{U} = \mathbf{I} + \mathcal{O}(r)$. This is proved by induction on D.

If $D = 1$, then $\mathbf{U}$ is a 1×1 matrix. Let $\mathbf{w} = 1$. In this case (3.74) implies that $\mathbf{U} = \pm 1 + \mathcal{O}(r)$. By possible change of sign of $\mathbf{R}$ we conclude that $\mathbf{U} = 1 + \mathcal{O}(r)$.

We assume that the claim is true for $D = k - 1$. Let $\mathbf{U}$ be a $k \times k$ upper rectangular matrix and express it as follows:

$$\mathbf{U} = \begin{pmatrix} \mathbf{V}_{k-1 \times k-1} & \mathbf{x}_{k-1 \times 1} \\ \mathbf{0}_{1 \times k-1} & \mathbf{U}_{kk} \end{pmatrix},$$

where $\mathbf{V}$ is $(k-1) \times (k-1)$ upper triangular matrix, $\mathbf{0}_{1 \times k-1}$ is a row vector of $k-1$ zeros, $\mathbf{x}_{k-1 \times 1}$ is a column vector in $\mathbb{R}^{k-1}$ and $\mathbf{U}_{kk} \in \mathbb{R}$. We assume that $\mathbf{U}$ satisfies (3.74) and show that $\mathbf{U} = \mathbf{I} + \mathcal{O}(r)$ by basic estimates with different choices of $\mathbf{w} \in \mathbb{R}^k$ used in (3.74).

Assume first that $\mathbf{w} = [\mathbf{v}^T, 0]^T$, where $\mathbf{v} \in \mathbb{R}^{k-1}$ is of unit-length. Then (3.74) implies that

$$||\|\mathbf{V}\mathbf{v}\|_2 - 1| \leq Cr. \tag{3.75}$$

The induction hypothesis and (3.75) results in the estimate

$$\mathbf{V} = \mathbf{I} + \mathcal{O}(r) \tag{3.76}$$

up to a change of sign in the first $k - 1$ rows of $\mathbf{R}$ (the rotation associated with $\mathbf{U}$).

Next, we show that $\mathbf{U}_{kk} = 1 + \mathcal{O}(r)$. We first let $\mathbf{w} = [\mathbf{0}_{1 \times k-1}, 1]^T$; in this case (3.74) implies that

$$\sqrt{\|\mathbf{x}\|_2^2 + \mathbf{U}_{kk}^2} - 1 = \mathcal{O}(r), \tag{3.77}$$

which leads to

$$\|\mathbf{x}\|_2^2,\ |\mathbf{U}_{kk}|^2 \leq 1 + \mathcal{O}(r). \tag{3.78}$$

We next let $\mathbf{w} = [-\mathbf{x}^T, 1]^T / \|[-\mathbf{x}^T, 1]^T\|_2$. Then (3.74), with $\|\mathbf{x}\|_2^2$ being bounded by $1 + \mathcal{O}(r)$, implies that

$$\sqrt{\mathbf{U}_{kk}^2 + \mathcal{O}(r^2)} - \|[-\mathbf{x}^T, 1]^T\|_2 = \mathcal{O}(r). \tag{3.79}$$

Moving the second term of the LHS of (3.79) to the RHS of (3.79) and squaring both sides result in

$$\mathbf{U}_{kk}^2 \geq \|[-\mathbf{x}^T, 1]^T\|_2^2 - \mathcal{O}(r) \geq 1 - O(r). \tag{3.80}$$

The combination of (3.78) and (3.80) implies that

$$|\mathbf{U}_{kk}^2 - 1| \leq \mathcal{O}(r). \tag{3.81}$$

Since $\mathbf{U}_{kk} \geq 0$ WLOG (otherwise one can change the sign of the kth row of $\mathbf{R}$) and since $|\mathbf{U}_{kk}^2 - 1|$ is a Lipschitz function on $\mathbf{U}_{kk}$, (3.81) implies that

$$|\mathbf{U}_{kk} - 1| \leq \mathcal{O}(r). \tag{3.82}$$

In other words, $\mathbf{U}_{kk} = 1 + \mathcal{O}(r)$.

At last, we show that $\mathbf{x}_i = \mathcal{O}(r)$. Moving the second term of the LHS of (3.77) to the RHS of (3.77) and squaring both sides result in

$$\|x\|_2^2 + \mathbf{U}_{kk}^2 = 1 + \mathcal{O}(r). \tag{3.83}$$

It follows from (3.82) and (3.83) that $\|\mathbf{x}\|_2^2 = \mathcal{O}(r)$, which implies that

$$\mathbf{x}_i = \mathcal{O}(\sqrt{r}). \tag{3.84}$$

Denote the standard basis of $\mathbb{R}^k$ by $\{\mathbf{e}_i\}_{i=1}^k$, that is, $\mathbf{e}_1 = [1, 0, \ldots, 0]^T, \ldots, \mathbf{e}_k = [0, \ldots, 0, 1]^T$. Let $\mathbf{w}_i = \frac{\sqrt{2}}{2}\mathbf{e}_i + \frac{\sqrt{2}}{2}\mathbf{e}_k$. Plugging $\mathbf{w}_i$ into (3.74) and further simplification result in

$$\left[\frac{1}{2}(\mathbf{x}_1^2 + \ldots + \mathbf{x}_{k-1}^2) + 1 + \frac{1}{2}\mathbf{x}_i + \mathcal{O}(r)\right]^{1/2} = 1 + \mathcal{O}(r). \tag{3.85}$$

Further application of (3.84) into (3.85) yields the equality

$$\left[1 + \frac{1}{2}\mathbf{x}_i + \mathcal{O}(r)\right]^{1/2} = 1 + \mathcal{O}(r). \tag{3.86}$$

Finally, squaring both sides of (3.86) and simplifying concludes the desired estimate

$$\mathbf{x}_i = \mathcal{O}(r). \tag{3.87}$$

Equations (3.76), (3.82) and (3.87) imply that $\mathbf{U} = \mathbf{I} + \mathcal{O}(r)$ (up to a change of signs of the rows of $\mathbf{R}$) and thus conclude the induction and consequently (3.70).

3.10.3 Proof of (3.39)

Let $H_1 = B_I(\Phi_z^{-1}(x_0), r - \mathcal{O}(r^2)) \cap \Phi_z^{-1}(S_1 \cup S_2)$ and $H_2 = B_I(\Phi_z^{-1}(x_0), r + \mathcal{O}(r^2)) \cap \Phi_z^{-1}(S_1 \cup S_2)$. It follows from (3.54) (applied with $R = \mathcal{O}(r)$) that

$$B_I(\Phi_z^{-1}(x_0), r - \mathcal{O}(r^2)) \subset \Phi_z^{-1}(B(x_0, r)) \subset B_I(\Phi_z^{-1}(x_0), r + \mathcal{O}(r^2)). \tag{3.88}$$

The intersection of all sets in (3.88) with $L_1 \cup L_2 = \Phi_z^{-1}(S_1 \cup S_2)$ and the definitions of H_1, H_2 and H' result in the set inequality

$$H_1 \subset H' \subset H_2. \tag{3.89}$$

Thus,

$$H \setminus H' \subset H_2 \setminus H', \quad H' \setminus H \subset H' \setminus H_1. \tag{3.90}$$

By first applying (3.90) (or its consequence $(H_2 \setminus H') \cup (H' \setminus H_1) = H_2 \setminus H_1$) and then direct estimates (whose details are excluded) we obtain that

$$\mu_{ES}((H_2 \setminus H') \cup (H' \setminus H_1)) = \mu_{ES}(H_2 \setminus H_1) = \mathcal{O}(r)\mu_{ES}(H_1). \tag{3.91}$$

Finally, (3.39) follows from (3.90) and (3.91).

3.10.4 Proof of (3.54)

Denote by $l(t)$ the parameterized line segment in $T_{x_1}M$ connecting $l(0) = \mathbf{x}$ and $l(1) = \mathbf{y}$, where $\mathbf{x}$ and $\mathbf{y}$ are specified in (3.54). We note that

$$\begin{aligned}\operatorname{dist}_g(\mathbf{x}, \mathbf{y}) &= \int_0^1 \sqrt{l'(t)^T g(l(t)) l'(t)} dt = \int_0^1 \sqrt{l'(t)^T (I + \mathcal{O}(R^2)) l'(t)} dt \\ &= \operatorname{dist}_E(\mathbf{x}, \mathbf{y}) + \mathcal{O}(R^2) \operatorname{dist}_E(\mathbf{x}, \mathbf{y}). \end{aligned} \tag{3.92}$$

Equation (3.92) clearly implies (3.54), where $C_4 > 0$ depends only on the Riemannian manifold M.

3.10.5 Proof of (3.58)

We first claim that for any $\alpha > 0$

$$\sin(\theta_{\max}(T^E_{x_1}S_1, T_{x_1}S_1)) \leq \|\mathbf{P}_{T^E_{x_1}S_1} - \mathbf{P}_{T_{x_1}S_1}\| < \frac{\sqrt{2}\|\mathbf{C}_{x_1} - \frac{\alpha r^2}{d+2}\mathbf{P}_{T_{x_1}S_1}\|}{\frac{\alpha r^2}{d+2}}. \tag{3.93}$$

The first inequality of (3.93) follows from Lemma 15 in [24]. Whereas the second inequality follows from the Davis-Kahan Theorem [89].

For the rest of the proof we upper bound the RHS of (3.93). We work in the tangent space T_zM, where z is defined as

$$z = \operatorname{argmin}_{y \in S_1 \cap S_2} \operatorname{dist}_g(x_1, y).$$

Similarly as in the proof of Proposition 3.5.7, if argmin is not uniquely defined, z is arbitrarily chosen among all minimizers. Let the composition map $\phi_{x_1} = \Phi_z^{-1} \circ \Phi_{x_1}$ be the transition map from $T_{x_1}M$ to T_zM. Note that ϕ_{x_1} maps the subspace $T_{x_1}S_1$ to another subspace T_zS_1. Let $\mathbf{R}_{x_1}(L_1)$ denote the image of L_1 in T_zM under the rotation matrix $\mathbf{R}_{x_1}$ (here we identify both $T_{x_1}M$ and T_zM with $\mathbb{R}^D$ via their normal coordinate charts). Using the new terminology the main term in the RHS of (3.93) can be expressed as follows

$$\|\mathbf{C}_{x_1} - \frac{\alpha r^2}{d+2}\mathbf{P}_{T_{x_1}S_1}\| = \|\mathbf{R}_{x_1}\mathbf{C}_{x_1}\mathbf{R}^T_{x_1} - \frac{\alpha r^2}{d+2}\mathbf{P}_{\mathbf{R}_{x_1}(T_{x_1}S_1)}\|. \tag{3.94}$$

The RHS of (3.94) can be bounded by the triangle inequality and (3.41) as follows

$$\begin{aligned}
&\|\mathbf{R}_{x_1}\mathbf{C}_{x_1}\mathbf{R}^T_{x_1} - \frac{\alpha r^2}{d+2}\mathbf{P}_{\mathbf{R}_{x_1}(T_{x_1}S_1)}\| \leq \|\mathbf{R}_{x_1}\mathbf{C}_{x_1}\mathbf{R}^T_{x_1} - \mathbb{E}_{\mu_{ES}}\mathbf{C}_{H'}\| \\
&+ \|\mathbb{E}_{\mu_{ES}}\mathbf{C}_{H'} - \frac{\alpha r^2}{d+2}\mathbf{P}_{T_zS_1}\| + \|\frac{\alpha r^2}{d+2}\mathbf{P}_{T_zS_1} - \frac{\alpha r^2}{d+2}\mathbf{P}_{\mathbf{R}_{x_1}(T_{x_1}S_1)}\| \\
&\leq C'_S r^3 + \|\mathbb{E}_{\mu_{ES}}\mathbf{C}_{H'} - \frac{\alpha r^2}{d+2}\mathbf{P}_{T_zS_1}\| + \|\frac{\alpha r^2}{d+2}\mathbf{P}_{T_zS_1} - \frac{\alpha r^2}{d+2}\mathbf{P}_{\mathbf{R}_{x_1}(T_{x_1}S_1)}\|.
\end{aligned} \tag{3.95}$$

Next, we bound the last term in the RHS of (3.95). It follows from (3.69), (3.67), (3.73) (which implies (3.70)) that for $\mathbf{y} = \phi_{x_1}(\mathbf{x})$

$$\|\mathbf{y} - \mathbf{b}_{x_1} - \mathbf{R}_{x_1}\mathbf{x}\|_2 \leq C'''_S r^2 \quad \forall \|\mathbf{x}\|_2 \leq r, \tag{3.96}$$

where

$$C'''_S = D \cdot f(2C_4 \operatorname{diam}(M)((C'+1)^2+1) + 2C''_S, D) + C''_S. \tag{3.97}$$

It is immediate to see that $\mathbf{b} \in T_z S_1$ by letting $\mathbf{x} = \mathbf{0}$ in the Taylor's expansion. If $\mathbf{v} \in \mathbf{R}_{x_1}(T_{x_1}S_1)$ is a vector such that $\|\mathbf{v}\|_2 = r$ and $\theta(\mathbf{v}, T_z S_1) = \theta_{\max}(\mathbf{R}_{x_1}(T_{x_1}S_1), T_z S_1)$, then (3.96) and the fact that $\phi_{x_1}(\mathbf{R}_{x_1}^{-1}\mathbf{v}) - \mathbf{b} \in T_z S_1$ imply that $\operatorname{dist}(\mathbf{v}, T_z S_1) \le C'''_S r^2$. Consequently,

$$\|\mathbf{P}_{\mathbf{R}_{x_1}(T_{x_1}S_1)} - \mathbf{P}_{T_z S_1}\| = \sin(\theta_{\max}(\mathbf{R}_{x_1}(T_{x_1}S_1), T_z S_1)) = \frac{\operatorname{dist}(\mathbf{v}, T_z S_1)}{\|\mathbf{v}\|_2} \le C'''_S r. \quad (3.98)$$

If $\alpha_0 = (1 + (1 - \delta^2(x_1))_+^{d/2})^{-1}$ (the same as in Lemma 21 of [24]), then the argument in [24, page 41] shows that

$$\|\mathbb{E}_{\mu_{ES}}\mathbf{C}_{H'} - \frac{\alpha_0 r^2}{d+2}\mathbf{P}_{T_z S_1}\| \le 2C_2^{\frac{d}{2}}\eta^{\frac{d}{d+2}} r^2. \quad (3.99)$$

Inequalities (3.95) (with $\alpha = \alpha_0$), (3.98) (with $\alpha = \alpha_0$) and (3.99) imply that

$$\|\mathbf{R}_{x_1}\mathbf{C}_{x_1}\mathbf{R}_{x_1}^T - \frac{\alpha_0 r^2}{d+2}\mathbf{P}_{\mathbf{R}_{x_1}(T_{x_1}S_1)}\| \le C'_S r^3 + 2C_2^{\frac{d}{2}}\eta^{\frac{d}{d+2}} r^2 + \frac{C'''_S \alpha_0}{d+2} r^3. \quad (3.100)$$

Plugging (3.94) (with $\alpha = \alpha_0$) and (3.100) in (3.93) (with $\alpha = \alpha_0$) and applying the fact that $\frac{1}{2} \le \alpha_0 \le 1$ yield

$$\sin(\theta_{\max}(T_{x_1}^E S_1, T_{x_1}S_1)) < 2\sqrt{2}(d+2)(C'_S r + 2C_2^{\frac{d}{2}}\eta^{\frac{d}{d+2}} + \frac{C'''_S}{d+2} r). \quad (3.101)$$

Let

$$C'_3 = 2\sqrt{2}(d+2)\max(2C_2^{\frac{d}{2}}, C'_S + \frac{C'''_S}{d+2}), \quad (3.102)$$

then (3.58) clearly follows from (3.101) and (3.102).

3.10.6 Proof of (3.59)

We prove(3.59), while generalizing the setting to work with two subspaces L_1, L_2 and a line l. Let $\angle(l, L_1) = \theta_1$ and $\angle(l, L_2) = \theta_2$. Assume that

$$\theta_1 \le \alpha \quad (3.103)$$

for an arbitrarily fixed $0 < \alpha < \pi/2$. We use the fact that $\sin(\theta)$ is a concave function. If $\theta_2 > \alpha$, then

$$\frac{1 - \sin(\alpha)}{\pi/2 - \alpha} \le \frac{\sin(\theta_2) - \sin(\alpha)}{\theta_2 - \alpha} < \frac{\sin(\theta_2) - \sin(\theta_1)}{\theta_2 - \theta_1}. \quad (3.104)$$

On the other hand, the fact that $\sin^{-1}(x)$ is a Lipschitz function over the interval $[0, \sin(\alpha)]$ implies that if $\theta_2 \leq \alpha$,

$$|\theta_2 - \theta_1| \leq \frac{1}{\cos(\alpha)} |\sin(\theta_2) - \sin(\theta_1)| \quad \text{for } \theta_1, \theta_2 \in [0, \alpha]. \tag{3.105}$$

Equation (3.104) and (3.105) imply that

$$|\theta_2 - \theta_1| \leq \max\left(\frac{\pi/2 - \alpha}{1 - \sin(\alpha)}, \frac{1}{\cos(\alpha)}\right) |\sin(\theta_2) - \sin(\theta_1)|. \tag{3.106}$$

If $\alpha = \pi/6$, then (3.106) and Lemma 3.2 of [100] lead to the inequality

$$|\theta_2 - \theta_1| \leq \frac{2\pi\sqrt{d}}{3} \sin(\theta_{\max}(L_1, L_2)). \tag{3.107}$$

Thus,

$$\theta_1 \geq \theta_2 - \frac{2\pi\sqrt{d}}{3} \sin(\theta_{\max}(L_1, L_2)) \tag{3.108}$$

as long as (3.103) holds. If (3.103) is not assume, (3.108) can be replaced with

$$\theta_1 \geq \min(\theta_2 - \frac{2\pi\sqrt{d}}{3} \sin(\theta_{\max}(L_1, L_2)), \pi/6) \quad \forall \theta_1 \in [0, \pi/2], \tag{3.109}$$

which translates to (3.59).

Chapter 4

Part III: Nonparametric Bayesian Regression on Manifolds via Brownian Motion

4.1 Introduction

In many applications of regression analysis, the response variables lie in Riemannian manifolds. For example, in directional statistics [101, 102, 103] the response variables take values in the sphere or the group of rotations. Applications of directional statistics include crystallography [104], altitude determination for navigation and guidance control [105], testing procedure for Gene Ontology cellular component categories [106], visual invariance studies [107] and geostatics [108]. Other modern applications of regression give rise to different types of manifold-valued responses. In the regression problem of estimating shape deformations of the brain over time (e.g., for studying brain development, aging or diseases), the response variables lie in the space of shapes [109, 110, 111, 112, 113, 114]. In the analysis of landmarks [115] the response variables lie in the Lie group of diffeomorphisms.

The quantitative analysis of regression with manifold-valued responses (which we refer to as manifold-valued regression) is still in early stages and is significantly less developed than statistical analysis of vector-valued regression with manifold-valued predictors [116, 117, 118, 119, 120, 121, 122]. A main obstacle for advancing the analysis of manifold-valued regression is that there is no linear structure in general Riemannian manifolds and thus no direct method for averaging responses. Parametric methods for regression problems with manifold-valued responses [109, 111, 107, 123, 115] directly generalize the linear or polynomial real-valued regressions to geodesic or Riemannian polynomial manifold-valued regression. Nevertheless, the geodesic or Riemannian polynomial assumption on the underlying function is often too restrictive and for many applications non-parametric models are required. To address this issue, Hein [124] and Bhattacharya [112] proposed kernel-smoothing estimators, where in [124] the predictors and responses take values in manifolds and in [112] the predictors and responses take values in compact metric spaces with special kernels. Hein [124] proved convergence of the risk function to a minimal risk (w.p. 1; conditioned on the predictor) and Bhattacharya [112] established consistency of the joint density function of the predictors and the responses. However, the rate of contraction (that is, the rate at which the posterior distribution contracts to a δ distribution with respect to the underlying regression function) of any previously proposed manifold-valued regression estimator was not established. To the best of our knowledge, rate of contraction was only established when both the predictor and response variables are real [125] and this work does not seem to extend to manifold-valued regression.

The main goal of this paper is to establish the rate of contraction of a natural estimator for manifold-valued regression (with real-valued predictors). This estimator is proposed here for the first time.

4.1.1 Setting for Regression with Manifold-Valued Responses

We assume that the predictor t takes values in $[0,1]$ and the response x takes values in a compact D-dimensional Riemannian manifold M. We denote the Riemannian measure on M by μ ($d\mu$ is the volume form). We also assume an underlying function $f_0 \in C([0,1], M)$, which relates between the predictor variables and response variables by determining a density function $p_{f_0(t)}(x)$, so that

$$x|t \sim p_{f_0(t)}(x). \tag{4.1}$$

We find it natural to define

$$p_{f_0(t)}(x) = p_{\sigma^2}(f_0(t), x), \tag{4.2}$$

where $p_{\sigma^2}(f_0(t), x)$ denotes the heat kernel on M centered at $f_0(t)$ and evaluated at time σ^2. Equivalently, $p_{\sigma^2}(f_0(t), x)$ is the transition probability of Brownian motion on M (with the measure μ) from $f_0(t)$ to x at time σ^2. We note that σ^2 controls the variance of the distribution of $x|t$ and as $\sigma^2 \to 0$, the distribution of $x|t$ approaches $\delta_{f_0(t)}$. In the special case where $M = \mathbb{R}^D$:

$$p_{\sigma^2}(f_0(t), \mathbf{x}) = \frac{1}{(\sqrt{2\pi}\sigma)^D} \exp\left(\frac{-\|\mathbf{x} - f_0(t)\|^2}{2\sigma^2}\right),$$

and this implies the common model: $x - f_0(t) \,|\, t \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$.

We also assume a distribution $p(t)$ of t, whose support equals $[0, 1]$, though its exact form is irrelevant in the analysis. At last, we assume n i.i.d. observations $\{(t_i, x_i)\}_{i=1}^n \subset [0, 1] \times M$ with the joint distribution P_0^n and the density function

$$p_0^n = \prod_{i=1}^n p(t_i) p_{\sigma^2}(f_0(t_i), x_i). \tag{4.3}$$

The aim of the regression problem is to estimate f_0 among all functions in $C([0, 1], M)$ given the observations $\{(t_i, x_i)\}_{i=1}^n$.

For simplicity, we denote throughout the rest of the paper

$$\mathcal{P} := C([0, 1], M).$$

4.1.2 Bayesian Perspective: Prior and Posterior Distributions Based on the Brownian Motion

Since the set of functions $\mathcal{P}$ includes Brownian paths, the heat kernel, which expresses the Brownian transition probability, can be used to form a prior distribution on $\mathcal{P}$. For the sake of clarity, we need to distinguish between two different ways of using the heat kernel in this paper. The first one applies the heat kernel $p_{\sigma^2}(f_0(t), x)$ with $t \in [0, 1]$, $f_0 \in \mathcal{P}$ and $x \in M$ (see e.g., Section 4.1.1), where the time (or variance) parameter σ^2 quantifies the "noise" in x w.r.t. the underlying function $f_0(t)$. The second one uses the heat kernel $p_h(x, y)$ with $h \in \mathbb{R}_+$ and $x, y \in M$, where the time parameter h

inversely characterizes the "smoothness" of the path between x and y. The smaller h, the smoother the path between x and y (since smaller h makes it less probable for y to get further away from x). Using the heat kernel $p_h(x, y)$, we define in Section 4.1.2 a continuous Brownian motion (BM) prior distribution and in Section 4.1.2 a discretized BM prior distribution. Section 4.1.2 then defines posterior distributions in terms of the prior distributions and the given observations $\{(t_i, x_i)\}_{i=1}^n \subset [0, 1] \times M$ of the setting.

The Continuous BM Prior on $\mathcal{P}$

We note that a function $f \in \mathcal{P}$ can be identified as a parametrized path in M. Let's assume that $x \in M$ is a starting point of this path, that is $f(0) = x$. We denote $\mathcal{P}_x := \{f \in \mathcal{P} : f(0) = x\}$. Corollary 2.19 of [126] implies that there exists a unique probability measure W_x on $\mathcal{P}_x$ such that for any $n \in \mathbb{N}$, $0 < t_1 < ... < t_n = 1$, and open subsets $U_1, \dots, U_n \in M$, the following identify is satisfied

$$\begin{aligned} &W_x(f \in \mathcal{P}_x \mid f(t_1) \in U_1, \dots, f(t_n) \in U_n) = \\ &\qquad \int_{U_1 \times \ldots \times U_n} p_{t_n - t_{n-1}}(x_n, x_{n-1}) \cdots p_{t_2 - t_1}(x_2, x_1) p_{t_1}(x_1, x) d\mu(x_1) \cdots d\mu(x_n). \quad (4.4) \end{aligned}$$

We define the conditional prior distribution of $f \in \mathcal{P}$ given $x \in M$ by W_x. We assume that the distribution of $f(0) = x$ is $\mu/\mu(M)$ and thus obtain that the prior distribution $\Pi(f)$ of $f \in \mathcal{P}$ is $W_x \times \mu/\mu(M)$.

The Discretized BM Prior $\mathcal{P}$

The continuous BM prior often does not have a density function. We discuss here a special case of discretized BM, where the density function of the prior is well-defined. For $0 < h < 1$ such that $1/h$ is an integer, we define $PGF(h)$ as the set of *piecewise geodesic functions* from $[0, 1]$ to M, where for each $0 \leq k < 1/h$, $k \in \mathbb{N}$, the interval $[kh, (k+1)h]$ is mapped to the geodesic curve from $f(kh)$ to $f((k+1)h)$. Each function in $PGF(h)$ is determined by its values at $f(kh)$. Let the distribution of $f(0)$ be uniform w.r.t. the Riemannian measure μ and let the transition probability from $f(kh)$ to $f((k+1)h)$ be given by the heat kernel $p_h(f(kh), f((k+1)h))$. Then the density function π_h (w.r.t.

μ) of the discretized BM prior on $PGF(h)$ can be specified as follows:

$$\pi_h(f) = \frac{1}{\mu(M)} \prod_{k=1}^{1/h} p_h(f(kh-h), f(kh)).$$

The corresponding distribution is denoted by Π_h.

Throughout the paper we assume a sequence $b_n \to 0$ with $0 < b_n < 1$ and with some abuse of notation denote by Π_n the sequence of discretized BM priors defined above with $h = b_n$. By construction, Π_n is supported on $PGF(b_n)$. Since $PGF(b_n) \subset \mathcal{P}$, Π_n can also be considered as a set of priors on $\mathcal{P}$.

Posterior Distributions

Given observations $\{(t_i, x_i)\}_{i=1}^n$ drawn according to the setting of Section 4.1.1, the posterior distribution of Π has the density function

$$\begin{aligned}\Pi(f \in A | \{(t_i, x_i)\}_{i=1}^n) &\propto \int_{f \in A} \prod_{i=1}^n p(t_i, x_i | f) d\Pi(f) \\ &= \int_{f \in A} \prod_{i=1}^n p_{f(t_i)}(x_i) p(t_i) d\Pi(f),\end{aligned} \tag{4.5}$$

where the equality in (4.5) follows by applying (4.1) and (4.2) to the estimator f of f_0.

4.1.3 Main Theorems: Posterior Consistency and Rate of Contraction

We establish the posterior consistency for the discretized and continuous BM priors respectively. That is, we show that as n approaches infinity, the posterior distributions contract with high probability to the distribution δ_{f_0} (recall that f_0 is the underlying function in $\mathcal{P}$). Furthermore, for the discretized BM we study the rate of contraction of the posterior distribution. The theorem for the discretized BM is formulated in Section 4.1.3 and the one for the continuous BM (with weaker convergence) in Section 4.1.3.

Posterior Consistency and Rate of Contraction for Discretized BM

Theorem 4.1.1 below formulates the rate of contraction of the posterior distribution of the discretized BM with respect to the L_q metric on $\mathcal{P}$, where $1 \leq q < \infty$. This metric,

d_q, is defined as follows:

$$d_q(f_1, f_2) = \left(\int_{t \in [0,1]} \operatorname{dist}_M(f_1(t), f_2(t))^q p(t) dt \right)^{1/q}, \tag{4.6}$$

where dist_M denotes the geodesic distance on M and $p(t)$ is the pdf for the predictor t.

Theorem 4.1.1. *Assume a regression setting with a predictor variable $t \in [0, 1]$, whose pdf $p(t)$ is strictly positive on $[0, 1]$, a response variable x in a compact finite-dimensional Riemannian manifold M and an underlying and unknown Lipschitz function $f_0 \in \mathcal{P}$, which relates between x and t according to (4.1) and (4.2). Assume an arbitrarily fixed $0 < \epsilon < 1/4$ and for $n \in \mathbb{N}$, let $b_n = n^{-1/2+2\epsilon}$ be the sidelength of the set $PGF(b_n)$ and let $\{\Pi_n\}_{n\in\mathbb{N}}$ denote the sequence of discretized BM priors on $PGF(b_n)$. Then there exists an absolute constant A_0 and a fixed constant C_0 depending only on the positive minimum value of $p(t)$ on $[0, 1]$, the volume of M and the Riemannian metric of M*[1] *, such that $\Pi_n(\cdot|\{(t_i, x_i)\}_{i=1}^n)$ contracts to f_0 according to the rate $\epsilon_n = \sqrt{b_n/C_0} = O(n^{-1/4+\epsilon})$. More precisely, for any $1 \le q < \infty$*

$$\Pi_n(f : d_q(f, f_0) \ge A_0\epsilon_n|\{(t_i, x_i)\}_{i=1}^n) \to 0$$

in P_0^n-probability (see (4.3)) as $n \to \infty$.

The proof of Theorem 4.1.1 appears in Section 4.2 and utilizes a general strategy for establishing contraction according to [127]. The significance of the theorem is in properly determining the sidelength parameter b_n (as a function of n). Practical application of the discretized BM prior can suffer from underfitting or overfitting as a result of too small or too large choice of b_n respectively. Theorem 4.1.1 implies that for n observations, b_n should be picked as $n^{-1/2+2\epsilon}$ to achieve a contraction rate of $O(n^{-1/4+\epsilon})$ for any fixed $\epsilon > 0$.

Posterior Consistency for Continuous BM

We show here that the posterior distribution $\Pi(\cdot|\{(t_i, x_i)\}_{i=1}^n)$ is weakly consistent. In order to clearly specify the weak convergence, it is natural to identify functions in $\mathcal{P}$

[1] *More precisely, the dependence of the constant C_{II} (which is later defined in (4.15)) on the Riemannian metric.*

with density functions of observations. Let $\mathcal{D}$ denote the set of densities $p(t,x)$ from which the observations $\{(t_i,x_i)\}_{i=1}^n \subset [0,1]\times M$ are drawn. Assuming a fixed variance σ^2, a function $f\in\mathcal{P}$ can be identified with a density function $p_f\in\mathcal{D}$ as follows:

$$\Phi: \quad f \longrightarrow p_f(t,x) := p_{\sigma^2}(f(t),x)p(t). \tag{4.7}$$

Therefore, Π induces a prior on the set $\mathcal{D}$, which is again denoted by Π with some abuse of notation. For the simplicity of analysis, we assume here that σ^2 is known. Section 4.4.1 discusses the modification needed when σ^2 is unknown.

For the underlying function f_0, we define its weak neighborhood of radius ϵ by

$$N_\epsilon(f_0) = \left\{ f\in\mathcal{P} : \left| \int_{[0,1]\times M} p_g p_f dt d\mu(x) - \int_{[0,1]\times M} p_g p_{f_0} dt d\mu(x) \right| \le \epsilon, \, \forall g\in\mathcal{P} \right\}.$$

Theorem 4.1.2 states the weak posterior consistency of the continuous BM prior Π. It is proved later in Section 4.3.

Theorem 4.1.2. *If M is a compact Riemannian manifold and if the true underlying function $f_0\in\mathcal{P}$ of the regression model is Lipschitz continuous, then the posterior distribution $\Pi(\cdot|\{(t_i,x_i)\}_{i=1}^n)$ is weakly consistent. In other words, for any $\epsilon>0$,*

$$\Pi(N_\epsilon(f_0)|\{(t_i,x_i)\}_{i=1}^n) \longrightarrow 1$$

almost surely w.r.t. the true probability measure P_0^n (defined in (4.3)*) as $n\to\infty$.*

4.1.4 Contributions of this Work

The first contribution of this paper is the proposal of a natural model for manifold-valued regression (with real-valued predictors). Indeed, the heat kernel on the Riemannian manifold gives rise to an averaging process, which generalizes basic averages of vector-valued regression. In particular, the heat kernel on $\mathbb{R}^D$ is the same as the Gaussian kernel (applied to the difference of $f(t)$ and x), which is widely used in regression when $x\in\mathbb{R}^D$ (due to an additive Gaussian noise model). The Bayesian setting is natural for the proposed model, since it uses the discretized or continuous Brownian motion on M as a prior distribution of f and it does not directly use the heat kernel. It is not hard to simulate the Brownian motion, but tight estimates of the heat kernel for general M are hard.

The second and main contribution of this work is the derivation of the contraction rate of the posterior distribution for the discretized Brownian motion. To the best of our knowledge the rate of contraction was only established before for regression with real-valued predictors and responses. For this case, van Zanten [125] established contraction rate $n^{-1/4}$ for the posterior distribution of n samples under the L_p-norm, where $1 \leq p < \infty$. His analysis does not seem to extend to our setting. It is unclear to us if this stronger contraction rate also applies to the general case of manifold-valued regression (see discussion in Section 4.6.3).

The third contribution is the consistency result for the continuous Brownian motion. The only other consistency result for manifold-valued regression we are aware of is by Bhattacharya [112]. It suggests a general nonparametric Bayesian kernel-based framework for modeling the conditional distribution $x|t$, where the predictor t and response x take values in metric spaces with kernels. Under a suitable assumption on the kernels, [112] established the posterior consistency for the conditional distribution w.r.t. the L_1 norm (see [112, Proposition 13.1]). We remark that [112] applies to responses and predictors in Riemannian manifolds (where the corresponding metric kernels are the heat kernels). However, both the conditional distribution (of x given t) and the prior distribution are different than the ones proposed here. It is unclear how to obtain a rate of contraction for [112].

The last contribution is the implication of a new numerical procedure for manifold-valued regression, which is based on simulating a Brownian motion on M. The flexibility of the shapes of the sample paths of the Brownian motion is advantageous over state-of-the-art geodesic regression methods. Real applications often do not give rise to geodesics and thus the nonparametric regression method is less likely to suffer from underfitting. Another nonparametric approach is kernel regression [124, 112]. In Section 4.5, we compare between kernel regression and Brownian motion regression (our method) for a particular example, which is easy to visualize.

4.1.5 Organization of the Rest of the Paper

The paper is organized as follows. Theorems 4.1.1 and 4.1.2 are proved in Sections 4.2 and 4.3 respectively. Section 4.4 extends the framework to the cases where σ^2 is unknown and $p(t)$ is supported on a subset of $[0, 1]$. Section 4.5 demonstrates the performance of the proposed procedure on a particular example, which is easy to visualize, and compares it to kernel regression [124, 112]. At last, Section 4.6 concludes this work and discusses some open problems.

4.2 Proof of Theorem 4.1.1

Our proof utilizes Theorem 2.1 of [127, page 4]. The latter theorem establishes the contraction rate for a sequence of priors Π_n over the set $\mathcal{D}$ of joint densities of the predictor t and response x under some conditions on Π_n and the covering number of $\mathcal{D}$. We thus conclude Theorem 4.1.1 by establishing these conditions.

We use the following distance $d_{q,\mathcal{D}}$ on the space $\mathcal{D}$ with an arbitrarily fixed $1 \leq q < \infty$:

$$d_{q,\mathcal{D}}(p_1, p_2) = \frac{1}{2}\|p_1 - p_2\|_q \quad \text{for } p_1, p_2 \in \mathcal{D}.$$

The regression framework is formulated in terms of the space $\mathcal{P}$ (see Section 4.1.1, in particular, the mapping of $\mathcal{P}$ to $\mathcal{D}$ in (4.7)) and the metric d_q on $\mathcal{P}$ (see (4.6)). We also use the d_∞ metric on $\mathcal{P}$, which is defined by

$$d_\infty(f_1, f_2) = \max_{t\in[0,1]} \operatorname{dist}_M(f_1(t), f_2(t)), \tag{4.8}$$

The proof is organized as follows. Section 4.2.1 shows that under the mapping (4.7) of $\mathcal{P}$ to $\mathcal{D}$, $d_{q,\mathcal{D}}$ is bounded from below by d_q (and above by d_∞). Therefore, the posterior contraction w.r.t. $d_{q,\mathcal{D}}$ implies the posterior contraction w.r.t. d_q. Then, Sections 4.2.2-4.2.4 show that if the sidelengths $\{b_n\}_{n\in\mathbb{N}}$ and a constant $\alpha > 0$ are chosen properly, then the priors $\{\Pi_n\}_{n\in\mathbb{N}}$ and the sieve of functions $\{\mathcal{P}_{n,\alpha}\}_{n\in\mathbb{N}}$ (defined later in (4.21)) satisfy conditions (2.2)-(2.4) respectively in Theorem 2.1 of [127]. The posterior contraction of Π_n is then concluded.

4.2.1 Relations between $d_{q,\mathcal{D}}$, d_q and d_∞

We formulate and prove the following lemma, which relates between $d_{q,\mathcal{D}}$, d_q and d_∞. It is later used as follows: The first inequality of (4.9) deduces L_q convergence in $\mathcal{P}$ from L_q convergence in $\mathcal{D}$. The second inequality of (4.9) is used in finding the covering number of the space $\mathcal{D}$.

Lemma 4.2.1. *If $0 < m_p$, $M_p \in \mathbb{R}$ and $m_p \leq p(t) \leq M_p$ for all $t \in [0,1]$, then there exists two constants $C_0, C_1 > 0$ depending only on m_p, M_p and the Riemannian manifold M such that for any $f_1, f_2 \in \mathcal{P}$ with corresponding densities p_{f_1}, p_{f_2} in $\mathcal{D}$ (via (4.7))*

$$C_0 d_q(f_1, f_2) \leq d_{q,\mathcal{D}}(p_{f_1}, p_{f_2}) \leq C_1 d_\infty(f_1, f_2). \tag{4.9}$$

Proof. For $x_1 \neq x_2$, we define the function

$$F(x_1, x_2, y) = \frac{|p_{\sigma^2}(x_1, y) - p_{\sigma^2}(x_2, y)|}{\text{dist}_M(x_1, x_2)}. \tag{4.10}$$

We note that the first inequality of (4.9) is true if there exists a constant $C_0 > 0$ such that

$$\int_{y \in M} F(x_1, x_2, y)^q d\mu(y) \geq \frac{C_0^q}{m_p^{q-1}}, \quad \forall x_1 \neq x_2 \in M. \tag{4.11}$$

Since M is compact and $p_{\sigma^2}(x, y)$ is infinitely differentiable, for any $\epsilon > 0$, there exists $\delta > 0$ such that

$$\left| F(x_1, x_2, y) - \left| \frac{\partial p_{\sigma^2}(x_1, y)}{\partial v_{12}} \right| \right| \leq \epsilon, \quad \forall x_1, x_2, y \in M, \, \text{dist}_M(x_1, x_2) \leq \delta, \tag{4.12}$$

where $v_{12} \in T_{x_1}M$ is the unit vector of the geodesic connecting x_1 and x_2. Since the heat kernel $p_{\sigma^2}(x_1, y)$ is not constant and due to the compactness of the space of unit tangent vectors, there exists $C_0' > 0$ such that

$$\int_{y \in M} \left| \frac{\partial p_{\sigma^2}(x_1, y)}{\partial v_{12}} \right| d\mu(y) \geq C_0', \quad \forall x_1 \in M, v_{12} \in T_{x_1}M, \|v_{12}\| = 1. \tag{4.13}$$

Inequalities (4.12), (4.13) and the Schwarz inequality imply that

$$\int_{y \in M} F(x_1, x_2, y)^q d\mu(y) \geq C_I := \left(\frac{C_0' - \epsilon \mu(M)}{\mu(M)} \right)^q, \, \forall x_1, x_2, y \in M, \, \text{dist}_M(x_1, x_2) \leq \delta. \tag{4.14}$$

If we pick ϵ small enough (with its δ in (4.12)), C_I is a positive number. On the other hand, if the pair (x_1, x_2) satisfies that $\text{dist}_M(x_1, x_2) \geq \delta$, we show that for some constant $C_{II} > 0$,

$$\int_{y \in M} F(x_1, x_2, y)^q d\mu(y) \geq C_{II}, \, \forall x_1, x_2, y \in M, \, \text{dist}_M(x_1, x_2) \geq \delta. \tag{4.15}$$

Since the set $\{(x_1, x_2) | \, \text{dist}_M(x_1, x_2) \geq \delta\}$ is compact, the existence of C_{II} is guaranteed if we can show that

$$\int_{y \in M} F(x_1, x_2, y)^q d\mu(y) > 0, \, \forall x_1, x_2, y \in M, \, \text{dist}_M(x_1, x_2) \geq \delta,$$

which can further be reduced to showing that given any pair $(x_1, x_2) \in M^2$,

$$\exists y \in M, \quad p_{\sigma^2}(x_1, y) \neq p_{\sigma^2}(x_2, y). \tag{4.16}$$

We prove (4.16) by contradiction. If (4.16) is not true, then

$$p_{\sigma^2}(x_1, y) = p_{\sigma^2}(x_2, y), \quad \forall y \in M. \tag{4.17}$$

If we plug $y = x_1$ and $y = x_2$ respectively in (4.17), and use the symmetry of the heat kernel, to get $p_{\sigma^2}(x_1, x_1) = p_{\sigma^2}(x_1, x_2) = p_{\sigma^2}(x_2, x_2)$, which means that

$$p_{\sigma^2}(x_1, x_2) = \sqrt{p_{\sigma^2}(x_1, x_1) p_{\sigma^2}(x_2, x_2)}. \tag{4.18}$$

On the other hand,

$$\begin{aligned} p_{\sigma^2}(x_1, x_2) &= \int_{z \in M} p_{\sigma^2/2}(x_1, z) p_{\sigma^2/2}(z, x_2) d\mu(z) \\ &\leq \sqrt{\int_{z \in M} p_{\sigma^2/2}(x_1, z)^2 d\mu(z) \int_{z \in M} p_{\sigma^2/2}(z, x_2)^2 d\mu(z)} \\ &= \sqrt{p_{\sigma^2}(x_1, x_1) p_{\sigma^2}(x_2, x_2)}. \end{aligned} \tag{4.19}$$

In view of (4.18) the Cauchy-Schwartz inequality used in (4.19) is an equality and consequently

$$p_{\sigma^2/2}(x_1, z) = p_{\sigma^2/2}(x_2, z), \quad \forall z \in M.$$

Applying the same argument iteratively, we conclude that for any $m > 0$,

$$p_{\sigma^2/2^m}(x_1, z) = p_{\sigma^2/2^m}(x_2, z), \quad \forall z \in M.$$

However, as $m \to \infty$, $p_{\sigma^2/2^m}(x_1, z) \to \delta_{x_1}$ but $p_{\sigma^2/2^m}(x_2, z) \to \delta_{x_2} \neq \delta_{x_1}$. This is a contradiction. Inequality (4.16) and thus (4.15) are proved. We conclude from (4.14) and (4.15), the first inequality of (4.9) with $C_0 = \left(\min(C_I, C_{II}) m_p^{q-1}\right)^{1/q}$.

Next, we establish the second inequality of (4.9). Theorem 4.1.4 in [128, page 105] states that $p_{\sigma^2}(x, y)$ is infinitely differentiable in both variables x and y. In particular, its first partial derivatives are continuous. Furthermore, the fact that M is compact implies that the first partial derivatives are bounded. That is, there exists $C_M > 0$ such that

$$\left|\frac{\partial p_{\sigma^2}(x, y)}{\partial x}\right| \leq C_M, \quad \left|\frac{\partial p_{\sigma^2}(x, y)}{\partial y}\right| \leq C_M.$$

Consequently,

$$|p_{\sigma^2}(x_1, y) - p_{\sigma^2}(x_2, y)| \leq C_M \operatorname{dist}_M(x_1, x_2). \tag{4.20}$$

Applying (4.20) and then bounding $p(t)$ by M_p and dist_M by d_∞, we conclude (4.12) with $C_1 = C_M M_p^{(q-1)/q} \mu(M)$ as follows:

$$\begin{aligned} d_{q,\mathcal{D}}(p_{f_1}, p_{f_2}) &= \left(\iint |p_{\sigma^2}(f_1(t), y)p(t) - p_{\sigma^2}(f_2(t), y)p(t)|^q \, d\mu(y)dt \right)^{1/q} \\ &\leq \left(\iint C_M^q \operatorname{dist}_M(f_1(t), f_2(t))^q p(t)^q d\mu(y)dt \right)^{1/q} \\ &\leq C_M M_p^{(q-1)/q} \mu(M) d_\infty(f_1, f_2). \end{aligned}$$

□

Remark 4.2.2. *We note that when $q = 1$, the constants C_0, C_1 in Lemma 4.2.1 are independent of $p(t)$. In particular, in this case the condition $m_p \leq p(t) \leq M_p$ is not needed.*

4.2.2 Verification of Inequality 2.2 of [127]

We estimate the covering numbers of special subsets of $\mathcal{P}$ and $\mathcal{D}$. The final estimate verifies inequality 2.2 of [127]. We start with some notation and definitions that also include these special subsets of $\mathcal{P}$ and $\mathcal{D}$.

For $0 < \alpha \leq 1$ and $f \in \mathcal{P}$, let

$$\|f\|_\alpha := \max_{t_1, t_2 \in [0,1]} \frac{\operatorname{dist}_M(f(t_1), f(t_2))}{|t_1 - t_2|^\alpha}$$

and

$$\mathcal{P}_\alpha := \{f \in \mathcal{P} | \, \|f\|_\alpha < \infty\}.$$

For a sequence $\{M_n\}_{n \in \mathbb{N}}$ increasing to infinity we define the sieve of functions

$$\mathcal{P}_{n,\alpha} = \{f \in \mathcal{P}_\alpha | \, \|f\|_\alpha \leq M_n\}. \tag{4.21}$$

This induces a sieve of densities $\mathcal{D}_{n,\alpha}$ of $\mathcal{D}_\alpha$ by the map (4.7). For $\epsilon > 0$ and a metric space $\mathcal{E}$ with the metric d, we denote by $N(\epsilon, \mathcal{E}, d)$ the ϵ-covering number of $\mathcal{E}$, which is the minimal number of balls of radius ϵ needed to cover $\mathcal{E}$.

In the rest of the section we estimate the covering numbers of the sets M, $\mathcal{P}_{n,\alpha}$ and $\mathcal{D}_{n,\alpha}$. We assume a decreasing sequence ϵ_n approaching zero. Section 4.2.2 upper bounds $N(\epsilon_n, M, \mathrm{dist}_M)$ for an arbitrary such sequence ϵ_n. Section 4.2.2 upper bounds $N(\epsilon_n, \mathcal{P}_{n,\alpha}, d_\infty)$ for arbitrary sequences ϵ_n and M_n as above. At last, Section 4.2.2 upper bounds $N(\epsilon_n, \mathcal{D}_{n,\alpha}, d_{q,\mathcal{D}})$ for sequences ϵ_n and M_n satisfying an additional condition (see (4.37) below). It verifies inequality 2.2 of [127].

Covering Numbers of M

For any $\epsilon_n > 0$, we construct an ϵ_n-net on the D-dimensional compact Riemannian manifold M. Let $\mathrm{D}(M)$ be the diameter of M. That is,

$$\mathrm{D}(M) = \max_{x,y \in M} \mathrm{dist}_M(x, y).$$

The Nash embedding theorem [129] and Whitney embedding theorem [130] imply that there exists an isometric map

$$E : M \longrightarrow \mathbb{R}^{2D}.$$

Since $\mathrm{D}(E(M)) \leq \mathrm{D}(M)$, the image $E(M)$ is contained in an hypercube HC with side length $2\mathrm{D}(M)$. We partition this HC as a regular grid with grid spacing $\epsilon_n/\sqrt{2D}$ in each direction. Since each point in HC has distance less than ϵ_n to some grid vertex, the set of grid vertices, $GV(\epsilon_n)$, is an ϵ_n-net of HC. Thus the ϵ_n-covering number of HC can be bounded as follows:

$$N(\epsilon_n, HC, \mathrm{dist}_{\mathbb{R}^{2D}}) \leq \left(\frac{2\mathrm{D}(M)}{\epsilon_n/\sqrt{2D}} \right)^{2D}. \tag{4.22}$$

Next, we construct an ϵ_n-net of M using the $\epsilon_n/3$-net $GV(\epsilon_n/3)$ of HC. To begin with, we show in Lemma 4.2.3 that the Riemannian distance and the Euclidean distance are equivalent locally under an isometric embedding.

Lemma 4.2.3. *Let M be a compact Riemannian manifold and E be an isometric embedding to $\mathbb{R}^{2D}$. Then for any fixed constant $C > 0$, there exists a constant $\delta_C > 0$ such that $\forall x, y \in M$ with* $\mathrm{dist}_{\mathbb{R}^{2D}}(E(x), E(y)) < \delta_C$*,*

$$|\,\mathrm{dist}_M(x, y) - \mathrm{dist}_{\mathbb{R}^{2D}}(E(x), E(y))| < C\,\mathrm{dist}_{\mathbb{R}^{2D}}(E(x), E(y)).$$

Proof. Suppose this is not true. Then there exists a sequence of $(x_n, y_n) \in M^2$ such that $\operatorname{dist}_{\mathbb{R}^{2D}}(E(x_n), E(y_n)) \to 0$ and

$$|\operatorname{dist}_M(x_n, y_n) - \operatorname{dist}_{\mathbb{R}^{2D}}(E(x_n), E(y_n))| \geq C \operatorname{dist}_{\mathbb{R}^{2D}}(E(x_n), E(y_n)). \tag{4.23}$$

Since M is compact, there is a subsequence, denoted again by (x_n, y_n), and a point $z \in M$ such that $x_n, y_n \to z$. By picking an orthonormal basis of the tangent space T_zM and using the exponential map $\exp_z$, one has normal coordinates

$$\Phi : \mathbb{R}^D \equiv T_zM \supset B_z(\mathbf{0}, r) \longrightarrow M$$

where $B_z(\mathbf{0}, r)$ is the r-ball centered the origin on T_zM. Let $\log_z = \exp_z^{-1}$ be the logarithm map at z and dist_I be the Euclidean distance on T_zM. Let $\mathbf{x}_n = \log_z(x_n)$ and $\mathbf{y}_n = \log_z(y_n)$. Applying Lemma 12 in [131, page 24] for $\mathbf{x}_n, \mathbf{y}_n$,

$$|\operatorname{dist}_M(x_n, y_n) - \operatorname{dist}_I(\mathbf{x}_n, \mathbf{y}_n)| < O(\max\{\|\mathbf{x}_n\|_2^2, \|\mathbf{y}_n\|_2^2\}) \operatorname{dist}_I(\mathbf{x}_n, \mathbf{y}_n). \tag{4.24}$$

Let f be the composition of Φ with E,

$$f : \quad \mathbb{R}^D \supset B_z(\mathbf{0}, r) \longrightarrow \mathbb{R}^{2D}.$$

We note that $f(\mathbf{x}_n) = E(x_n)$ and $f(\mathbf{y}_n) = E(y_n)$. The Tyler series of f is

$$f(\mathbf{y}_n) - f(\mathbf{x}_n) = (\nabla f(\mathbf{x}_n))^T(\mathbf{y}_n - \mathbf{x}_n) + \frac{1}{2}(\mathbf{y}_n - \mathbf{x}_n)^T(\nabla^2 f(\mathbf{x}_n))(\mathbf{y}_n - \mathbf{x}_n) + \cdots.$$

This implies that

$$\|f(\mathbf{y}_n) - f(\mathbf{x}_n)\|_2 = \|(\nabla f(\mathbf{x}_n))^T(\mathbf{y}_n - \mathbf{x}_n)\|_2 + O(\|\mathbf{y}_n - \mathbf{x}_n\|_2^2). \tag{4.25}$$

On the one hand, since E is an isometric embedding, the linear map

$$\nabla f(\mathbf{0}) : \quad \mathbb{R}^D \longrightarrow \mathbb{R}^{2D}$$

preserves the Euclidean distance. On the other hand, the smoothness of f implies that $\nabla f(\mathbf{x})$ has bounded derivatives. Thus,

$$\nabla f(\mathbf{x}_n) = \nabla f(\mathbf{0}) + O(\|\mathbf{x}_n\|_2). \tag{4.26}$$

Then, (4.25) and (4.26) and the triangle inequality imply that

$$\|f(\mathbf{y}_n) - f(\mathbf{x}_n)\|_2 = \|\mathbf{y}_n - \mathbf{x}_n\|_2 + O(\|\mathbf{x}_n\|_2\|\mathbf{y}_n - \mathbf{x}_n\|_2 + \|\mathbf{y}_n - \mathbf{x}_n\|_2^2).$$

In other words,

$$|\operatorname{dist}_I(\mathbf{x}_n, \mathbf{y}_n) - \operatorname{dist}_{\mathbb{R}^{2D}}(E(x_n), E(y_n))| \leq O(\|\mathbf{x}_n\|_2 + \|\mathbf{y}_n - \mathbf{x}_n\|_2) \operatorname{dist}_I(\mathbf{x}_n, \mathbf{y}_n). \quad (4.27)$$

By (4.24) and (4.27),

$$|\operatorname{dist}_M(x_n, y_n) - \operatorname{dist}_{\mathbb{R}^{2D}}(E(x_n), E(y_n))| < c_n \operatorname{dist}_I(\mathbf{x}_n, \mathbf{y}_n),$$

where $c_n = O(\|\mathbf{x}_n\|_2 + \|\mathbf{y}_n - \mathbf{x}_n\|_2 + \max\{\|\mathbf{x}_n\|_2^2, \|\mathbf{y}_n\|_2^2\})$. Moreover, by (4.27),

$$\operatorname{dist}_I(\mathbf{x}_n, \mathbf{y}_n) < (1 - O(\|\mathbf{x}_n\|_2 + \|\mathbf{y}_n - \mathbf{x}_n\|_2))^{-1} \operatorname{dist}_{\mathbb{R}^{2D}}(E(x_n), E(y_n)).$$

Therefore, if $c'_n = c_n/(1 - O(\|\mathbf{x}_n\|_2 + \|\mathbf{y}_n - \mathbf{x}_n\|_2))$, then

$$|\operatorname{dist}_M(x_n, y_n) - \operatorname{dist}_{\mathbb{R}^{2D}}(E(x_n), E(y_n))| < c'_n \operatorname{dist}_{\mathbb{R}^{2D}}(E(x_n), E(y_n)).$$

We note that $c'_n \to 0$ as $n \to \infty$ since $\mathbf{x}_n, \mathbf{y}_n \to \mathbf{0}$ and this contradicts assumption (4.23). □

Now, we construct an ϵ_n-net of M from $GV(\epsilon_n/3)$.

Lemma 4.2.4. *Let $\widehat{GV}(\epsilon_n/3) = \{x \in GV(\epsilon_n/3) | \operatorname{dist}_{\mathbb{R}^{2D}}(x, M) \leq \epsilon_n/3\}$. There exists a constant $\delta > 0$ such that $E^{-1}(\operatorname{Proj}_{E(M)}(\widehat{GV}(\epsilon_n/3)))$ is an ϵ_n-net of M when $\epsilon_n < \delta$. Consequently,*

$$N(\epsilon_n, M, \operatorname{dist}_M) \leq N(\epsilon_n/3, HC, \operatorname{dist}_{\mathbb{R}^{2D}}).$$

Proof. Suppose $\epsilon_n < \delta := \delta_{1/3}$ where $\delta_{1/3}$ is the constant δ_C in Lemma 4.2.3 with $C = 1/3$. For any point $x \in M$, let y be the vertex in $GV(\epsilon_n/3)$ that is closest to $E(x)$ w.r.t. $\operatorname{dist}_{\mathbb{R}^{2D}}$. Then, by definition, $y \in \widehat{GV}(\epsilon_n/3)$. Let $z = E^{-1}(\operatorname{Proj}_{E(M)}(y))$. To prove the lemma, it is sufficient to show that

$$\operatorname{dist}_M(x, z) < \epsilon_n. \quad (4.28)$$

Since $\operatorname{dist}_{\mathbb{R}^{2D}}(E(x), E(z)) < 2\epsilon_n/3 < \delta_{1/3}$, Lemma 4.2.3 states that

$$|\operatorname{dist}_M(x, z) - \operatorname{dist}_{\mathbb{R}^{2D}}(E(x), E(z))| < \frac{1}{3} \operatorname{dist}_{\mathbb{R}^{2D}}(E(x), E(z)). \quad (4.29)$$

Inequality (4.29) implies (4.28) and thus the lemma as follows:

$$\operatorname{dist}_M(x, z) < \frac{4}{3} \operatorname{dist}_{\mathbb{R}^{2D}}(E(x), E(z)) < 8\epsilon_n/9.$$

□

From now on, we fix an $\epsilon_n/3$-net S_n of M, generated as above from the projection of regular grid vertices $GV(\epsilon_n/9)$ of HC with grid spacing $\epsilon_n/(9\sqrt{2D})$. Lemma 4.2.5 provides an upper bound of the number of points in S_n in the ϵ_n-neighborhood of $x \in S_n$.

Lemma 4.2.5. *For $x \in S_n$ and $X := \{y \in S_n | \operatorname{dist}_M(x,y) \le \epsilon_n\}$,*

$$\#X \le 21^{2D}.$$

Proof. If $y \in X \subset S_n$, then there is a point $z \in GV(\epsilon_n/9)$ such that

$$E^{-1}(\operatorname{Proj}_{E(M)}(z)) = y \quad \text{and} \quad \operatorname{dist}_{\mathbb{R}^{2D}}(z, E(y)) \le \epsilon_n/9. \tag{4.30}$$

We note that

$$\operatorname{dist}_{\mathbb{R}^{2D}}(E(x), E(y)) \le \operatorname{dist}_M(x,y) \tag{4.31}$$

since E is an isometric embedding. Inequalities (4.30) and (4.31) and the triangle inequality imply that $\operatorname{dist}_{\mathbb{R}^{2D}}(E(x), z) \le 10\epsilon_n/9$. Thus, if

$$Y := \{z \in GV(\epsilon_n/9) | \operatorname{dist}_{\mathbb{R}^{2D}}(E(x), z) \le 10\epsilon_n/9\},$$

then $X \subset E^{-1}(\operatorname{Proj}_{E(M)}(Y))$. Since the grid spacing is $\epsilon_n/9$, $\#X \le \#Y = 21^{2D}$.

□

Covering Numbers of $\mathcal{P}_{n,\alpha}$

Recall that $PGF(a) \subset \mathcal{P}_\alpha$ is the set of piecewise geodesic functions which map each interval $[ka, (k+1)a]$ to a geodesic on M for $0 \le k < 1/a$. We define

$$F_{S_n}(a) = \{f \in PGF(a) | f(ka) \in S_n \text{ for } 0 \le k < 1/a\},$$

where S_n was defined just before Lemma 4.2.5. The following Lemma upper bounds $N(\epsilon_n, \mathcal{P}_{n,\alpha}, d_\infty)$. It uses the constant $\delta_{1/9}$ which was defined in Lemma 4.2.3 (here $C = 1/9$).

Lemma 4.2.6. *If M is a D-dimensional compact Riemannian manifold with diameter $D(M)$, two sequences $\{M_n\}_{n\in\mathbb{N}}, \{\epsilon_n\}_{n\in\mathbb{N}}$ such that $M_n \to \infty$ and $\epsilon_n < \delta_{1/9}$, and $a = \frac{\epsilon_n}{3M_n}$, then there is a subset of $F_{S_n}(a)$, which forms an ϵ_n-net of $\mathcal{P}_{n,\alpha}$ and*

$$N(\epsilon_n, \mathcal{P}_{n,\alpha}, d_\infty) \le \left(21^{2D}\right)^{3M_n/\epsilon_n} \left(\frac{18\sqrt{2D}D(M)}{\epsilon_n}\right)^{2D}. \tag{4.32}$$

Proof. Given $f \in \mathcal{P}_{n,\alpha}$, an approximation $\hat{f} \in F_{S_n}(a)$ is determined uniquely by specifying its boundary value $\hat{f}(ka)$ for $0 \le k < 1/a$, which is given by

$$\hat{f}(ka) = \arg\min_{x \in S_n} \operatorname{dist}_M(x, f(ka)).$$

To show that $d_\infty(f, \hat{f}) \le \epsilon_n$, We check the inequality $\operatorname{dist}_M(f(t), \hat{f}(t)) \le \epsilon_n$ for all $t \in [0,1]$. Suppose $t \in [ka, ka+a]$. Since $\|f\|_\alpha \le M_n$,

$$\operatorname{dist}_M(f(ka), f(t)) \le M_n a \le \epsilon_n/3. \tag{4.33}$$

Moreover, because $\hat{f}$ is a mapping to a geodesic on $[ka, ka+a]$ and the fact that S_n is $\epsilon_n/3$-net of M,

$$\operatorname{dist}_M(f(ka), \hat{f}(ka)) \le \epsilon_n/3 \tag{4.34}$$

and

$$\begin{aligned} \operatorname{dist}_M(f(ka), \hat{f}(t)) &\le \operatorname{dist}_M(f(ka), \hat{f}(ka+a)) \qquad (4.35)\\ &\le \operatorname{dist}_M(f(ka), f(ka+a)) + \operatorname{dist}_M(f(ka+a), \hat{f}(ka+a))\\ &\le 2\epsilon_n/3. \end{aligned}$$

It follows from (4.33), (4.35) and the triangle inequality that

$$\operatorname{dist}_M(f(t), \hat{f}(t)) \le \epsilon_n. \tag{4.36}$$

Define subset of $F_{S_n}(a)$:

$$SF_{S_n}(a) = \{f \in F_{S_n}(a) | \operatorname{dist}_M(f(ka), f(ka+a)) \le \epsilon_n, \forall 0 \le k < 1/a\}.$$

By the definitions of $\hat{f}$ and S_n and (4.33), we conclude that $\hat{f} \in SF_{S_n}(a)$. Thus, $SF_{S_n}(a)$ is an ϵ_n-net of $\mathcal{P}_{n,\alpha}$.

By definition, $N(\epsilon_n, \mathcal{P}_{n,\alpha}, d_\infty) \le \#SF_{S_n}(a)$. It is thus sufficient to estimate $\#SF_{S_n}(a)$. Lemma 4.2.4 and (4.22) imply that

$$\#S_n \le \left(\frac{18\mathrm{D}(M)}{\epsilon_n/\sqrt{2D}}\right)^{2D},$$

which is the upper bound of the number of values that $\hat{f}(0)$ can take. Given the value of $\hat{f}(ka)$, there are 21^{2D} choices for $\hat{f}(ka+a)$ by Lemma 4.2.5. Thus, for $a = \epsilon_n/(3M_n)$, (4.32) is concluded as follows

$$\#SF_{S_n}(a) \leq \left(21^{2D}\right)^{3M_n/\epsilon_n} \left(\frac{18\sqrt{2D}\mathrm{D}(M)}{\epsilon_n}\right)^{2D}.$$

□

Covering Numbers of $\mathcal{D}_{n,\alpha}$

In this section, we prove the following lemma.

Lemma 4.2.7. *If M_n satisfies*

$$M_n \leq \frac{n\epsilon_n^3}{6C_1 D(\log(21)+1)}, \tag{4.37}$$

where C_1 was defined in Lemma 4.2.1, then for n sufficiently large $\mathcal{D}_{n,\alpha}$ satisfies the inequality 2.2 of [127, Theorem 2.1], that is,

$$\log N(\epsilon_n, \mathcal{D}_{n,\alpha}, d_{q,\mathcal{D}}) \leq n\epsilon_n^2. \tag{4.38}$$

Proof. Recall that $\mathcal{D}_{n,\alpha} = \Phi(\mathcal{P}_{n,\alpha})$ and $d_{q,\mathcal{D}}(p_{f_1}, p_{f_2}) \leq C_1 d_\infty(f_1, f_2)$ (see Lemma 4.2.1). A consequence of this is that an ϵ_n-net of $\mathcal{D}_{n,\alpha}$ can be induced from an ϵ_n/C_1-net of $\mathcal{P}_{n,\alpha}$. Therefore,

$$N(\epsilon_n, \mathcal{D}_{n,\alpha}, d_{q,\mathcal{D}}) \leq N(\epsilon_n/C_1, \mathcal{P}_{n,\alpha}, d_\infty) \leq \left(21^{2D}\right)^{3C_1M_n/\epsilon_n} \left(\frac{18C_1\sqrt{2D}\mathrm{D}(M)}{\epsilon_n}\right)^{2D}. \tag{4.39}$$

To conclude (4.38), it is enough to show that

$$\frac{3C_1M_n}{\epsilon_n}(2D\log(21)+2D) + 2D\log\left(\frac{18C_1\sqrt{2D}\mathrm{D}(M)}{3C_1M_n}\right) \leq n\epsilon_n^2. \tag{4.40}$$

We verify it for n sufficiently large. Since $M_n \to \infty$, the second term of the LHS of (4.40) will be less than zero for large n. On the other hand, it follows from (4.37) that the first term of the LHS of (4.40) is less than or equal to $n\epsilon_n^2$.

□

4.2.3 Verification of Inequality 2.3 of [127]

Recall that the prior Π_n, with support on $PGF(b_n) \subset \mathcal{P}_\alpha$, is given by the discretized Brownian motion at times $b_n, 2b_n, \ldots, 1$. More specifically, we define the prior Π_n on $PGF(b_n)$ by fixing the joint distribution of $f(kb_n)$ for $0 \le k < 1/b_n$, whose density is given by

$$\pi_n(f) = s(f(0)) \prod_{k=0}^{1/b_n} p_{b_n}(f(kb_n), f(kb_n + b_n)). \tag{4.41}$$

where s is a fixed density function with support on M for $f(0)$, and $p_{b_n}(x, y)$ is the transition probability from x to y of the Brownian motion at time b_n.

In this section, we show that if the sequence b_n is properly chosen, then Π_n satisfies the inequality 2.3 of [127, Theorem 2.1], that is,

$$\Pi_n(\mathcal{D}_\alpha \backslash \mathcal{D}_{n,\alpha}) \le \exp[-n\epsilon_n^2(C+4)]. \tag{4.42}$$

We first establish Lemma 4.2.8 below and then use it to conclude (4.42) in Lemma 4.2.9 below (under a condition on b_n). We use the following set

$$\begin{aligned} X := \{f \in PGF(b_n) | \operatorname{dist}_M(f(k_1 b_n), f(k_2 b_n)) \le M_n (k_2 b_n - k_1 b_n)^\alpha / 3,& \\ \forall 0 \le k_1 < k_2 < 1/b_n\}& \end{aligned} \tag{4.43}$$

Lemma 4.2.8. *The set X is contained in $PGF(b_n) \cap \mathcal{P}_{n,\alpha}$.*

Proof. By definition of $\mathcal{P}_{n,\alpha}$, it is enough to show that if $f \in X$, then

$$\operatorname{dist}_M(f(t_1), f(t_2)) \le M_n |t_2 - t_1|^\alpha, \quad \forall 0 \le t_1 < t_2 \le 1.$$

Suppose $t_1, t_2 \in [kb_n, (k+1)b_n]$ for some k without loss of generality. Since f is geodesic on this interval and $f \in X$,

$$\begin{aligned} \operatorname{dist}_M(f(t_1), f(t_2)) &= \frac{|t_2 - t_1|}{b_n} \operatorname{dist}_M(f(kb_n), f((k+1)b_n)) \\ &\le \frac{|t_2 - t_1|^\alpha}{b_n^\alpha} \operatorname{dist}_M(f(kb_n), f((k+1)b_n)) \\ &\le \frac{M_n}{3} |t_2 - t_1|^\alpha. \end{aligned}$$

Now, let $t_1 \in [k_1 b_n, (k_1+1)b_n]$ and $t_2 \in [k_2 b_n, (k_2+1)b_n]$ for $k_1 < k_2$. By the triangle inequality,

$$\begin{aligned}\mathrm{dist}_M(f(t_1), f(t_2)) &\leq \frac{M_n}{3}|k_1 b_n + b_n - t_1|^\alpha + \frac{M_n}{3}|(k_2 - k_1 - 1)b_n|^\alpha + \frac{M_n}{3}|t_2 - k_1 b_n|^\alpha \\ &\leq M_n |t_2 - t_1|^\alpha.\end{aligned}$$

This completes the proof.

□

Next we consider the upper bound of the probability $\Pi_n(\mathcal{P}_\alpha \backslash \mathcal{P}_{n,\alpha})$. It uses a constant C_2 which is presented in Theorem 5.3.4 in [128, page 141]. It also introduces a constraint on M_n and ϵ_n (see (4.44)).

Lemma 4.2.9. *If* $\frac{1}{2} \leq \alpha \leq 1$, $b_n = M_n^{-c}$ *for a constant* c *s.t.* $0 < c < 1/\alpha$ *and*

$$C_2 \mathrm{Vol}(M) M_n^{c(2D+3)/2} \exp[-M_n^{2-(2\alpha-1)c}/18] \leq \exp[-n\epsilon_n^2 (C+4)], \tag{4.44}$$

then (4.42) *is satisfied.*

Proof. We define

$$X_{k_1,k_2} := \{f \in PGF(b_n) | \, \mathrm{dist}_M(f(k_1 b_n), f(k_2 b_n)) > M_n (k_2 b_n - k_1 b_n)^\alpha / 3\}.$$

When $\alpha \geq \frac{1}{2}$, Theorem 5.3.4 in [128, page 141] implies that for the constant $C_2 > 0$

$$\Pi_n(X_{k_1,k_2}) \leq \frac{C_2}{b_n^{(2D-1)/2}} \exp[-b_n^{2\alpha}(M_n/3)^2/(2b_n)] \mathrm{Vol}(M). \tag{4.45}$$

Consequently,

$$\begin{aligned}\Pi_n(\mathcal{P}_\alpha \backslash \mathcal{P}_{n,\alpha}) &\leq \Pi_n(PGF(b_n) \backslash (PGF(b_n) \cap \mathcal{P}_{n,\alpha})) \qquad (4.46)\\ &\leq \Pi_n(PGF(b_n) \backslash X) \leq \sum_{0 \leq i < j \leq \frac{1}{b_n}} \Pi_n(X_{k_i,k_j}) \\ &\leq \frac{1}{b_n^2} \left(\frac{C_2}{b_n^{(2D-1)/2}} \exp[-b_n^{2\alpha}(M_n/3)^2/(2b_n)] \mathrm{Vol}(M) \right) \\ &= \frac{C_2 \mathrm{Vol}(M)}{b_n^{(2D+3)/2}} \exp[-b_n^{2\alpha-1} M_n^2 / 18].\end{aligned}$$

The first inequality of (4.46) follows from the fact that the support of Π_n is $PGF(b_n)$. The second inequality of (4.46) follows from Lemma 4.2.8. The third inequality follows from the definitions of X and X_{k_i,k_j}. The fourth inequality of (4.46) follows from (4.45). The proof concludes by plugging $b_n = M_n^{-c}$ in (4.46) and the fact

$$\Pi_n(\mathcal{P}_\alpha \backslash \mathcal{P}_{n,\alpha}) = \Pi_n(\mathcal{D}_\alpha \backslash \mathcal{D}_{n,\alpha}).$$

□

4.2.4 Verification of Inequality 2.4 of [127]

We recall that inequality 2.4 of [127, Theorem 2.1] states that

$$\Pi_n\left(P_0\left(\log\frac{p_0}{p}\right) \leq \epsilon_n^2, P_0\left(\log\frac{p_0}{p}\right)^2 \leq \epsilon_n^2\right) \geq \exp[-n\epsilon_n^2 C]. \tag{4.47}$$

We first establish two technical lemmas (Lemmas 4.2.10 and 4.2.11) and then prove (4.47) in Lemma 4.2.12. The formulation of Lemma 4.2.10 requires the following notation. We recall that by choosing a density $p(t)$ on the predictor t, there is a map $\Phi : \mathcal{P} \to \mathcal{D}$. For simplicity, we use the following notation:

$$p(t,x) = \Phi(f) = p_{\sigma^2}(f(t),x)p(t), \quad p_0(t,x) = \Phi(f_0) = p_{\sigma^2}(f_0(t),x)p(t),$$

where f is any continuous function and f_0 is the true function. Let P_0 be the probability with density $p_0(t,x)$ and $P_0 f$ denote $\int f dP_{f_0}$. Here the density $p(t)$ of the predictor t is assumed to be positive on $[0,1]$, so that both $p(t,x)$ and $p_0(t,x)$ are positive (their exact forms are irrelevant). We consider first the upper bounds of $P_0\left(\log\frac{p_0}{p}\right)$ and $P_0\left(\log\frac{p_0}{p}\right)^2$.

Lemma 4.2.10. *There exists a constant $C_3 > 0$ such that*

$$P_0\left(\log\frac{p_0}{p}\right) \leq C_3 d_\infty(f_0,f), \quad P_0\left(\log\frac{p_0}{p}\right)^2 \leq C_3 d_\infty(f_0,f). \tag{4.48}$$

Proof. Theorem 4.1.1 in [128, page 102] states that $p_{\sigma^2}(x,y)$ is strictly positive on $M \times M$. Since $M \times M$ is compact, there exists two constants c_1, $c_2 > 0$ such that

$$c_1 \leq p_{\sigma^2}(x,y) \leq c_2 \tag{4.49}$$

for all $(x, y) \in M \times M$. Moreover, for the same reason, $p_{\sigma^2}(x, y)$ is uniformly continuous. that is, there exists a constant $c_3 > 0$,

$$|p_{\sigma^2}(x_1, x) - p_{\sigma^2}(x_2, x)| \leq c_3 \operatorname{dist}_M(x_1, x_2) \quad \forall x_1, x_2, x \in M. \tag{4.50}$$

Then, the inequality $\log(x) \leq x - 1$, (4.49) and (4.50) imply that

$$\begin{aligned} P_0\left(\log \frac{p_0}{p}\right) &= \iint \log\left(\frac{p_0}{p}\right) p_0 d\mu(x)dt \leq \iint (p_0 - p)\frac{p_0}{p} d\mu(x)dt \\ &\leq \iint \frac{c_2 c_3}{c_1} \operatorname{dist}_M(f(t), f_0(t)) p(t) d\mu(x)dt \leq \frac{c_2 c_3}{c_1} \operatorname{Vol}(M) d_\infty(f_0, f). \end{aligned} \tag{4.51}$$

Similarly,

$$\begin{aligned} P_0\left(\log \frac{p_0}{p}\right)^2 &= \iint \left[\log\left(\frac{p}{p_0}\right)\right]^2 p_0 d\mu(y)dt \\ &= \iint_{p>p_0} \left[\log\left(\frac{p}{p_0}\right)\right]^2 p_0 d\mu(y)dt + \iint_{p<p_0} \left[\log\left(\frac{p_0}{p}\right)\right]^2 p_0 d\mu(y)dt \\ &\leq \iint_{p>p_0} \left(\frac{p - p_0}{p_0}\right)^2 p_0 d\mu(y)dt + \iint_{p<p_0} \left(\frac{p_0 - p}{p}\right)^2 p_0 d\mu(y)dt \\ &\leq \frac{c_3^2}{c_1^2} d_\infty(f_0, f). \end{aligned}$$

Consequently, (4.48) is satisfied with $C_3 = \max(\frac{c_2 c_3}{c_1}\operatorname{Vol}(M), c_3^2/c_1^2)$.

□

Lemma 4.2.11. *Assume that C_3 is an arbitrarily chosen positive constant. If f_0 is a Lipschitz continuous function with the Lipschitz constant $L > 0$ and $f \in PGF(b_n)$ such that $f(kb_n)$ is in the r_n-ball $B(f_0(kb_n), r_n)$ on M, where $r_n = \dfrac{\epsilon_n^2}{3C_3} - \dfrac{2Lb_n}{3}$, then $d_\infty(f_0, f) \leq \epsilon_n^2/C_3$.*

Proof. Since f_0 is Lipschitz,

$$\operatorname{dist}_M(f_0(kb_n), f_0(kb_n + t)) \leq Lb_n \quad \forall 0 \leq k < 1/b_n,\ 0 \leq t \leq b_n. \tag{4.52}$$

Since f is geodesic on each interval $[kb_n, kb_n + b_n]$,

$$\operatorname{dist}_M(f(kb_n), f(t)) \leq \operatorname{dist}_M(f(kb_n), f(kb_n + b_n)) \quad \text{for } t \in [kb_n, kb_n + b_n]. \tag{4.53}$$

By $\text{dist}_M(f_0(kb_n), f(kb_n)) \leq r_n$, (4.52), (4.53) and the triangle inequality,

$$\text{dist}_M(f(kb_n), f(t)) \leq 2r_n + Lb_n. \tag{4.54}$$

Similarly,

$$\text{dist}_M(f(kb_n), f_0(t)) \leq r_n + Lb_n. \tag{4.55}$$

Inequalities (4.54) and (4.55) imply that

$$\text{dist}_M(f_0(t), f(t)) \leq 3r_n + 2Lb_n = \epsilon_n^2/C_3. \tag{4.56}$$

The proof is concluded by the fact that (4.56) is true for every t.

□

Lemma 4.2.12. *If f_0 is a Lipschitz continuous function, then there exists a sufficiently large constant $C_0 > 0$ such that if $b_n = C_0\epsilon_n^2$ and $n\epsilon_n^{4+\delta} \to \infty$ ($\delta > 0$), then the sequence of priors Π_n satisfies* (4.47) *for all $n > N_0$ (N_0 depends on δ).*

Proof. By Lemma 4.2.10, it is enough to show that

$$\Pi_n(f : C_3 d_\infty(f_0, f) \leq \epsilon_n^2) \geq \exp[-n\epsilon_n^2 C], \tag{4.57}$$

where C_3 is the constant in Lemma 4.2.10. Let $f \in PGF(b_n)$ and L be the Lipschitz constant of f_0. It follows from Lemma 4.2.11 that if

$$\text{dist}_M(f_0(kb_n), f(kb_n)) \leq r_n = \frac{\epsilon_n^2}{3C_3} - \frac{2Lb_n}{3} \quad \forall 0 \leq k < 1/b_n, \tag{4.58}$$

then $d_\infty(f_0, f) \leq \epsilon_n^2/C_3$.

Moreover, we note that (4.52), (4.58) and the triangle inequality imply that

$$\text{dist}_M(f(kb_n), f(kb_n + b_n)) \leq Lb_n + 2r_n. \tag{4.59}$$

It follows from Theorem 5.3.4 in [128, page 141] and (4.59) that for a constant $C_4 > 0$,

$$\begin{aligned} p_{b_n}(f(kb_n), f(kb_n + b_n)) &\geq \frac{C_4}{b_n^{D/2}} \exp\left[-\frac{\text{dist}_M(f(kb_n), f(kb_n + b_n))^2}{2b_n}\right] \\ &\geq \frac{C_4}{b_n^{D/2}} \exp\left[-\frac{(Lb_n + 2r_n)^2}{2b_n}\right]. \end{aligned}$$

Recall that the support of Π_n is $PGF(b_n)$. Therefore,

$$\Pi_n(f : C_3 d_\infty(f_0, f) \leq \epsilon_n^2) \leq$$
$$\frac{\mathrm{Vol}(B(f_0(0), r_n))}{\mathrm{Vol}(M)} \prod_{1 \leq k < 1/b_n} \left[\frac{C_4}{b_n^{D/2}} \exp\left[-\frac{(Lb_n + 2r_n)^2}{2b_n} \right] \mathrm{Vol}(B(f_0(kb_n), r_n)) \right]. \tag{4.60}$$

Since

$$\mathrm{Vol}(B(f_0(kb_n), r_n)) \geq C_5 r_n^D \quad \text{for a constant } C_5 > 0,$$

the RHS of (4.60) is at least

$$\frac{1}{\mathrm{Vol(M)}} \left(C_5 r_n^D\right)^{1/b_n + 1} \frac{C_4^{1/b_n}}{b_n^{D/(2b_n)}} \exp\left[-\frac{(Lb_n + 2r_n)^2}{2b_n^2} \right]. \tag{4.61}$$

Plugging the expression of r_n in (4.58) and $C_0 b_n = \epsilon_n^2$ for a constant $C_0 > 0$, the logarithm of (4.61) being greater or equal to $-n\epsilon_n^2 C$ is simplified as

$$\frac{1}{b_n} \left[-\log\left(\frac{C_4 C_5^{1+b_n}}{\mathrm{Vol}(M)^{b_n}} \right) - D(1 + b_n) \log\left(\frac{C_0}{3C_3} - \frac{2L}{3} \right) - (\frac{D}{2} + Db_n) \log(b_n) \right] \tag{4.62}$$
$$+ \frac{1}{2} \left(\frac{2C_0}{3C_3} - \frac{L}{3} \right)^2 \leq n\epsilon_n^2 C.$$

We fix a constant $C_0 > 0$ large enough so that for all b_n,

$$-\log\left(\frac{C_4 C_5^{1+b_n}}{\mathrm{Vol}(M)^{b_n}} \right) - D(1 + b_n) \log\left(\frac{C_0}{3C_3} - \frac{2L}{3} \right) \leq 0.$$

The constant C_0 exists since $b_n \to 0$. Moreover, we note that since the fourth term of (4.62) is a constant, to satisfy (4.62), it is enough to show that

$$-\frac{1}{b_n} \left(\frac{D}{2} + Db_n \right) \log(b_n) \leq \frac{C}{2} n\epsilon_n^2. \tag{4.63}$$

Substituting $C_0 b_n = \epsilon_n^2$ in (4.63) yields the inequality

$$\frac{C_0}{\epsilon_n^2} K \left(\frac{D}{2} + \frac{D}{C_0} \epsilon_n^2 \right) \log\left(\frac{C_0^{1/K}}{\epsilon_n^{2/K}} \right) \leq \frac{C}{2} n\epsilon_n^2. \tag{4.64}$$

We note that by using $\log(x) \leq x$, it is enough to show that

$$K \left(\frac{D}{2} + \frac{D}{C_0} \epsilon_n^2 \right) C_0^{1+1/K} \leq \frac{C}{2} n\epsilon_n^{4+2/K}. \tag{4.65}$$

If we pick any $K > 0$ such that $\dfrac{2}{K} < \delta$, then the right-hand side of (4.65) approaches infinity while the left-hand side is bounded. This implies that there exists a constant $N_0 > 0$ such that for all $n > N_0$, (4.65) is satisfied, which guarantees that (4.57) and thus the lemma are true.

□

4.2.5 Conclusion of Theorem 4.1.1

Under the assumptions that $\frac{1}{2} \leq \alpha \leq 1$, $0 < c < \frac{1}{\alpha}$ and f_0 is Lipschitz, we showed, in previous sections, that if we pick b_n, M_n, ϵ_n such that

$$b_n = M_n^{-c},\ b_n = C_0\epsilon_n^2,\ n\epsilon_n^{4+\delta} \to \infty \text{ and (4.37) \& (4.44) hold,} \tag{4.66}$$

then Theorem 4.1.1 follows directly from [127, Theorem 2.1]. In this section, we conclude the proof by solving the inequalities for parameters and showing the optimal choice of the sequence ϵ_n (which determines the contraction rate).

The first two equalities of (4.66) imply that

$$M_n = C_0^{-1/c}\epsilon_n^{-2/c}. \tag{4.67}$$

Plugging (4.67) into (4.37) and simplifying the expression yields

$$6C_0^{-1/c}C_1 D(\log(21) + 1) \leq n\epsilon_n^{3+2/c}. \tag{4.68}$$

Plugging (4.67) into (4.44) and taking the logarithm of both sides (with simplification) results in the inequality

$$\begin{aligned}&\Big(-\log\Big(C_0^{-(2D+3)/2}C_2\mathrm{Vol}(M)\Big) + (2D+3)\log(\epsilon_n)\Big)\,\epsilon_n^{4/c-4\alpha+2} \\ &+ \frac{1}{18}C_0^{-2/c+2\alpha-1} \geq n\epsilon_n^{4/c-4\alpha+4}(C+4).\end{aligned} \tag{4.69}$$

We note that the first term of (4.69) approaches zero when $4/c - 4\alpha + 2 > 0$. Therefore, to satisfy (4.69), we only need that the second term, which is a constant, is no less than the right-hand side. That is,

$$\frac{1}{18}C_0^{-2/c+2\alpha-1} \geq n\epsilon_n^{4/c-4\alpha+4}(C+4). \tag{4.70}$$

If we pick α, c and ϵ_n so that the right-hand side of (4.70) approaches zero, then (4.69) is satisfied for large n. It follows from (4.68), (4.70) and the fact that $n\epsilon_n^{4+\delta} \to \infty$ that the constants α and c need to satisfy

$$3 + \frac{2}{c} \leq 4 + \delta < \frac{4}{c} - 4\alpha + 4.$$

One choice is $c = \frac{2}{1+\delta}$ and $\alpha = \frac{1}{2}$. Under this choice, the sequence $\epsilon_n = n^{-1/(4+3\delta/2)}$ satisfies (4.68) and (4.70). Since $\delta > 0$ can be arbitrarily small, the best achievable contraction rate is

$$\epsilon_n = n^{-1/4+\epsilon} \quad \text{for any fixed } \epsilon > 0.$$

4.3 Proof of Theorem 4.1.2

We first prove a technical lemma (Lemma 4.3.1) which requires some definitions and then conclude the proof of Theorem 4.1.2. Let $Q_{x_0,\ldots,x_k}^{T,\ldots,kT}$ be the Brownian bridge probability measure on the path space $V^{(k)} = \{f \in C([0,T], M) : f(0) = x_0, f(iT) = x_i, \forall 0 \leq i \leq k\}$. In particular, we denote by $Q_{x,y}^T$ the Brownian bridge probability measure on the path space $V = \{f \in C([0,T], M) : f(0) = x, f(T) = y\}$.

Lemma 4.3.1. *If $x, y \in M$ s.t. $\text{dist}_M(x,y) < \epsilon_0/2$, then there exists $T_0 > 0$ such that*

$$Q_{x,y}^T(\text{dist}_M(f,x) \geq \epsilon_0) < 1, \quad \forall T \leq T_0,$$

where $\text{dist}_M(f,x) = \max_{t\in[0,T]} \text{dist}_M(f(t),x)$. In other words, the Brownian bridge assumes positive measure over the subset of paths $\{f \in V : f([0,T]) \subset B(x,\epsilon_0)\}$.

Proof. Equation 2.6 in [132] implies that there exists $T_0 > 0$ such that if $T \leq T_0$, then

$$T \log(Q_{x,y}^T(\text{dist}_M(f,x) \geq \epsilon_0)) \leq -\epsilon_0^2 + 4\,\text{dist}_M(x,y)^2. \tag{4.71}$$

That is,

$$Q_{x,y}^T(\text{dist}_M(f,x) \geq \epsilon_0) \leq \exp[(-\epsilon_0^2 + 4\,\text{dist}_M(x,y)^2)/T] < 1. \tag{4.72}$$

The first inequality in (4.72) follows from (4.71) and the second inequality follows from the assumption that $\text{dist}_M(x,y) < \epsilon_0/2$.

□

We now conclude the proof of Theorem 4.1.2. Recall that the Kullback-Leibler (KL) divergence between p_{f_0} and p_f is defined as

$$d_{KL}(p_{f_0}, p_f) = \int_{[0,1]\times M} p_{f_0} \log\left(\frac{p_{f_0}}{p_f}\right) dt d\mu(x).$$

A corollary of Theorem 6.1 in [133] implies that if Π assumes positive mass on any Kullback-Leibler neighborhood of p_{f_0}, then the posterior distribution is weakly consistent. Thus, it is enough to show that

$$\Pi(\{f : d_{KL}(p_{f_0}, p_f) \leq \epsilon\}) > 0, \quad \forall \epsilon > 0.$$

We note that Lemma 4.2.10 shows that $d_{KL}(p_{f_0}, p_f)$ is upper bounded by $d_\infty(f_0, f)$. Therefore, we only need to prove that

$$\Pi(\{f : d_\infty(f_0, f) \leq \epsilon\}) > 0, \quad \forall \epsilon > 0.$$

Fix a positive number $\epsilon_1 < \epsilon$. We consider a regular (e.g., equidistant) grid of $[0, 1]$ with spacing T. We assume the regular grid satisfies the following conditions:

1. $B(x_i, \epsilon_1) \subset B(x, \epsilon)$, $\forall x \in f_0([iT, (i+1)T])$ and $x_i = f_0(iT)$,
2. $\text{dist}_M(x_i, x_{i+1}) < \epsilon_1/4$.

The Lipschitz assumption of f_0 guarantees the existence of T. Indeed, Condition (1) is guaranteed by the triangle inequality of the metric dist_M and the Lipschitz assumption and Condition (2) is guaranteed by picking a sufficiently small T.

Given a positive number $\delta < \epsilon_1/24$, the triangle inequality implies that

$$B(\hat{x}_i, 2\epsilon_1/3) \subset B(x_i, \epsilon_1) \text{ and } \text{dist}_M(\hat{x}_i, \hat{x}_{i+1}) < \epsilon_1/3, \quad \forall \hat{x}_i \in B(x_i, \delta). \tag{4.73}$$

Applying Lemma 4.3.1 to $\hat{x}_i$ and $\hat{x}_{i+1}$ implies that $Q^{T,\ldots,1}_{\hat{x}_0,\ldots,\hat{x}_{1/T}}$ assumes positive measure over the set of paths

$$V_{\hat{\mathbf{x}}} = \{f : f(0) = x_0, f(iT) = \hat{x}_i, f([iT, (i+1)T]) \in B(\hat{x}_i, 2\epsilon_1/3),\ \forall i \in [0, 1/T]\}.$$

If $f \in V_{\hat{\mathbf{x}}}$, then for any $t \in [iT, (i+1)T]$,

$$f(t) \in B(\hat{x}_i, 2\epsilon_1/3) \subset B(x_i, \epsilon_1) \subset B(f_0(t), \epsilon). \tag{4.74}$$

The first inclusion in (4.74) follows from (4.73) and the second inclusion in (4.74) follows from condition (1) of the regular grid. By definition, (4.74) implies that $V_{\hat{\mathbf{x}}} \subset \{f : d_\infty(f_0, f) \leq \epsilon\}$. Therefore,

$$\Pi(\{f : d_\infty(f_0, f) \leq \epsilon\}) \geq \int_{\hat{\mathbf{x}} \in \prod_{i=0}^{1/T} B(x_i,\delta)} Q^{T,\dots,1}_{\hat{x}_0,\dots,\hat{x}_{1/T}}(V_{\hat{\mathbf{x}}})\Pi_n(d\hat{\mathbf{x}}) > 0,$$

where Π_n is the probability measure of the discretized Brownian motion with spacing $b_n = T$ and $\hat{\mathbf{x}} = (\hat{x}_1, \dots, \hat{x}_{1/T})^T$.

4.4 Extensions of the Regression Framework

In this section, we briefly discuss two extensions of the current framework, where Theorem 4.1.1 and 4.1.2 equally apply. In Section 4.4.1, we consider the case where the variance σ^2 is unknown. Section 4.4.2 explains how to possibly relax the assumption that $p(t)$ has a positive lower bound.

4.4.1 The Case of Unknown Variance σ^2

The mapping Φ of (4.7) assumes that σ^2 is a fixed and known parameter. If it is unknown, the prior on it can be chosen as the uniform distribution on the interval $[1/A, A]$ for some constant $A > 0$ (or other distributions as long as it is bounded away from zero and infinity).

Under this prior of σ^2, the probability density of (t, x) is given by

$$p_f(t, x) = \int_{\sigma^2 \in [1/A, A]} p_{\sigma^2}(f(t), x)p(t)d\sigma^2.$$

Since $p_{\sigma^2}(x, y)$ and its partial derivatives (w.r.t. x and y) are uniformly continuous in the variable σ^2 over the interval $[1/A, A]$, it is easy to see that Lemmas 4.2.1 and 4.2.10 still hold for this type of probability densities. Therefore, the contraction rate for the case of unknown variance is the same as the case of fixed variance.

4.4.2 More General $p(t)$

Throughout the paper, we assume that the distribution of the predictor t has a smooth density $p(t)$ on $[0, 1]$ with strict lower and upper bounds $0 < m_p \leq M_p$. This assumption is used in Lemma 4.2.1. Since $p(t)$ is continuous, the upper bound M_p always exists, but the lower bound can be restrictive. We can relax the lower bound on $p(t)$ as follows. Let $r > 0$ and $S_r = \{t \in [0,1] | p(t) \geq r\}$. By following the same arguments in the proof, we note that the posterior distribution contracts at the same rate to the true function when considering the L_q norm of functions restricted to S_r.

4.5 Numerical Demonstrations

In this section, we demonstrate the proposed Bayesian scheme and compare it with a kernel method for the simple manifold $\mathbb{S}^1$. We also investigate the effect of changing various parameters for this special case.

One reason of using $\mathbb{S}^1$ is its simplicity of visualization. Indeed, $\mathbb{S}^1$ can be identified with the interval $[0, 2\pi]$ and this makes it easy to plot the $\mathbb{S}^1$-valued functions. The other reason is that $\mathbb{S}^1$, as a Lie group, has the addition operator on it. Thus, the kernel method in Euclidean spaces directly applies to this situation, with special awareness of the issue of averaging (more specifically, the average of the points 0 and 2π on $\mathbb{S}^1$ is 0, not π).

For the discretized and continuous BM Bayesian schemes, we obtain the maximum a posteriori (MAP) probability estimators by implementing a simulated annealing (SA) algorithm on the corresponding posterior distributions. The starting state (function) of SA is defined as follows: the value $f(t)$ at time t is the mode of all observed values, whose observation times are in $[t - 0.05, t + 0.05]$. For the discretized BM Bayesian scheme, the sidelength parameter b_n is fixed to be 1/40. For the kernel method, we use the Matlab code [134] implemented according to the Nadaraya-Watson kernel regression with the optimal bandwidth suggested by Bowman and Azzalini [135].

We remark that we use Brownian motion of various scales and not the standard one, BM_t, assumed in the proof. Nevertheless, the convergence result clearly holds for any scaled Brownian motion BM_{ct}, where $c > 0$. In fact, c is an additional hyperparameter (see Section 4.5.2).

4.5.1 Comparison with Kernel Regression

In the first experiment, we compare three estimators, namely, the discretized BM MAP (DBM) estimator, the continuous BM MAP (CBM) estimator and the kernel regression estimator (KER). We fix the scaling hyperparameter $c = 0.01$ for DBM and CBM and the optimal bandwidth for KER. We generate datasets of 30 observations according to the pdf $p_{f_0(t)}(x)$ defined in (4.2), where $\sigma^2 = 0.1$ and $f_0 : [0,1] \to \mathbb{S}^1$ defined by

$$f_0(t) := (t + 0.5)^2, \quad \text{for } t \in [0,1].$$

Figure 4.1 shows the original function and its different estimators according to DBM, CBM and KER. The L_1 errors between the estimated functions and the true function are also displayed. Among them, the CBM achieves the minimal L_1 error.

4.5.2 The Hyperparameter c

The hyperparameter c plays a similar role as the hyperparameter in the regularized regression. The second experiment shows how the hyperparameter c (with values in $\{0.01, 0.1, 1, 10\}$) affects the estimation. We fix a dataset of 40 observations with noise variance 0.05 from the same function as in the first experiment. Figures 4.2 and 4.3 demonstrate the MAP estimators obtained by CBM and DBM respectively. In both figures, the estimators become smoother when c decreases. Indeed, smaller c means shorter time for the BM to travel. But smaller c also introduces more bias in the estimators. This is more evident for DBM in Figure 4.3 while CBM seems less sensitive to small values of c (see Figure 4.2).

4.5.3 The Sidelength Parameter b_n

For DBM we have another important parameter, b_n, which determines the number of pieces of a piecewise geodesic function. When $b_n = 1$, the piecewise geodesic function becomes geodesic. In this experiment, we show the change of L_1 error of the DBM estimator for different choices of b_n ($1/b_n$ ranges from 1 to 100). The data set is generated from the same model as in the first experiment. Figure 4.4 shows that for geodesic functions or functions with large b_n, there is a large L_1 error due to large bias. As b_n becomes smaller, there is a steady decrease of the L_1 error due to the decrease of bias.

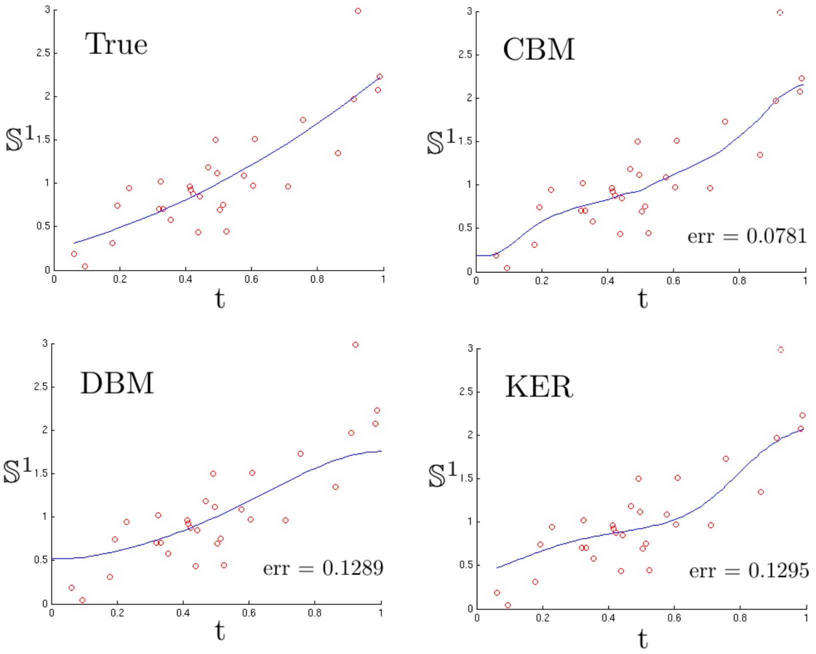


Figure 4.1: Demonstration of the continuous and discretized BM Bayesian estimators with comparison to a kernel estimator. The data was generated according to the pdf $p_{f_0(t)}(x)$, where f_0 is demonstrated in the top left subfigure. The estimators obtained by CBM (continuous Brownian motion), DBM (discretized Brownian motion) and KER (kernel method) are shown in the rest of the subfigures together with their L_1 errors.

4.6 Conclusion

We established the consistency of the Bayesian estimator with a Brownian motion prior in the manifold regression setting. For the discretized Brownian motion, we even specified a contraction rate via a well-known general approach [127, 136]. We thus propose a new nonparametric Bayesian framework with solid statistical analysis beyond the existing kernel methods and Gaussian process priors. In fact, one of our motivations to this work is the incapability of applying a Gaussian process prior to manifold responses that lack linear structure.

We also list a few interesting questions for possible future study.

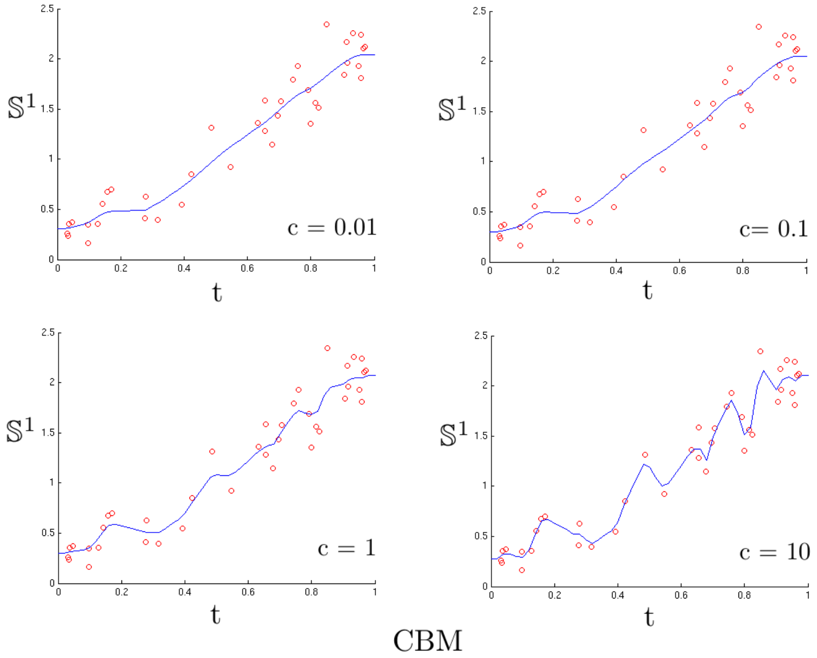


Figure 4.2: The estimators obtained by CBM for different values of c ($c = 0.01, 0.1, 1, 10$), where c is the scaling parameter of the Brownian motion.

4.6.1 Better Quantitative Estimate of C_0 and C_1

The constants C_0 and C_1 in Lemma 4.2.1 (comparing the distance of functions and the distance of distributions) are not specified due to our proof by contradiction. The specification of their dependencies on the underlying Riemannian geometry worth further investigation.

4.6.2 L_∞ Convergence

We only proved L_p-convergence for the Brownian motion prior. It is interesting to investigate the L_∞ convergence if it exists at all. If it does not exist, then it is interesting to know if a smoother prior (e.g., integrated BM) has L_∞ convergence.

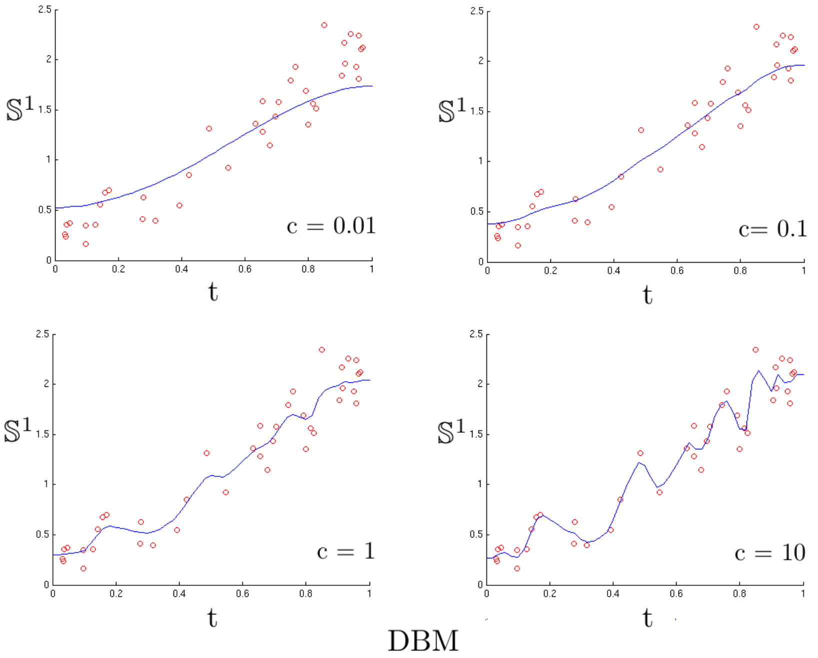


Figure 4.3: The estimators obtained by DBM for different values of c ($c = 0.01, 0.1, 1, 10$), where c is the scaling parameter of the Brownian motion. Unlike CBM, an underestimation (i.e., sensitivity to bias) is observed when $c = 0.01$.

4.6.3 A Better Contraction Rate?

For regression with real-valued predictors and responses, van Zanten [125] established posterior contraction rate of $n^{-1/4}$ for n samples under the L_q-norm, where $1 \leq q < \infty$. His analysis does not seem to extend to our setting. It is possible that even for the general case of manifold-valued regression the contraction rate is $n^{-1/4}$ and not just $n^{-1/4+\epsilon}$. The particular method used here does not seem to obtain a better rate.

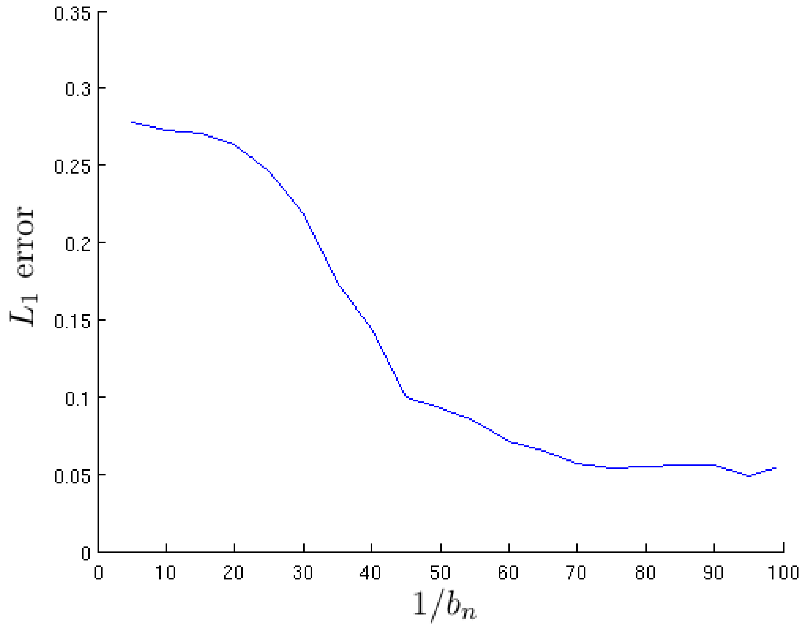


Figure 4.4: L_1 error of DBM for different sidelengths b_n.

References

[1] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[2] K. Clarkson. A randomized algorithm for closest-point queries. *SICOMP*, 17:830–847, 1988.

[3] S. Har-Peled, P. Indyk, and R. Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of computing*, 8:321–350, 2012.

[4] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[5] R. Basri, T. Hassner, and L. Zelnik-Manor. Approximate nearest subspace search. *TPAMI*, 33(2), 2011.

[6] A. Magen. Dimensionality reductions that preserve volumes and distance to affine spaces, and their algorithmic applications. *RANDOM*, 2002.

[7] P. Jain, S. Vijayanarasimhan, and K. Grauman. Hashing Hyperplane Queries to Near Points with Applications to Large-Scale Active Learning. In *NIPS*, 2010.

[8] W. Liu, J. Wang, Y. Mu, S. Kumar, and S. Chang. Compact hyperplane hashing with bilinear functions. In *International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, 2012.

[9] Y. Mu, J. Wright, and S. Chang. Accelerated large scale optimization by concomitant hashing. In *Computer Vision–ECCV*, pages 414–427. Springer, 2012.

[10] A. Andoni, P. Indyk, R. Krauthgamer, and H. L. Nguyen. Approximate line nearest neighbor in high dimensions. *SODA*, 2009.

[11] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. *TPAMI*, 25(2):218–233, February 2003.

[12] A. Yang, J. Wright, Y. Ma, and S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *Comput. Vis. Image Underst.*, 110(2), May 2008.

[13] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity. *Image Processing, IEEE Transactions on*, 21(5):2481–2499, 2012.

[14] G. H. Golub and C. F. Van Loan. Matrix computations, 3rd edition. *Baltimore: Johns Hopkins University Press*, 1996.

[15] A. Andoni and P. Indyk. Near optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Comm. of the ACM*, 51(1), 2008.

[16] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[17] P. Ndajah, H. Kikuchi, M. Yukawa, H. Watanabe, and S. Muramatsu. SSIM image quality metric for denoised images. In *VIS 10 Proceedings of the 3rd WSEAS international conference on Visualization, imaging and simulation*, pages 53–57, 2010.

[18] R. Epstein, P. Hallinan, and A. Yuille. 5 ± 2 eigenimages suffice: An empirical investigation of low-dimensional lighting models. In *IEEE PBMCV*, June 1995.

[19] J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *CVPR*, 2003.

[20] R. Gross, I. Matthews, J.F. Cohn, T. Kanade, and S. Baker. Multi-pie. *Proceedings of the Eighth IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.

[21] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *TPAMI*, 23(6):643–660, 2001.

[22] K.C. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *TPAMI*, 27(5):684–698, 2005.

[23] E. Arias-Castro, G. Chen, and G. Lerman. Spectral clustering based on local linear approximations. *Electron. J. Statist.*, 5:1537–1587, 2011.

[24] E. Arias-Castro, G. Lerman, and T. Zhang. Spectral clustering based on local PCA. *ArXiv e-prints*, 2013.

[25] H. E. Cetingul and R. Vidal. Intrinsic mean shift for clustering on Stiefel and Grassmann manifolds. In *Proc. CVPR*, pages 1896–1902, June 2009.

[26] E. Elhamifar and R. Vidal. Sparse manifold clustering and embedding. In *Proc. NIPS*, pages 55–63, 2011.

[27] D. Kushnir, M. Galun, and A. Brandt. Fast multiscale clustering and manifold identification. *Pattern Recognition*, 39(10):1876–1891, 2006.

[28] J. Ho, Y. Xie, and B. Vemuri. On a nonlinear generalization of sparse coding and dictionary learning. In *Proc. ICML*, volume 28, pages 1480–1488, 2013.

[29] Y. M. Lui. Advances in matrix manifolds for computer vision. *Image Vision Comput.*, 30(6-7):380–388, 2012.

[30] Y. Wang, Y. Jiang, Y. Wu, and Z.-H. Zhou. Spectral clustering on multiple manifolds. *IEEE Trans. Neural Networks*, 22(7):1149–1161, 2011.

[31] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Trans. Pattern Analysis Machine Intell.*, 33(11):2273–2286, 2011.

[32] J. Y. Tou, Y. H. Tay, and P. Y. Lau. Gabor filters as feature images for covariance matrix on texture classification problem. In *Advances in Neuro-Information Processing*, volume 5507 of *Lecture Notes in Computer Science*, pages 745–751. Springer Berlin Heidelberg, 2009.

[33] G. Chen and G. Lerman. Spectral curvature clustering (SCC). *Int. J. Comput. Vision*, 81:317–330, 2009.

[34] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *Proc. CVPR*, 2009.

[35] A. Goh and R. Vidal. Clustering and dimensionality reduction on Riemannian manifolds. In *Proc. CVPR*, pages 1–7, June 2008.

[36] A. Goldberg, X. Zhu, A. Singh, Z. Xu, and R. Nowak. Multi-manifold semi-supervised learning. In *Proc. CVPR*, volume 5, pages 169–176, 2009.

[37] M. T. Harandi, C. Sanderson, C. Shen, and B. C. Lovell. Dictionary learning and sparse coding on Grassmann manifolds: An extrinsic solution. In *Proc. ICCV*, 2013.

[38] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Analysis Machine Intell.*, 35, 2013.

[39] T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Hybrid linear modeling via local best-fit flats. *Intern. J. Computer Vision*, 100:217–240, 2012.

[40] P. Gruber and F. J. Theis. Grassmann clustering. In *Proc. EUSIPCO*, 2006.

[41] O. Tuzel, R. Subbarao, and P. Meer. Simultaneous multiple 3D motion estimation via mode finding on Lie groups. In *Proc. ICCV*, volume 1, pages 18–25, Oct. 2005.

[42] R. Subbarao and P. Meer. Nonlinear mean shift for clustering over analytic manifolds. In *Proc. CVPR*, volume 1, pages 1168–1175, June 2006.

[43] S. O'Hara, Y. M. Lui, and B. A. Draper. Unsupervised learning of human expressions, gestures, and actions. In *Proc. Automatic Face Gesture Recognition and Workshops*, pages 1–8, March 2011.

[44] Y. Rathi, A. Tannenbaum, and O. V. Michailovich. Segmenting images on the tensor manifold. In *CVPR*, 2007.

[45] P. Bradley and O. Mangasarian. k-plane clustering. *J. Global optim.*, 16(1):23–32, 2000.

[46] Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1546–1562, September 2007.

[47] B. Poling and G. Lerman. A new approach to two-view motion segmentation using global dimension minimization. *International Journal of Computer Vision*, 108(3):165–185, 2014.

[48] P. Tseng. Nearest q-flat to m points. *Journal of Optimization Theory and Applications*, 105:249–252, 2000.

[49] T. Zhang, A. Szlam, and G. Lerman. Median K-flats for hybrid linear modeling with many outliers. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on Computer Vision*, pages 234–241, Kyoto, Japan, 2009.

[50] T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Randomized hybrid linear modeling by local best-fit flats. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1927–1934, 2010.

[51] T. E. Boult and L. G. Brown. Factorization-based segmentation of motions. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 179–186, 1991.

[52] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[53] K. Kanatani. Motion segmentation by subspace separation and model selection. In *Proc. of 8th ICCV*, volume 3, pages 586–591. Vancouver, Canada, 2001.

[54] K. Kanatani. Evaluation and selection of models for motion segmentation. In *7th ECCV*, volume 3, pages 335–349, May 2002.

[55] Y. Ma, A. Y. Yang, H. Derksen, and R. Fossum. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Review*, 50(3):413–458, 2008.

[56] N. Ozay, M. Sznaier, C. Lagoa, and O. Camps. GPCA with denoising: A moments-based convex approach. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3209–3216, 2010.

[57] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12), 2005.

[58] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482, 1999.

[59] A. Y. Yang, S. R. Rao, and Y. Ma. Robust statistical estimation and segmentation of multiple subspaces. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 99, Washington, DC, USA, 2006. IEEE Computer Society.

[60] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PP(99):1–1, 2013.

[61] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and nondegenerate. In *ECCV*, volume 4, pages 94–106, 2006.

[62] R. Vidal. Subspace clustering. *SPM*, 28:52 –68, 2011.

[63] A. Aldroubi. A review of subspace segmentation: Problem, nonlinear approximations, and applications to motion segmentation. *ISRN Signal Processing*, 2013(Article ID 417492):1–13, 2013.

[64] G. Chen and G. Lerman. Foundations of a multi-way spectral clustering framework for hybrid linear modeling. *Found. Comput. Math.*, 9(5):517–558, 2009.

[65] G. Lerman and T. Zhang. Robust recovery of multiple subspaces by geometric l_p minimization. *Annals Statist.*, 39(5):2686–2715, 2011.

[66] M. Soltanolkotabi and E. J. Candès. A geometric analysis of subspace clustering with outliers. *Ann. Stat.*, 40(4):2195–2238, 2012.

[67] M. Soltanolkotabi, E. Elhamifar, and E. J. Candès. Robust subspace clustering. *Ann. Statist.*, 42(2):669–699, 2014.

[68] D. Gong, X. Zhao, and G. Medioni. Robust multiple manifolds structure learning. In *Proc. ICML*, pages 321–328, 2012.

[69] H. E. Cetingul, M. J. Wright, P. M. Thompson, and R. Vidal. Segmentation of high angular resolution diffusion MRI using sparse Riemannian manifold clustering. *IEEE Trans. Medical Imaging*, 33(2):301–317, Feb. 2014.

[70] Q. Guo, H. Li, W. Chen, I-F. Shen, and J. Parkkinen. Manifold clustering via energy minimization. In *Proc. ICMLA*, pages 375–380, 2007.

[71] M. Polito and P. Perona. Grouping and dimensionality reduction by locally linear embedding. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, pages 1255–1262, 2001.

[72] R. Souvenir and R. Pless. Manifold clustering. In *Proceedings of the 10th International Conference on Computer Vision (ICCV 2005)*, volume 1, pages 648–653, 2005.

[73] D. Barbará and P. Chen. Using the fractal dimension to cluster datasets. In *KDD*, pages 260–264, 2000.

[74] A. Gionis, A. Hinneburg, S. Papadimitriou, and P. Tsaparas. Dimension induced clustering. In *KDD*, pages 51–60, 2005.

[75] G. Haro, G. Randall, and G. Sapiro. Stratification learning: Detecting mixed density and dimensionality in high dimensional point clouds. *Neural Information Processing Systems*, 2006.

[76] I. U. Rahman, I. Drori, V. C. Stodden, D. L. Donoho, and P. Schröder. Multiscale representations for manifold-valued data. *Multiscale Model. Simul.*, 4(4):1201–1232 (electronic), 2005.

[77] M. P. do Carmo. *Riemannian Geometry.* Birkhäuser, Boston, 1992.

[78] G. Lerman and J. T. Whitehouse. On d-dimensional d-semimetrics and simplex-type inequalities for high-dimensional sine functions. *J. Approx. Theory*, 156(1):52–81, 2009.

[79] F. Mémoli and G. Sapiro. Distance functions and geodesics on submanifolds of rd and point clouds. *SIAM Journal of Applied Mathematics*, 65(4):1227–1260, 2005.

[80] E. J. Canales-Rodriguez, L. Melie-Garcia, Y. Iturria-Medina, and Y. Aleman-Gomez. Website, 2013. `http://neuroimagen.es/webs/hardi_tools/`.

[81] T. Randen. Website, 2014. `http://www.ux.uis.no/~tranden/brodatz.html`.

[82] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *Intern. J. Computer Vision*, 51(2):91–109, 2003.

[83] B. Ghanem and N. Ahuja. Maximum margin distance learning for dynamic texture recognition. In *Proc. ECCV (2)*, pages 223–236, 2010.

[84] R. Basri, T. Hassner, and L. Zelnik-Manor. Approximate nearest subspace search. *IEEE Trans. Pattern Analysis Machine Intell.*, 33(2):266–278, 2011.

[85] Y. Wang and G. Mori. Human action recognition by semilatent topic models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(10):1762–1774, 2009.

[86] L. Zelnik-Manor and P. Perona. Self-Tuning Spectral Clustering. In *Proceedings of the 18th Annual Conference on Neural Information Processing Systems (NIPS'04)*, 2004.

[87] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

[88] G. W. Stewart and J. G. Sun. *Matrix perturbation theory.* Computer Science and Scientific Computing. Academic Press Inc., Boston, MA, 1990.

[89] C. Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM J. Numerical Analysis*, 7:1–46, Mar. 1970.

[90] A. Gray. Comparison theorems for the volumes of tubes as generalizations of the weyl tube formula. *Topology*, 21(2):201 – 228, 1982.

[91] N. Nikvand. Website, 2013. `https://ece.uwaterloo.ca/~nnikvand/Coderep/spectralclustering-1.1/`.

[92] K. Gallivan, A. Srivastava, X. Liu, and P. V. Dooren. Efficient algorithms for inferences on Grassmann manifolds. In *Proc. SSP*, pages 315–318, 2003.

[93] J. Ho, Y. Xie, and B. C. Vemuri. On a nonlinear generalization of sparse coding and dictionary learning. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1480–1488, 2013.

[94] R. Glowinski and A. Marrocco. Sur l'approximation par éléments finis et la résolution par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *Rev. Francaise d'Aut. Inf. Rech. Oper.*, 9(2):41–76, 1975.

[95] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite-element approximations. *Comp. Math. Appl.*, 2:17–40, 1976.

[96] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, New York, 2011.

[97] A. Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM Journal on Scientific Computing*, 23(2):517–541, 2001.

[98] D. Kushnir, M. Galun, and A. Brandt. Efficient multilevel eigensolvers with applications to data analysis tasks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1377–1391, 2010.

[99] X. Wang, S. Atev, J. Wright, and G. Lerman. Fast subspace search via grassmannian based hashing. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2776–2783, Dec 2013.

[100] G. Lerman and T. Zhang. ℓ_p-recovery of the most significant subspace among multiple subspaces with outliers. *Constructive Approximation*, pages 1–57, 2014.

[101] K. V. Mardia and P. E. Jupp. *Directional Statistics*. Wiley Series in Probability and Statistics. Wiley, 2009.

[102] N. I. Fisher, T. Lewis, and B. J. J. Embleton. *Statistical analysis of spherical data*. Cambridge University Press, 1993.

[103] N. I. Fisher. *Statistical Analysis of Circular Data*. Cambridge University Press, 1995.

[104] G. N. Murshudov, P. Skubák, A. A. Lebedev, N. S. Pannu, R. A. Steiner, R. A. Nicholls, M. D. Winn, F. Long, and A. A. Vagin. *REFMAC*5 for the refinement of macromolecular crystal structures. *Acta Crystallographica Section D*, 67(4):355–367, Apr 2011.

[105] M. D. Shuster and S. D. Oh. Three-axis attitude determination from vector observations. *Journal of Guidance Control and Dynamics*, 4:70–77, 1981.

[106] J. V. Olsen, M. Vermeulen, A. Santamaria, C. Kumar, M. L. Miller, L. J. Jensen, F. Gnad, J. Cox, T. S. Jensen, E. A. Nigg, S. Brunak, and M. Mann. Quantitative phosphoproteomics reveals widespread full phosphorylation site occupancy during mitosis. *Science Signaling*, 3(104):ra3–ra3, 2010.

[107] X. Miao and R. P. N. Rao. Learning the lie groups of visual invariance. *Neural Comput.*, 19(10):2665–2693, October 2007.

[108] R. Webster and M. A. Oliver. *Geostatistics for Environmental Scientists*. John Wiley & Sons, Ltd, 2008.

[109] J. Fishbaugh, M. Prastawa, G. Gerig, and S. Durrleman. Geodesic regression of image and shape data for improved modeling of 4d trajectories. In *IEEE 11th International Symposium on Biomedical Imaging*, pages 385–388, 2014.

[110] F. Nielsen. *Emerging Trends in Visual Computing: LIX Fall Colloquium, ETVC 2008, Palaiseau, France.* Springer, 2009.

[111] Y. Hong, Y. Shi, M. Styner, M. Sanchez, and M. Niethammer. Simple geodesic regression for image time-series. In *Biomedical Image Registration*, volume 7359 of *Lecture Notes in Computer Science*, pages 11–20. Springer Berlin Heidelberg, 2012.

[112] A. Bhattacharya and R. Bhattacharya. *Nonparametric Inference on Manifolds: With Applications to Shape Spaces.* Cambridge University Press, New York, NY, USA, 2012.

[113] M. Niethammer, Y. Huang, and F. X. Vialard. Geodesic regression for image time-series. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 2011*, volume 6892 of *Lecture Notes in Computer Science*, pages 655–662. Springer Berlin Heidelberg, 2011.

[114] J. Fishbaugh, M. Prastawa, G. Gerig, and S. Durrleman. Geodesic shape regression in the framework of currents. In *Information Processing in Medical Imaging*, volume 7917 of *Lecture Notes in Computer Science*, pages 718–729. Springer Berlin Heidelberg, 2013.

[115] J. Hinkle, P. T. Fletcher, and S. Joshi. Intrinsic polynomials for regression on riemannian manifolds. *Journal of Mathematical Imaging and Vision*, 50(1-2):32–52, 2014.

[116] A. Aswani, P. Bickel, and C. Tomlin. Regression on manifolds: Estimation of the exterior derivative. *Ann. Statist.*, 39(1):48–81, 2011.

[117] F. Cucker and S. Smale. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39:1–49, 2002.

[118] J. Nilsson, F. Sha, and M. I. Jordan. Regression on manifolds using kernel dimension reduction. In *ICML 2007, Corvallis, Oregon, USA*, pages 697–704, 2007.

[119] R. Calandra, J. Perters, C. E. Rasmussen, and M. P. Deisenroth. Manifold gaussian processes for regression. *ArXiv e-prints*, 2014.

[120] B. Pelletier. Nonparametric regression estimation on closed riemannian manifolds. *J. of Nonparametric Stat.*, 18:57–67, 2006.

[121] Y. Yang and D. B. Dunson. Bayesian manifold regression. *ArXiv e-prints*, 2014.

[122] M.-Y. Cheng and H.-T. Wu. Local linear regression on manifolds and its geometric interpretation. *J. Amer. Statist. Assoc.*, 108(504):1421–1434, 2013.

[123] P. T. Fletcher. Geodesic regression and the theory of least squares on riemannian manifolds. *International Journal of Computer Vision*, 105(2):171–185, 2013.

[124] M. Hein. Robust nonparametric regression with metric-space valued output. In *Advances in Neural Information Processing Systems*, pages 718–726, 2009.

[125] H. van Zanten. On Brownian motion as a prior for nonparametric regression. *Statist. Decisions*, 27(4):335–356, 2009.

[126] C. Bär and F. Pfäffle. Wiener measures on Riemannian manifolds and the Feynman-Kac formula. *Mat. Contemp.*, 40:37–90, 2011.

[127] S. Ghosal, J. K. Ghosh, and A. W. van der Vaart. Convergence rates of posterior distributions. *The Annals of Statistics*, 28(2):500–531, 04 2000.

[128] E. P. Hsu. *Stochastic analysis on manifolds*, volume 38 of *Graduate Studies in Mathematics*. American Mathematical Society, 2002.

[129] J. Nash. c^1-isometric imbeddings. *The Annals of Mathematics*, 60(3):383–396, 1954.

[130] H. Whitney. The self-intersections of a smooth n-manifold in $2n$-space. *The Annals of Mathematics*, 45(2):220–246, 1944.

[131] X. Wang, K. Slavakis, and G. Lerman. Clustering geodesic riemannian submanifolds. *ArXiv e-prints*, 2014.

[132] P. Hsu. Brownian bridges on riemannian manifolds. *Probability Theory and Related Fields*, 84(1):103–118, 1990.

[133] L. Schwartz. On bayes procedures. *Zeitschrift fr Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 4(1):10–26, 1965.

[134] Y. Cao. Website, 2008. `http://www.mathworks.com/matlabcentral/fileexchange/19195-kernel-smoothing-regression`.

[135] A. W. Bowman. *Applied smoothing techniques for data analysis*. Oxford University Press, 1997.

[136] A. W. van der Vaart and J. H. van Zanten. Rates of contraction of posterior distributions based on Gaussian process priors. *Ann. Statist.*, 36(3):1435–1463, 2008.