Geometric algorithms for interpretable manifold learning

Samson Koelle

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2021

Reading Committee:
Marina Meila, Chair
Yen-Chi Chen
Zaid Harchaoui

Program Authorized to Offer Degree:
University of Washington Department of Statistics

University of Washington

Abstract

Geometric algorithms for interpretable manifold learning

Samson Koelle

Chair of the Supervisory Committee:
Professor Marina Meila
Department of Statistics

This thesis proposes several algorithms in the area of interpretable unsupervised learning. Chapters 3 and 4 introduce a sparse convex regression approach for identifying local diffeomorphisms from a dictionary of interpretable functions. In Chapter 3, this algorithm makes use of an embedding learned by a manifold learning algorithm, while in Chapter 4, this algorithm is applied without the use of a precomputed embedding. Chapter 5 then introduces a set of alternative algorithms that avoid issues stemming from sparse regression, characterizes the tangent space version of this algorithm as identifying isometries when available, and gives a two-stage algorithm combining this approach with the computational advantages of the algorithms in Chapters 3 and 4. Finally, Chapter 6 gives an alternate tangent space estimator based on a learned embedding, and uses this as an initial estimator to tackle the related gradient estimation problem. Together, these approaches provide a toolbox of methods for computing and associating gradient information to learn descriptive parameterizations of data manifolds.

TABLE OF CONTENTS

LIST OF AlgorithmS

LIST OF FIGURES

Figure Number Page

GLOSSARY

: $\mathbb{R}^D$ - the feature space in which data are observed

: $\mathcal{M}$ - a d or $d_{\mathcal{M}}$ smooth submanifold of $\mathbb{R}^D$

: $\mathcal{E}$ - a $d_{\mathcal{E}}$ dimensional noise manifold

: $\xi \in \mathcal{M}$ - a data point

: i - an index over samples

: n - the number of samples

: $[n]$ - the index set $[1, \ldots, n]$

: $\mathcal{D}$ - the observed data set $\xi_{1:n}$

: ϕ - a learned embedding of $\mathcal{M}$

: ϕ^k - the coordinate functions of this embedding

: m - the number of embedding coordinates

: Φ - $\phi(\mathcal{D})$

: j - an index over covariate functions

: $\mathcal{G} = \{g^1 \ldots g^p\}$ a dictionary of smooth covariate functions

: $p = |\mathcal{G}|$ - the number of covariate functions

: $[p]_d$ - a size d set of elements sampled from $[p]$ without replacement.

: S - the functional support of $\mathcal{M}$

: $\mathcal{T}_\xi \mathcal{M}$ - a tangent space to $\mathcal{M}$ at point ξ

: $T_\xi^{\mathcal{M}} \in \mathbb{R}^{D \times d}$ - an orthonormal basis for this space

: f - a smooth function from $\mathcal{M}$ to $\mathbb{R}$

: $\operatorname{grad}_{\mathcal{M}} f$ - the coordinate free gradient operator

: $X \in \mathbb{R}^{n \times d \times p}$ - an array of vectors (usually $\operatorname{grad}_{T_i^{\mathcal{M}}} g^j$)

: $Y \in \mathbb{R}^{n \times d \times m}$ - an array of vectors (usually $\operatorname{grad}_{T_i^{\mathcal{M}}} \phi^k$)

: $\mathcal{N}_i \subset [n]$ - the neighbors of ξ_i

: Ξ_i - $[\xi_{i'} : i \in \mathcal{N}_i]$

: δ_i^{ξ} - $[\xi_{i'} : i \in \mathcal{N}_i] \in \mathbb{R}^{k_i \times D}$

: $X_{i..} \in \mathbb{R}^{d \times p}, X_{.k.} \in \mathbb{R}^{n \times p}$ - example of array slicing notation

ACKNOWLEDGMENTS

I'd like to thank my advisor, Marina Meila, who provided invaluable inspiration, commitment, and support. This research program was your well-developed idea, and I am happy that I was able to contribute to it! I'd like to thank my committee members Yen-Chi Chen, Zaid Harchaoui, and Jack Lee, whom I am very happy to have gotten to know over the last few years. I'd like to thank, chronologically, other valued teachers and mentors like Fritzie Reisner, Robert Knatt, Gary Pounder, Jennifer Judge, Jeff Nomura, Meei-Li Huang, Maksym Lipyanski, Michael Thaddeus, Harmen Bussemaker, Mikhail Khovanov, Cindy Dunbar, Joy Wu, Vlad Minin, Jason Xu, Raphael Gottardo, Daniela Witten, Johannes Lederer, Kari Torkkola, Jim Pfaendtner, and Stefan Mihalas. I'd like to thank my groupmates Hanyu Zhang, Yu-Chia Chen, Zhenman Yuan, and James Buenfil. And of course! my friends, in statistics and otherwise. Finally, I'd like to thank my family members, and especially my parents Susan McArdle and David Koelle; love.

DEDICATION

To the University of Washington. I am lucky that I can see the house I grew up in from Padelford, and vice-versa.

Chapter 1

INTRODUCTION

Machine learning algorithms derive utility from the quality of their learned representations. The ability of non-parametric algorithms to learn good representations of large datasets is constantly increasing. This progress powers a wide range of applications. However, the use of complicated black-box algorithms hinders the crucial notion of *interpretability*.

Interpretable models offer many advantages. They impart mechanistic understanding and are safer to use. For example, a scientist may wish to know if a circular manifold they observe in their cells' gene-expression corresponds to position in the cell cycle, or a physician using supervised learning to predict breast cancer from a mammogram may want to know which parts of the image are actually driving the automated diagnosis. Interpretable models can learn more efficiently and generalize better. From weather-prediction to robotics, non-parametric algorithms constrained to solve intelligently-specified tasks require less data to achieve comparable performance.

In all contexts, interpretability emerges through comparison of what is learnable by the model with what is already known to be important in the domain of the problem. A particular gene may govern the cell-cycle, a particular feature may drive a diagnosis, or a particular equation may characterize the observed dynamics. However, a unifying framework for the incorporation of intepretable information into non-parametric models has yet to emerge.

The lack of interpretability of learned representations contrasts with the estimates made using parametric statistical methods such as linear regression. Parametric statistical models give clarity about how individual covariate features drive response, and so the model is as interpretable as the covariates. The modeller has control over the functions in the covariate set, and so epistemic certainty is under her control. The is especially true when the model is constrained to be sparse. Unfortunately, there does not exist a comparable paradigm for interpreting non-linear representations.

This thesis therefore introduces a set of algorithmic, statistical, and mathematical tools for ascribing meaning to learned representations. In Chapters 3 and 4, we introduce a class of non-linear regression models that find approximations to learned representations from within sets of user-defined interpretable functions. Chapter 5 then gives some geometric implications of these algorithms. Much of the material from Chapter 3 and 5 is taken from (107). Finally, Chapter 6 gives tools for expanding the possible set of interpretable functions to observed covariates, and provides a new perspective on the usefulness of original learned manifold representation. A summary is given in Figure 1.1.

$$\begin{tikzcd}
\text{Data } \mathcal{D} \arrow[r, "\text{Manifold Learning}"] \arrow[d, "\text{Chapter 6}"'] \arrow[dr, "\text{Chapter 4/5}"] & \text{Embedding } \phi \\
\text{Covariate } f & \text{Dictionary } \mathcal{G} \arrow[u, "\text{Chapter 3/5}"']
\end{tikzcd}$$

Figure 1.1: The mathematical setting of the thesis. We give algorithms for establishing relations between several classes of functions. Data $\mathcal{D}$ is the original data observed in high-dimensional feature functions. An embedding ϕ consists of low-dimensional learned representation functions. $\mathcal{G}$ refers to a set of analytically available interpretable covariate functions which we call a dictionary. f refers to an interpretable covariates function available only through sampling at the data points.

1.1 Chapter 3 Manifold Coordinates with Physical Meaning

Understanding the structure of a dataset $\mathcal{D}$ observed in a high dimensional space is typically accomplished by applying an unsupervised manifold learning algorithm to learn an embedding representation Φ. However, the learned coordinates of the embedding map are abstract, and so interpretation is often left to a domain expert. Typically, this process relies on visualization inspection of the learned coordinates for the presence of covariates of interest. This chapter introduces a method - ManifoldLasso - that automates this inspection process. It takes as input $\mathcal{D}$, Φ, and covariates of interest $\mathcal{G}$, and learns map from $\mathcal{G}$ to Φ that is sparse with respect to the functions in $\mathcal{G}$. The selected functions in $\mathcal{G}$ form an interpretable representation of $\mathcal{D}$.

This automation procedure is a regularized regression method that linearizes the non-linear support recovery problem by considering it on the differential level. Our contributions are to i) formulate the interpretability problem as finding a set of functions in $\mathcal{G}$ that are a local diffeomorphism to the data manifold, ii) provide an algorithm based on group lasso for recovering this support, iii) provide an algorithm for estimation of gradients based on the differential geometric notion of pullback, iv) provide a method for enabling our overall differential algorithm in the rotationally and translationally invariant molecular shape space. We provide results on multiscale real data from molecular dynamics.

1.2 Chapter 4 Dictionary-based Manifold Learning

This chapter introduces a simplified version of ManifoldLasso called TSLasso, or Tangent Space Lasso. The Tangent Space Lasso does not interpret the coordinates of an existing embedding, but rather composes a local diffeomorphism directly from a dictionary of supplied functions by identifying a suitable subset whose gradients span the estimated data manifold tangent spaces. The contributions of this chapter are i) the extension of the regression technique from Chapter 3 to explaining subspaces rather than functions, and ii) application to interpretable manifold learning. We apply this algorithm on molecular dynamics data, and show that it generates functional support recovery comparable to ManifoldLasso without use of an embedding.

1.3 Chapter 5 Manifold and Tangent Space Basis Pursuit

The previous chapters propose a set of regression methods for identifying sparse parameterizations of manifolds. These approaches suffer in the overcomplete dictionary setting and when $\mathcal{G}$ is large. First, the local diffeomorphism condition may not be a uniquely satisfied success criteria. Second, the algorithm can fail to select a suitable parameterization when support recovery conditions are violated.

Therefore, in this chapter we propose a modified set of objectives that trade computational speed for robustness and specificity. These objectives are related to the convex dual of the objectives from Chapter 3 and 4 but distinct in several important ways. Our contributions are i) derivation of the Manifold and Tangent Space Basis Pursuit algorithms, ii) characterization of the solution of the Tangent Space Basis Pursuit solution as finding a most isometric embedding, and iii) introduction of a two-stage procedure that combines the regularized and basis pursuit objectives to increase computational efficiency. Results are again given on molecular dynamics data.

1.4 Chapter 6 Embedding-based Tangent Spaces for Gradient Estimation.

The paradigm for interpretable dimension reduction proposed in the previous chapters made several assumptions that may not always be satisfied. First, we have assumed that the ”interpretable” covariates of interest are analytically available with respect to the features. However, some covariates are separate pieces of information about the same samples, and so the gradient information necessary for our Manifold and Tangent Space Lasso methods is not immediately available. Second, we have assumed that the dataset is observed without noise and that the learned embedding map is a diffeomorphism into a lower dimensional space. This assumption does not reflect that manifold learning methods can also remove noise. This chapter examines these deficiencies in greater detail.

We study the estimation of the gradient of a function f available only via its values at the data points using local linear regression. At each data point, this gradient is an element of the corresponding data manifold tangent space. Our contributions are i) we examine estimation of this tangent space through the lens of manifold learning, ii) give a condition under which the gradients of a learned representation may be used to estimate the data manifold tangent space, iii) use a tangent space estimated in this manner in subsequent estimation of the

gradient of a black-box function available only at the sample points. We show results on simulated and astronomical data.

Chapter 2
BACKGROUND

The core contribution of this thesis is a set of algorithms for interpretable unsupervised learning. However, some of these algorithms may have independent interest. This chapter reviews our general problem set-up, and relevant topics in differential geometry, manifold learning, molecular dynamics, and statistics.

2.1 Problem formulations

We are given a dataset $\mathcal{D} \in \mathbb{R}^{n \times D}$ of points $\xi_i \in \mathbb{R}^D$ for $i \in [n]$. In Chapters 3, 4, and 5, we assume that this data is sampled from a smooth manifold $\mathcal{M}$ of intrinsic dimension d, where $d << D$. This assumption is relaxed in Chapter 6. In Chapters 3, 5 and 6, we assume access to a data embedding $\Phi \in \mathbb{R}^{n \times m}$ learned by a manifold learning algorithm acting on $\mathcal{D}$ that preserves geometric or topological properties of $\mathcal{M}$.

The task of Chapter 3 is to ascribe meaning to the learned representation Φ with respect to a dictionary of user-defined and domain-related smooth functions $\mathcal{G} = \{g^1, \dots g^p$, with $g^j : U \subseteq \mathbb{R}^D \to \mathbb{R}\}$, where U is an open set containing $\mathcal{M}$. We give a regression method based on the chain rule that sparsely selects a function basis among the elements of $\mathcal{G}$.

In Chapter 4, we show that a similar method can be used without an embedding Φ. That is, the method introduced in Chapter 3 itself can be used directly to find an interpretable embedding.

Chapter 5 expands the method of Chapters 3 and 4 to more robustly handle the setting where p is large. We provide a more specific manifold support recovery success criteria that we link to the mathematical notion of isometry.

In contrast to the dictionaries used in the previous chapters, the covariate functions

$f_{\epsilon,1:n} \in \mathbb{R}^n$ studied in Chapter 6 are available only through observation at the data points. Our task in this chapter is estimation the gradient of this function with respect to $\mathcal{M}$. We provide a two-stage method using the learned representation Φ for when data is sampled near $\mathcal{M}$ rather than from it.

2.2 Differential geometry

The mathematical field of differential geometry provides the language for describing many aspects of science, mathematics, and engineering, and therefore has attracted interest in the data analytics community. This section therefore describes relevant background in this area (1; 119). Where possible, we provide definitions in coordinates. For a complete treatment, see Lee (119).

2.2.1 Manifolds

Manifolds are mathematical formalizations of surfaces of arbitrary dimension.

Definition 1. *Let $\mathcal{M}$ be a topological space and $U \subseteq \mathcal{M}$ be an open set. Then a* ***coordinate system*** *ψ is a homeomorphism from U to V, an open subset of $\mathbb{R}^d$. The pair (U, ψ) is known as a* ***chart*** *and the inverse map ψ^{-1} is known as a* ***parameterization****.*

Definition 2. *An* ***atlas*** *is a set of charts $\{(U_\alpha, \psi_\alpha)\}$ such that $\mathcal{M} = \cup_{\alpha \in A} U_\alpha$.*

Definition 3. *A* ***topological manifold*** *$\mathcal{M}$ is a second countable Hausdorff topological space that admits an atlas $\{(U_\alpha, \psi_\alpha) : \alpha \in A\}$.*

Definition 4. *A* ***transition map*** *is a map*

$$\psi_\alpha \circ \psi_\beta^{-1} : \psi_\beta(U_\alpha \cap U_\beta) \to \psi_\alpha(U_\alpha \cap U_\beta). \tag{2.1}$$

Definition 5. *A C^k* ***smooth manifold*** *$\mathcal{M}$ is a topological manifold for which the transition maps have continuous partial derivatives of order k.*

We say two atlases are equivalent if their union is an atlas, and equivalence classes of atlases are called smooth structures. The smooth structure associated to a manifold is independent of choice of atlas.

2.2.2 Tangent spaces

Many of the techniques in this thesis use differential operators in the tangent spaces of various manifolds. While tangent spaces are defined intrinsically as sets of derivations of smooth functions, they have a geometric interpretation that more readily lends itself to computational analysis.

Definition 6. *A* ***derivation at*** ξ *is a linear map* $d : C^\infty(\mathcal{M}) \to \mathbb{R}$ *satisfying* $d(fg) = f(\xi)dg + g(\xi)df$.

Definition 7. *The* ***tangent space*** $\mathcal{T}_\xi\mathcal{M}$ *of a smooth manifold* $\mathcal{M}$ *at a point* $\xi \in \mathcal{M}$ *is the set of derivations at* ξ.

Theorem 8. *(118) The tangent space* $\mathcal{T}_\xi\mathcal{M}$ *of a* d *dimensional manifold* $\mathcal{M}$ *is isomorphic to* $\mathbb{R}^d$.

Thus, the abstract notion of tangent space manifests as its familiar geometric intuition.

2.2.3 Differentials

The differential plays a central role in our comparisons of the geometric properties of manifolds and of maps between them (118).

Definition 9. *The* ***differential*** *of a smooth map* $\phi : \mathcal{M} \to \mathcal{N}$ *between* $d_\mathcal{M}$ *and* $d_\mathcal{N}$ *dimensional manifolds at a point* $\xi \in \mathcal{M}$ *is a map*

$$D\phi(\xi) : \mathcal{T}_\xi\mathcal{M} \to \mathcal{T}_{\phi(\xi)}\mathcal{N} \tag{2.2}$$

which, in bases $x^1 \dots x^{d_\mathcal{M}}$ of $\mathcal{T}_\xi\mathcal{M}$ and $y^1 \dots y^{d_\mathcal{N}}$ of $\mathcal{T}_{\phi(\xi)}\mathcal{N}$ has entries

$$D\phi(\xi) = \begin{bmatrix} \frac{\partial \phi^1(\xi)}{\partial x^1}(\xi) & \dots & \frac{\partial \phi^1(\xi)}{\partial x^{d_\mathcal{M}}}(\xi) \\ \vdots & & \vdots \\ \frac{\partial \phi^{d_\mathcal{N}}(\xi)}{\partial x^1}(\xi) & \dots & \frac{\partial \phi^{d_\mathcal{N}}(\xi)}{\partial x^{d_\mathcal{M}}}(\xi) \end{bmatrix}. \tag{2.3}$$

The rank of the differential characterizes two types of maps.

Definition 10. *A smooth map $\phi : \mathcal{M} \to \mathcal{N}$ from a $d_\mathcal{M}$ to a $d_\mathcal{N}$ dimensional manifold is a* ***submersion*** *if rank$(D\phi(\xi)) = d_\mathcal{N}$, an* ***immersion*** *if rank$(D\phi(\xi)) = d_\mathcal{M}$ for all $\xi \in \mathcal{M}$.*

That is, immersions preserve local structure while submersions lower dimensionality.

Definition 11. *A smooth map $\phi : \mathcal{M} \to \mathcal{N}$ is a* ***diffeomorphism*** *if it is a bijection with smooth inverse.*

Theorem 12. *(119) A smooth map $\phi : \mathcal{M} \to \mathcal{N}$ is a diffeomorphism if it is a bijection and $D\phi$ has constant rank.*

Definition 13. *When $\phi : \mathcal{M} \to \mathcal{N}$ is a immersion, $D\phi$ is known as the* ***pushforward****. When ϕ is a diffeomorphism, $D\phi^{-1}$ is known as the* ***pullback****.*

Definition 14. *A smooth map $\phi : \mathcal{M} \to \mathcal{N}$ is a* ***local diffeomorphism*** *at a point $\xi \in \mathcal{M}$ if it is a bijection with smooth inverse in a open neighborhood $U \subset \mathcal{M}$ containing ξ.*

2.2.4 Submanifolds

Often, data lies in a subspace of some higher-dimensional space. Therefore, we make there following definitions.

Definition 15. *A $d_\mathcal{M}$ dimensional* ***submanifold*** *$\mathcal{M}$ of a $d_\mathcal{E}$ dimensional manifold $\mathcal{E}$ is a subset $\mathcal{M} \subseteq \mathcal{E}$ such that the inclusion map $i : \mathcal{M} \to \mathcal{M} \subseteq \mathcal{E}$ is a smooth embedding.*

Definition 16. *A smooth map $\phi : \mathcal{M} \to \mathcal{N} \subseteq \mathcal{E}$ from a $d_\mathcal{M}$ to a $d_\mathcal{N}$ is an* ***embedding*** *if it is an immersion and $\mathcal{N}$ is a manifold in the subspace topology inherited from $\mathcal{E}$.*

The condition that the $\mathcal{N}$ only can be an embedding of $\mathcal{M}$ if it is an manifold in the subspace topology prevents self-crossings and other singularities. The following theorem justifies acquiring a low-dimensional representation of data observed in a high-dimensional space.

Theorem 17. ***Whitney Embedding*** *(119) Given a d dimensional smooth manifold $\mathcal{M}$, there exists an embedding of $\mathcal{M}$ into $\mathbb{R}^m$ with $d \leq m \leq 2d$.*

The notion of fibration is used in several places throughout this thesis. We use these spaces to model the removal of noise by an embedding algorithm in Chapter 6, but they are also used implicitly in our analysis of the molecular shape space in Chapters 3, 4, and 5.

Definition 18. *A* ***fibered smooth manifold*** *is a triple $(\mathcal{E}, \mathcal{M}, \phi)$ of a $d_{\mathcal{E}}$ dimensional manifold $\mathcal{E}$, a $d_{\mathcal{M}}$ dimensional manifold $\mathcal{M}$, and a mapping $\phi : \mathcal{E} \to \mathcal{M}$ that is a surjective submersion.*

Definition 19. *At a point $\xi \in \mathcal{M}$, define the* ***normal space*** *of $\mathcal{T}_\xi \mathcal{M}$ within $T_\xi \mathcal{E}$ to be $N_\xi(\mathcal{M}, \mathcal{E}) := T_\xi \mathcal{E} / \mathcal{T}_\xi \mathcal{M}$.*

Finally, note that we can represent tangent spaces of submanifolds using linear algebra. A tangent space $\mathcal{T}_\xi \mathcal{M}$ of a d dimensional submanifold $\mathcal{M}$ of $\mathbb{R}^D$ at a point $\xi \in \mathcal{M}$ is a d dimensional linear subspace of $\mathbb{R}^D$, and therefore an element of the Grassmannian $G(d, D)$. An orthonormal basis $T_\xi^{\mathcal{M}}$ for this this space is an element of the Stiefel manifold $St(d, D)$.

2.2.5 Metric geometry

Riemannian geometry allows introduction of angles and distances on manifolds.

Definition 20. *Given a d dimensional smooth manifold $\mathcal{M}$, a* ***Riemannian metric*** $\mathbf{g}$ *is a symmetric positive definitive tensor field that defines an inner product $\langle u, v \rangle_{\mathbf{g}(\xi)}$ on every $\mathcal{T}_\xi \mathcal{M} \in \mathcal{M}$.*

That is, $g(\xi) : \mathcal{T}_\xi\mathcal{M} \times \mathcal{T}_\xi\mathcal{M} \to \mathbb{R}_{\geq 0}$ that is positive definite, linear, and symmetric for all $\xi \in \mathcal{M}$. Given a point $\xi \in \mathcal{M}$, $\mathbf{g}(\xi)$ can be represented as a symmetric positive definite d by d matrix G_ξ that depends on choice of tangent basis for $\mathcal{T}_\xi\mathcal{M}$. When $\langle u, v\rangle_{\mathbf{g}(\xi)} = \langle u, v\rangle$ for all $u, v \in \mathcal{T}_\xi\mathcal{M}$, we call it inherited from the ambient space, and denote it as **id**.

Definition 21. *A **Riemannian manifold** $(\mathcal{M}, \mathbf{g})$ is a pair of a d dimensional smooth manifold $\mathcal{M}$ and a Riemannian metric $\mathbf{g}$ on $\mathcal{M}$.*

Using the Riemannian metric, we can state the condition for geometric equivalence of two manifolds.

Definition 22. *A diffeomorphism ϕ between two Riemannian manifolds $(M, \mathbf{g})$ and $(N, \mathbf{h})$ is an **isometry** if, for all points $\xi \in \mathcal{M}$*

$$\langle u, v\rangle_{\mathbf{g}(\xi)} = \langle D\phi u, D\phi v\rangle_{\mathbf{h}(\phi(\xi))}$$

for all $u, v \in \mathcal{T}_\xi\mathcal{M}$.

Proposition 23. *Given an isometry ϕ, if $\mathbf{g} = \mathbf{id}$ and $\mathbf{h} = \mathbf{id}$, then $D\phi(\xi)$ is a unitary $d \times d$ matrix at all points $\xi \in \mathcal{M}$.*

Proof. Since ϕ is an isometry, $u^T v = u^T D\phi(\xi)^T D\phi(\xi) v$ for all $u, v \in \mathcal{T}_\xi\mathcal{M}$. Therefore, $D\phi(\xi)^T D\phi(\xi) = I_d$, and so $D\phi(\xi)$ is unitary. □

Other mappings like similarity transformations, conformal maps, and orthogonal coordinates may also be characterized by the linear algebraic properties of $D\phi$.

Many statistical algorithms seek to identify low-dimensional isometric embeddings. The following theorem shows that this is possible.

Theorem 24. ***Nash Embedding*** *A d dimensional smooth manifold $\mathcal{M}$ can be isometrically embedded in $\mathbb{R}^s$ where $s = \max(\frac{d(d+1)}{2} + 2d, \frac{d(d+1)}{2} + d + 5)$.*

Unfortunately, the proof technique of this theorem is non-constructive, and does not give an algorithm for finding the desired mapping (54). In contrast, the locally isometric embedding of d dimensional manifolds with non-zero curvature into d dimensions is not possible (108). The challenge of finding isometric embeddings has lead to study of the metric induced by ϕ itself (151).

Definition 25. *For any diffeomorphism $\phi : \mathcal{M} \to \mathcal{N}$, exists a unique Riemiannian metric called the* ***pushforward metric*** $\mathbf{h}$ *such that (129)*

$$\langle u, v \rangle_{\mathbf{g}(\xi)} = \langle D\phi(\xi)u, D\phi(\xi)v \rangle_{\mathbf{h}(\phi(\xi))}. \tag{2.4}$$

Lemma 26. *For any diffeomorphism $\phi : \mathcal{M} \to \mathcal{N}$, when $\mathbf{g} = \mathbf{id}$, the pushforward metric $\mathbf{h}(\phi(\xi)) = D\phi^{-1}(\phi(\xi))^T D\phi^{-1}(\phi(\xi))$.*

Proof. $\langle u, v \rangle_{\mathbf{g}(\xi)} = u^T v$. On the other hand,

$$\langle D\phi u, D\phi v \rangle_{\mathbf{h}(\phi(\xi))} = u^T D\phi(\xi)^T H_{\phi(\xi)} D\phi(\xi) v. \tag{2.5}$$

Since ϕ is a diffeomorphism, $D\phi(\xi)$ is full rank, and so is invertible with $(D\phi^{-1}(\phi(\xi)))^T = (D\phi^T(\phi(\xi)))^{-1}$ with respect to orthogonal bases of $T_\xi\mathcal{M}$ and $T_{\phi(\xi)}\phi(\mathcal{M})$. Thus, $H_(\phi(\xi)) = D\phi^{-1}(\phi(\xi))^T D\phi^{-1}(\phi(\xi))$. □

As shown in (151), this also extends to tangent coordinates of submanifolds in the expected way using pseudoinverses. Their technique for estimating this metric uses the manifold Laplacian, which encodes geometric information about the manifold $\mathcal{M}$.

Definition 27. *The manifold Laplacian, also known as the* ***Laplace-Beltrami operator*** *is a differential operator on functions $f : \mathcal{M} \to \mathbb{R}$. In coordinates, it is given by*

$$\Delta_M : C^k(M) \to C^{k-2}(M) \tag{2.6}$$

$$f(\xi) \mapsto \frac{1}{det(\mathbf{g}(\xi))} \sum_{i=1}^{d} \sum_{j=1}^{d} \frac{1}{\partial x^i} \sqrt{det(\mathbf{g}(\xi))} \mathbf{g}_{ij}(\xi) \frac{\partial f}{\partial x^j} \tag{2.7}$$

In tangent coordinates when $\mathbf{g} = \mathbf{id}$

$$\Delta_M(f) = \sum_{k=1}^{d} \frac{\partial^2 f}{\partial x^2}. \tag{2.8}$$

In other words, $\Delta_{\mathcal{M}} f = \text{div grad } f$, where grad represents the gradient of f and div the divergence vector field operator.

Definition 28. *The* ***gradient*** *of a smooth function* $f : \mathcal{M} \to \mathbb{R}$ *at* $\xi \in \mathcal{M}$ *is the unique vector* $\operatorname{grad}_{\mathcal{M}} f(\xi) \in \mathcal{T}_\xi \mathcal{M}$ *satisfying*

$$\langle \operatorname{grad}_{\mathcal{M}} f(\xi), v \rangle_{\mathbf{g}(\xi)} = Df(\xi) \textit{ for any } u \in \mathcal{T}_\xi \mathcal{M}. \tag{2.9}$$

Here, Df is the differential of f consisting of partial derivatives, as in Equation 2.3. Thus, in coordinates, $\operatorname{grad}_{T_\xi^{\mathcal{M}}, \mathbf{id}} = [\frac{\partial f}{\partial x^1}(\xi) \dots \frac{\partial f}{\partial x^d}(\xi)]^T$, and, contravariantly, $\operatorname{grad}_{T_\xi^{\mathcal{M}}, \mathbf{g}} = G_\xi^{-1} \operatorname{grad}_{T_\xi^{\mathcal{M}}, \mathbf{id}}$. We generically refer to $\operatorname{grad}_{T_\xi^{\mathcal{M}}, \mathbf{id}}$ as $\operatorname{grad}_{T_\xi^{\mathcal{M}}, \mathbf{id}}$. Similarly, we can define the gradient of a function f w.r.t. $\mathcal{M}$ as a submanifold of $\mathbb{R}^D$. When $\mathcal{M} \subset \mathbb{R}^D$, and f is a smooth function of $\mathbb{R}^D$, then $\operatorname{grad}_{\mathcal{M}} f(\xi)$ satisfies $\operatorname{grad}_{\mathcal{M}} f(\xi) = \operatorname{Proj}_{\mathcal{T}_\xi \mathcal{M}}(\nabla_\xi f(\xi))$ where $\operatorname{Proj}_{\mathcal{T}_\xi \mathcal{M}}$ denotes the Euclidean projection onto the subspace $\mathcal{T}_\xi \mathcal{M}$, and $\nabla_\xi f(\xi) = [\frac{\partial f}{\partial x^1}(\xi), \dots \frac{\partial f}{\partial x^D}(\xi)]^T$ is the vector of partial derivatives of f w.r.t. the Euclidean coordinates of $\mathbb{R}^D$. Then $\operatorname{grad}_{T_\xi^{\mathcal{M}}} f = T_\xi^{\mathcal{M}^T} \nabla_\xi f$.

2.3 Manifold learning

Manifold learning (ML) is a collection of approaches for data analysis based on the principle that the dimensionality of many high-dimensional data sets is often only artificially high. That is, we assume data is sampled i.i.d. from a smooth manifold $\mathcal{M}$ of dimension d embedded within a high-dimensional feature space $\mathbb{R}^D$. ML includes methods from unsupervised dimension reduction, to supervised representation learning, to in situ denoising algorithms, and is motivated by diverse tasks like compression, visualization, and noise reduction. This section reviews several relevant algorithms with focus on statistical estimators of the geometric objects introduced in the previous sections.

2.3.1 Unsupervised learning

Algorithms like Isomap, Local Tangent Space Alignment, Autoencoders, Diffusion Maps, t-SNE, and UMAP are used to create low-dimensional representation of high-dimensional data in applications from genomics to chemistry. Ideally, these methods preserve topoligical and geometric features of the data manifold as it lies in the high-dimensional feature space. For example, if our data lie on a sphere in the original space, then we desire that they should also lie on a sphere in our learned lower-dimensional space, with distances between observations in the low-dimensional space corresponding to distances between observations in the high-dimensional space. Unfortunately, popular algorithms like t-SNE and UMAP have not been shown to converge to some geometric object from the manifold sample space. Without the statistical quality of consistency, the user of a dimension reduction algorithm is left to guess about its accuracy. This has lead to controversy (40) and the development of data-driven evaluation techniques (153). In contrast, the convergence of the sample graph Laplacian to the manifold Laplace-Beltrami operator provides a principled estimation strategy for uncovering geometric and topological information about large complex datasets (21). This directly leads to consistency results for the derivative Diffusion Maps and Laplacian Eigenmaps learning algorithms which we will use throughout the thesis.

2.3.2 The neighborhood graph

The **neighborhood graph** $\mathbb{G} = (V, E)$ consists of vertices representing data points ξ_i and edges denoting adjacency of a data point ξ_i to its set of **neighbors** $\mathcal{N}_i = \{i' \in [n], \text{with } \|\xi_{i'} - \xi_i\| \leq r\}$, where r is a **neighborhood radius** parameter. The neighborhood relation is symmetric, and so determines an undirected graph with nodes represented by the data points. This graph is an unsurprising starting place for manifold learning, since the empirical neighborhoods encode topological information about the data manifold. Many unsupervised learning methods including Diffusion Maps, Isomap, Local Tangent Space Alignment, t-SNE, and UMAP all begin with construction of $\mathbb{G}$.

2.3.3 The kernel matrix

The Gaussian **kernel matrix** $K \in \mathbb{R}^{n \times n}$ whose elements are

$$K_{i,i'} = \begin{cases} \exp\left(-\frac{\|\xi_i - \xi_{i'}\|}{\epsilon_N^2}\right) & \text{if } i' \in \mathcal{N}_i \\ 0 \text{ otherwise.} \end{cases} \tag{2.10}$$

encodes adjacency information in a more detailed way using real values weights. Typically, the radius r and the **bandwidth** parameter ϵ are related by $r = c\epsilon$ with c a small constant greater than 1. In this thesis we set $c = 3$. This ensures that K is close to its limit when $r \to \infty$ while remaining sparse, with sparsity structure induced by the neighborhood graph. We denote rows of this matrix as $K_{i,\mathcal{N}_i}$ to emphasize that only k_i values of each row i are needed. The kernel matrix will be used in both our embedding and tangent space estimation steps.

2.3.4 The renormalized graph Laplacian

We estimate the manifold Laplace-Beltrami operator $\Delta_{\mathcal{M}}$ by the **renormalized graph Laplacian**, also known as the **sample Laplacian**, or **Diffusion Maps Laplacian** L (49). As for K, construction of this neighborhood graph is the computationally expensive component of this algorithm. Elements and rows of L will be denoted by $L_{i,i'}$ and $L_{i,\mathcal{N}_i}$, respectively, since the sparsity pattern of L is given by the neighborhood graph.

Algorithm 1 LAPLACIAN(neighborhoods $\mathcal{N}_{1:n}$, local data $\Xi_{1:n}$, bandwidth ϵ)

```
Compute kernel matrix $K$ using (2.10)
Compute normalization weights $w_i \leftarrow \sum_{i' \in \mathcal{N}_i} K_{i,i'}$, $i = 1, \dots n$, $W \leftarrow \mathrm{diag}(w_i,\ i = 1:n)$
Normalize $\tilde{L} \leftarrow W^{-1} K W^{-1}$
Compute renormalization weights $\tilde{w}_i \leftarrow \sum_{i' \in \mathcal{N}_i} \tilde{L}_{i,i'}$, $i = 1, \dots n$, $\tilde{W} = \mathrm{diag}(\tilde{w}_i,\ i = 1:n)$
Renormalize $L \leftarrow \frac{4}{\epsilon^2}(\tilde{W}^{-1}\tilde{L} - I_n)$
**Output** Kernel matrix $K$, Laplacian $L$
```

The proof of this consistency of this estimator, constructed by the LAPLACIAN algorithm, is an important result in manifold learning (49). It is unbiased w.r.t. the sampling density on $\mathcal{M}$ (84; 85; 181). L is a sparse matrix; its i-th row contains non-zero values only for $i' \in \mathcal{N}_i$. In summary, we compute $L = I_n - K\tilde{W}^{-1}W^{-1}KW^{-1}$ where $W = \mathrm{diag}(K\mathbf{1_n})$ and $\tilde{W} = \mathrm{diag}(W^{-1}KW^{-1}\mathbf{1_n})$, where $\mathbf{1_n}$ is a vector of 1 of length n.

2.3.5 Diffusion Maps

Although any algorithm which asymptotically generates a smooth embedding would be acceptable for most of the methods in this thesis, we generally use the **Diffusion Map** (49) or the **Laplacian Eigenmap** (19) embedding of $\mathcal{D}$. The Diffusion Maps embedding coordinates ϕ^k are the m principal eigenvectors of L. This embedding has well-known statistical properties. As L converges to Δ_M, the m principal eigenvectors of L converge to the m principal eigenfunctions of $\Delta_{\mathcal{M}}$ (191).

2.3.6 Estimating tangent spaces

We use the **Weighted Local Principal Component Analysis** algorithm described in WLPCA to estimate tangent spaces of $\mathcal{M}$ at data points ξ_i (41). For this algorithm and others we define the SVD algorithm $\textsc{svd}(X, d)$ of a symmetric matrix $X \in \mathbb{R}^{D \times D}$ as outputting V, Λ, where $\Lambda \in \mathbb{R}^d$ and $V \in \mathbb{R}^{D \times d}$ are the largest d eigenvalues and their eigenvectors, respectively. Again, denote a column vector of ones of length k by $\mathbf{1}_k$.

Algorithm 2 WLPCA(local data Ξ_i, kernel row $K_{i,\mathcal{N}_i}$, intrinsic dimension d)

```
Compute normalization weights $w_i \leftarrow \sum_{i' \in \mathcal{N}_i} K_{i,i'}$
Compute weighted mean $\bar{\xi}_i \leftarrow \frac{1}{w_i} K_{i,\mathcal{N}_i} \Xi_i$
Compute weighted local differences
    $Z_i \leftarrow \mathrm{diag}(K_{i,\mathcal{N}_i}^{1/2})(\Xi_i - \mathbf{1}_{k_i} \bar{\xi}_i)$
Compute $T_i, \Lambda \leftarrow \mathrm{SVD}(Z_i^T Z_i, d)$
Output $T_i$
```

Estimation of tangent spaces in the presence of noise is an active area of research (6; 155).

2.3.7 *Isometric embeddings of data*

Isometric embeddings preserve information like angles and lengths of curves between points. This is important both for visualization, as well as for statistical tasks like nearest neighbor regression. This has motivated a search for algorithms that perform isometric embeddings. This can be induced either through preservation of distances between points (123; 147) or explicit control of the metric properties of the learned embedding map (156; 165; 109; 129). However, comprehensively useful algorithm for learning as isometric as possible embeddings has not emerged (185).

An alternative approach is to explicitly estimate the pushforward metric of the coordinates output by learned embedding map that takes data points ξ_i to points $\phi(\xi_i) \in \phi(\mathcal{M}) \subset \mathbb{R}^m$. The RMetric algorithm (152) for estimating this metric acts on local position matrices $\Phi_i := [\phi(\xi_{i'}) : i' \in \mathcal{N}_i] \in \mathbb{R}^{k_i \times m}$ of a neighborhood in $\mathbb{R}^m$. Let $\mathbf{g}$ now be the pushforward metric induced by a learned embedding ϕ, and denote by $\mathbf{g}^\dagger$ the inverse metric on the cotangent space of $\phi(\mathcal{M})$, which, in coordinates at a point $\phi(\xi)$, is given by $G_{\phi(\xi)}^\dagger = (G_{\phi(\xi)})^\dagger$, the pseudoinverse of $G_{\phi(\xi)}$. We can estimate G_i using the following estimator of the inverse

Riemmanian metric $G^{\dagger}_i$ at a data point ξ_i:

$$\widehat{G^{\dagger}_{kk'}(\xi_i)} = \frac{1}{2}\sum_{i\in\mathcal{N}_i} L_{ii'}(\phi^k(\xi_{i'}) - \phi^k(\xi))(\phi^{k'}(\xi_{i'}) - \phi^{k'}(\xi)) \tag{2.11}$$

where ϕ^k and $\phi^{k'}$ are coordinate functions of $\mathbb{R}^m$. That is, the L may be used to identify the pushforward metric in any coordinate system.

Algorithm 3 RMetric(Laplacian row $L_{i,\mathcal{N}_i}$, local embedding coordinates Φ_i, intrinsic dimension d)

```
Compute centered local embedding coordinates
    $\tilde{\Phi}_i = \Phi_i - \mathbf{1}_{k_i}\phi(\xi_i)^T$
Form matrix $G^{\dagger}_i$ by
    $G^{\dagger}_i \leftarrow [G^{\dagger}_{i,k,k'}]_{k,k'\in 1:m}$ with $G^{\dagger}_{i,k,k'} = \sum_{i'\in\mathcal{N}_i} L_{i,i'}\tilde{\Phi}_{i,i',k}\tilde{\Phi}_{i,i',k'}$ for $k, k' = 1 : m$.
Compute $V_i, \Lambda_i \leftarrow \mathrm{SVD}(G^{\dagger}_i, d)$
$G_i \leftarrow V_i\Lambda_i^{-1}V_i^T$.
Output $G_i$, optionally $V_i, \Lambda_i$
```

2.3.8 Tuning

The main hyperparameters used in this thesis are the manifold dimension d and the smoothing bandwidth ϵ. The methods we introduce are somewhat independent to questions of parameter tuning, and so we will assume that these parameters are known. However, we briefly review several methods to estimate d and ϵ.

Bandwidth estimation A common task in fitting manifold learning algorithms is determination of the neighborhood radius r and kernel bandwidth ϵ (104). Many approaches exist. We use a method that minimizes distortion measured using the pushforward metric (98).

Dimension estimation There are several commonly used types of manifold dimension estimation methods. One set of methods is based on the observation that, given a uniform density, the number of neighbors within a given radius of a point corresponds to the manifold dimension (105; 120). A second is based off of spectral thresholding of local principal components (145). In practice, while these estimators behave well under low or no noise, they are challenged by even limited amounts of noise. See Erba et al. (63) for a modern review.

2.4 Statistics background

The statistical setting of this thesis integrates two types of literature. On the one hand, this thesis uses tools like ordinary least squares and group lasso in new manifold learning contexts. On the other, manifold learning methods themselves have statistical properties. This thesis does not fully settle the convergence properties of the algorithms presented herein. However, the properties of these estimators give perspective on our contributions.

2.4.1 Linear regression

We use linear regression in several ways. Chapters 3, 4, and 5 propose regression methods for learning a parameterization of a manifold from a dictionary based off of linear regression in tangent bundles, Chapters 3 and 6 tackle gradient estimation using local linear regression, and Chapter 6 performs tangent space estimation via a local linear regression method.

Given data $\xi_{1:n} \in \mathbb{R}^{n \times D}$ and $f_{1:n} \in \mathbb{R}^n$, the ordinary least squares algorithm solves

$$\widehat{\beta} = \textsc{ols}(X, Y) := \arg\min_{\beta \in \mathbb{R}^D} \|f_{1:n} - \xi_{1:n}\beta\|_2^2. \tag{2.12}$$

This has the solution $\textsc{ols}(\xi_{1:n}, f_{1:n}) = (\xi_{1:n}^T \xi_{1:n})^{-1} \xi_{1:n}^T f_{1:n}$. Note that $(\xi_{1:n}^T \xi_{1:n})^{-1} \xi_{1:n}^T = \xi_{1:n}^\dagger$, the pseudoinverse of $\xi_{1:n}$.

In order for $\textsc{ols}$ to be well-defined, $\xi_{1:n}$ must be full rank. This can fail to happen either because $n < D$, or because ξ_i is sampled from some rank-deficient space. The latter setting is especially relevant, as tangent spaces of submanifolds are by definition rank-deficient.To address this problem, many popular regression softwares replace the inverse $(\xi_{1:n}^T \xi_{1:n})^{-1}$ with a pseudoinverse thresholded at small condition number $\kappa = \frac{\lambda_{\max}(\xi_{1:n}^T \xi_{1:n})}{\lambda_{\min}(\xi_{1:n}^T \xi_{1:n})}$. That is, small eigenvalues of $\xi_{1:n}^T \xi_{1:n}$ are set to 0 and not inverted. This algorithmic substitution is standard in many widely used software packages (78; 150), and is comparable to principal components regression (97) in that data is projected onto a principal subspace prior to OLS.

2.4.2 Error-in-variables

Error-in-variables refers to a collection of statistical problems all variables are observed with noise. This contrasts with the classical statistical regression paradigm in which only the response variable is observed with noise. In Chapter 6 we describe a new method for local linear regression in the presence of noise, and connect it to the error-in-variables manifold estimation problem (155).

Ordinary least squares with error-in-variables exhibits several elementary challenges. Given a linear model $f_\epsilon = \xi\beta + \epsilon$ with $\mathbb{E}(\epsilon) = 0$ and $\text{Var}(\xi)$ is full rank where ξ is observed as $\xi_\epsilon = \xi + \epsilon_\xi$ with $E(\epsilon_\xi) = 0$, the ordinary least squares estimator has expectation

$$E(\textsc{ols}(\xi_\epsilon, f_\epsilon)) = (\text{Var}(\xi) + 2\text{Cov}(\xi, \epsilon_\xi) + \text{Var}(\epsilon_\xi))^{-1}(\text{Var}(\xi)\beta + \text{Cov}(\xi, \epsilon_\xi)\beta + \text{Cov}(\epsilon, \xi) + \text{Cov}(\epsilon, \epsilon_\xi)). \tag{2.13}$$

A proof is found in (154; 7). Even when $\text{Cov}(\epsilon_x i, \xi_\epsilon) = 0$, this estimator has **attenuation bias**, and non-zero covariances cause additional bias. These problems have motivated application of alternative linear regression estimators such as total least squares (56).

2.4.3 Sparse linear regression

In Chapters 3 and 4 we introduce regularized group lasso regression for identifying sparse parameterizations of manifolds, and in Chapter 5 we discuss the theoretical setting of these methods in greater detail. Although our application of penalized regression is methodologically novel, it intimately relates to existing literature.

Penalties on learned coefficients can encourage selection of sparse solutions (4). Two examples of suitable optimization problems for sparse linear regression are

$$\arg \min_{\beta \in \mathbb{R}^D} \|\beta\|_0 \text{ s.t. } f_{1:n} = \xi_{1:n}\beta \tag{2.14}$$

$$\arg \min_{\beta \in \mathbb{R}^D} \|\beta\|_1 \text{ s.t. } f_{1:n} = \xi_{1:n}\beta. \tag{2.15}$$

Program (2.15) is known as basis pursuit.

The $l0$ regularized Program 2.14, although sparse, is computationally challenging to solve. In contrast, the seemingly combinatorial Program 2.15 is dual to the convex regularized **lasso** regression program

$$\arg \min_{\beta \in \mathbb{R}^D} \|f_{1:n} - \xi_{1:n}\beta\|_2^2 + \lambda\|\beta\|_1. \tag{2.16}$$

Here, dual means that, given a $\lambda > 0$, there exists a $C(\lambda) > 0$ such that the solution to Program (2.16) gives the same minimizer as

$$\arg \min_{\beta \in \mathbb{R}^D} \|\beta\|_1 \text{ s.t. } \|f_{1:n} - \xi_{1:n}\beta\|_2^2 < C(\lambda), \tag{2.17}$$

and that $C(\lambda)$ and λ are inversely monotonically related on values from $(0, \infty)$. Therefore, solving Program (2.16) at small λ approximates a solution to Program 2.15. All convex regularized regression methods admit a form of this duality. The conditions under which Program 2.15 recovers the same sparsity as Program 2.14 are well-studied (37; 134; 81).

In this thesis we use **group lasso**, a variant of the lasso that promotes joint sparsity within groups but not between them (197). In the group lasso, the user defines a set of p groups containing disjoint elements of the D features. An $\ell 2$ penalty is applied to coefficients within each group, while an $\ell 1$ penalty is applied across groups. Thus, as λ is increased, entire groups of coefficients enter the regularization path simultaneously. That is, given p disjoint subsets $S_j \subset [D]$,

$$\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^D} \|f_{1:n} - \xi_{1:n}\beta\|_2^2 + \sum_{j=1}^{p} \|\beta_{S_j}\|_2, \tag{2.18}$$

where β_{S_j} are the coefficients of the β vector belonging to the j-th group.

2.4.4 Convergence of Laplacian estimation

Considerable literature exists on the spectral and pointwise constency of the sample Laplacian L to the manifold Laplace-Beltrami operator $\Delta_{\mathcal{M}}$ (182; 24; 21; 49). However, the convergence rate is substantially less clear. As stated in Singer (174),

$$\frac{1}{\epsilon}\lim_{n\to\infty} Lf_{1:N} = \Delta_{\mathcal{M}} f + O(\epsilon^{1/2}) \tag{2.19}$$

for samples $1:n$ from an arbitrary function f on $\mathcal{M}$, and L estimated using $\xi_{1:n}$ and ϵ. This has motivated asymptotically shrinking the bandwidth ϵ at rates

$$\text{Uniform density } \epsilon = O((\frac{\log n}{n})^{\frac{1}{d+4}}) \text{ Hein et al. (83)} \tag{2.20}$$

$$\text{Non-uniform density } \epsilon = O((\frac{\log n}{\sqrt{n}})^{\frac{1}{2d+5}}) \text{ Shi (170)} \tag{2.21}$$

However, the relation of these convergence rates with noise in ξ is unclear. Coifman and Lafon (49) suggests that ϵ should remain larger than the magnitude of the noise as a useful heuristic. Although the role of noise has been explored for a variety of alternative manifold estimators (137; 75; 66), we are not aware of any result that L is a consistent estimator of $\Delta_{\mathcal{M}}$ in the presence of noise.

2.4.5 Tangent space estimation

The convergence of local PCA algorithms for tangent space estimation has similar challenges to that of the sample Laplacian in the presence of noise. The asymptotic rate of convergence of the local PCA estimator is given Aamari and Levrard (6) in the noiseless case with respect to the principal angle $\angle$ of the estimate $\widehat{\mathcal{T}_\xi\mathcal{M}}$ and true tangent space $\mathcal{T}_\xi\mathcal{M}$.

$$\angle(\widehat{\mathcal{T}_\xi\mathcal{M}}, \mathcal{T}_\xi\mathcal{M}) = O((\frac{1}{n})^{\frac{k-1}{d}}) \tag{2.22}$$

where $\mathcal{M}$ is $\mathcal{C}^k$-smooth. Unfortunately, this rate does not hold in the presence of non-asymptotically vanishing noise, and the consistency of this estimator is an open question (6; 155).

2.5 Molecular dynamics simulations

A motivating application of this thesis is the study of molecular dynamics simulations (MDS) using unsupervised learning (205; 160; 183). MDS is one of the principal tools in the study of molecular systems. See Friesner (71) for a review of methods in this field. Such simulations provide detailed information on the fluctuations and conformational changes of the simulated system, and are routinely used to investigate the structure, dynamics and thermodynamics of biological macromolecules and their complexes.

A **molecular dynamics simulation** generates samples $\xi_{1:T}$ from T states of the configuration space of the system. The distribution samples reflects the dynamics of the modelled physical process as it moves through time. The accuracy of molecular simulation varies for representing the "true" underlying physical phenomenon varies depending on the complexity of the phenonenom, as well as the type of simulation. One bifurcation is the choice of simulating quantum Schrodinger versus classical Langevin dynamics, which historically have been limited by their inability to predict quantum effects such as formation of bonds. At a middle level of complexity, density functional theory represents the molecular system by its Born-Oppenheimer approximation as atomic point masses, but with quantum dynamics (99). This contrasts with the most precise but computationally expensive available approximation, coupled cluster methods, in which electron correlations are explicitly modelled (200). The data used in this thesis from Chmiela et al. (48) is from a solver that uses machine learning to achieve coupled cluster accuracy with force fields. This is one of many recent methods that use machine learning to increase simulation speed and accuracy (175).

Even though the vector of atomic coordinates can take any value, due to interatomic interactions, the relative positions of atoms within the molecule lie near a low-dimensional manifold. Performing manifold learning on these data separates the conformational changes, modeled by the manifold, from the fluctuations represented by the "noise" around the manifold. In other words, it disentagles the **slow modes** of dynamical systems from the faster modes (205; 160; 183).

Inspection of eigenfunctions of stochastic systems for physically-relevant observables has a rich history (86; 198; 176). Depending on the spectral decomposition of the dynamical differential operator, it may have a central slow manifold on which large scale dynamics evolve accross relatively large lengths of time. See (38) for a more detailed accounting. Dynamical systems can have center manifolds corresponding to the eigenfunctions of their corresponding differential operators (136). However, in molecular dynamics, the Diffusion Maps embedding and other non-linear dimension reduction methods are somewhat colloquially said to identify slow modes, which refers to a related but non-rigorous notion of slowly evolving state. Such modes are important for interrogation of these other items in infrequent metadynamics, in which simulations are biased in the direction of collective variables (68; 205; 167). This can occur along axes by biasing simulations along physical axes visually identified as parameterizing the learned embedding, or directly moving in the direction of a differentiable embedding (28; 112).

2.5.1 Shape space

Symmetry functions are a way of featurizing molecular geometries consisting of particle information such as atomic positions so that the represented geometry is invariant to translation and rotation (8; 48; 44). This is particularly useful when simplifying molecular structure to consist of only atomic positions (**?** 64; 72). This invariance corresponds to the statistical notion of shape - what remains of position when translation and rotation, and dilation are ignored (103; 29; 101; 102; 115).

Definition 29. *Denote the* ***shape space*** *of k points in $\mathbb{R}^m$ as $\Sigma_m^k := (\mathbb{R}^m)^k / (E(m) \times R^+)$, where $E(m)$ is the Euclidean group of translations and rotations in a m dimensional space, and $\mathbb{R}^+$ is a dilation factor relative to the mean position of the N_a atoms.*

The shape space is a Riemannian manifold of dimension $mk - (2m + 1)$. Thus, in a molecular dynamics simulation of N_a atoms in $\mathbb{R}^3$, the intrinsic dimensionality of the shape space is $3N_a - 7$.

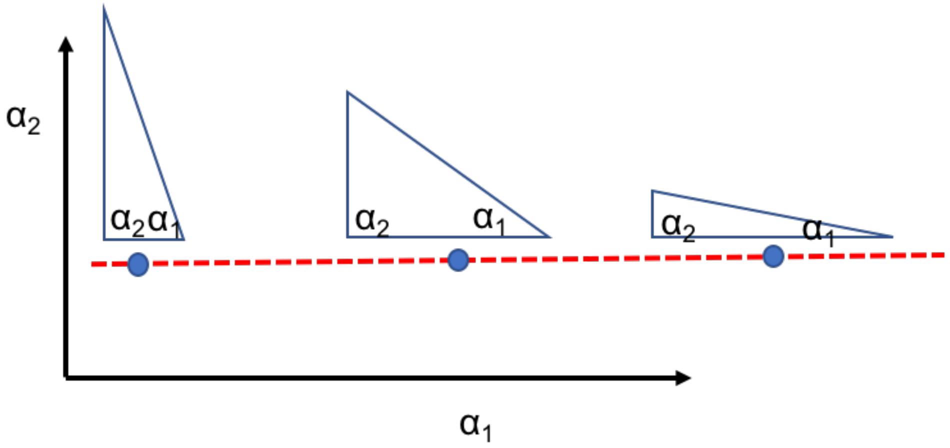


Figure 2.1: This diagram shows a simplified representation of the neighborhood of a point in the shape space Σ_2^3. Up to rotation, dilation, and translation, the shape of a triangle is determined by two angles, so we can see that this is a two-dimensional space. The diagram represents the logarithmic map of a region of Σ_2^3, with the red line indicating the logarithmic map of the subspace of right triangles, in a coordinate system given by α^1 and α^2, two angles in the triangle.

Chapter 3

MANIFOLD COORDINATES WITH PHYSICAL MEANING

Manifold embedding algorithms map high-dimensional data to coordinates in a much lower-dimensional space. One of the aims of dimension reduction is to find intrinsic coordinates that describe the data manifold. The coordinates returned by the embedding algorithm are abstract, and finding their physical or domain-related meaning is not formalized and often left to domain experts. This chapter studies the problem of recovering the meaning of the new low-dimensional representation in a semi-automatic and principled fashion. We propose a framework to explain embedding coordinates of a manifold as non-linear compositions of functions from a user-defined dictionary. We then show that this problem can be set up as a sparse linear Group Lasso recovery problem, and demonstrate its effectiveness on data.

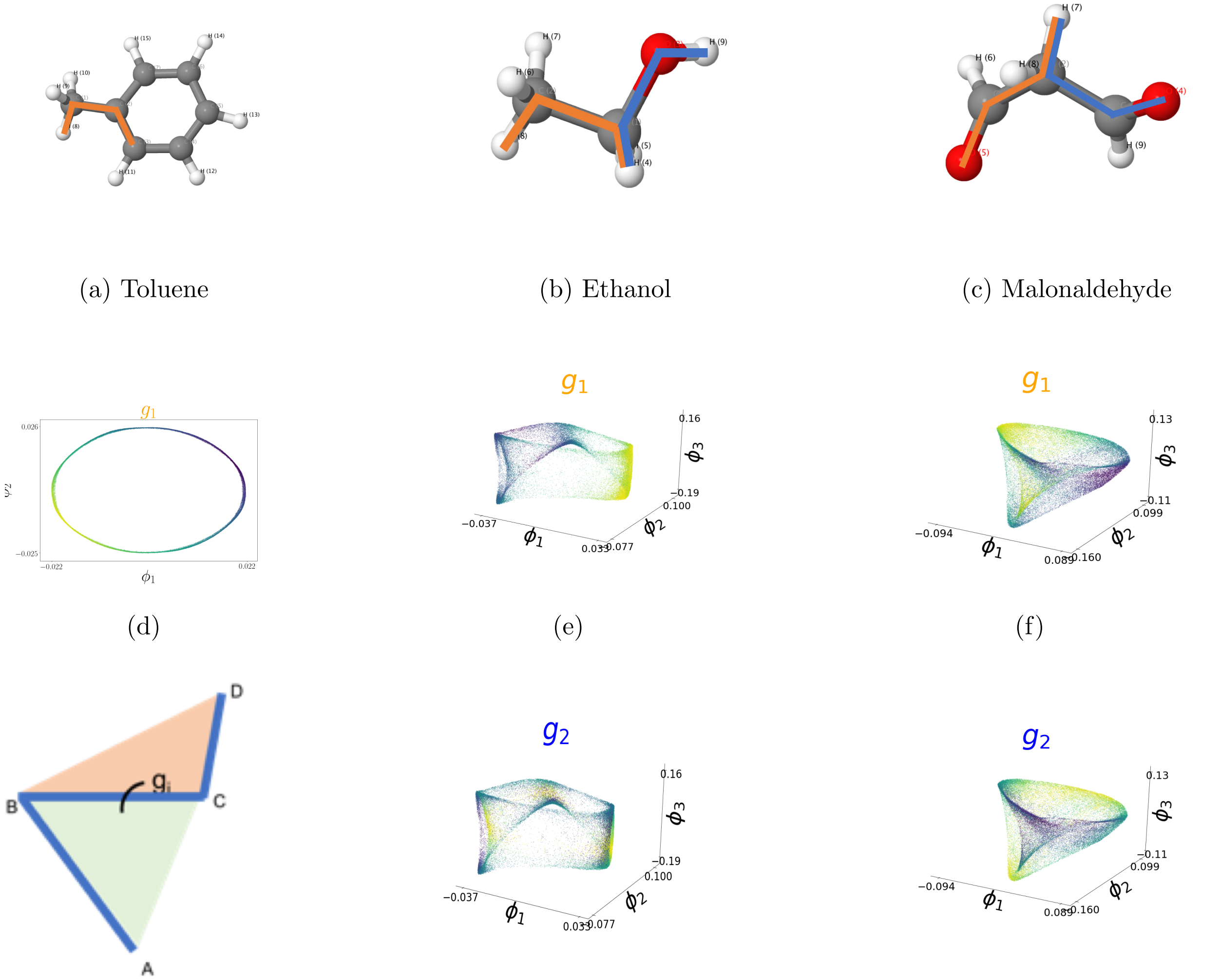
3.1 Introduction

This chapter describes an algorithm - Manifold Flasso (Functional Lasso) or Manifold Lasso for short - for interpreting the coordinates output by manifold learning (ML) algorithms. In the sciences, one of the motivating goals of dimension reduction is the discovery of descriptors of the data generating process. Linear dimension reduction algorithms like Principal Component Analysis (PCA) and non-linear algorithms such as Diffusion Maps (49) are used to uncover the variables describing large-scale properties of the interrogated system in applications from genomics to astronomy (12).

An example of this approach occurs in the analysis of MDS data. It has been shown empirically that manifolds approximate the high-dimensional distribution of simulated configurations configurations (58). Accordingly, application of manifold learning to find what are in this setting called the collective coordinates has achieved great success.

A key deficiency of such methods is with respect to the crucial notion of interpretability. A precise definition of interpretability is lacking, but justifications for its study coalesce around several important qualities. First, user-confidence in a learned representation may require knowledge of what is actually being represented. Second, an interpretable representation may be more likely to generalize, both intuitively, and in a statistical sense. Third, interpretable models may require less data to learn. Finally, interpretability increases causal understanding and ultimately the ability to intervene in a system to cause a desired effect. For this reason, the use of manifold learning for analysis of high-dimensional systems is often accompanied by a post-hoc analysis in which a domain-expert visually inspects the learned manifold for correspondences with interesting features.

We begin our study by demonstrating the standard scientific approach to this problem for several MD simulations in Figure 3.1. Figures 3.1a, 3.1b, and 3.1c show the toluene, ethanol, and malonaldehyde molecules, consisting of $N_a = 15, 9$ and 9 atoms, respectively, while 3.1d, 3.1e, 3.1h, 3.1f and 3.1i show the mappings of MD simulated trajectories into $m = 2, 3$ and 3 embedding coordinates by a manifold learning algorithm. Visual inspection shows that the learned low-dimensional manifolds representing these configuration spaces are parameterized (a technical term that we use here only intuitively) by certain functions of the molecular configuration. Here, these functions are **torsions** - the angles formed as in Figure 3.1g by the planes inscribing the first three and last three atoms of the colored lines joining four atoms in Figures 3.1a, 3.1b, and 3.1c. Thus, the displayed torsions are collective coordinate describing the slow motions of the toluene molecule, filtered out from the faster modes of vibration by the manifold learning algorithm.



(a) Toluene

(b) Ethanol

(c) Malonaldehyde

(d)

(e)

(f)

(g) Torsion example

(h)

(i)

Figure 3.1: Collective coordinates with physical meaning in MDS. 3.1a-3.1c Diagrams of the toluene (C_7H_8), ethanol (C_2H_5OH), and malonaldehyde ($C_3H_4O_2$) molecules, with the carbon (C) atoms in grey, the oxygen (O) atoms in red, and the hydrogen (H) atoms in white. Bonds defining important torsions g^j are marked in orange and blue. The bond torsion is the angle of the planes inscribing the first three and last three atoms on the line (3.1g). 3.1d Embedding of sampled configurations of toluene into $m = 2$ dimensions, showing a manifold of $d = 1$. The color corresponds to the values of the orange torsion g^1. 3.1e, 3.1h An embedding of configurations of ethanol in $m = 3$ dimensions showing a manifold of dimension $d = 2$ colored by the blue and orange torsions in Figure 3.1b. 3.1f, 3.1i: An embedding of configurations of malonaldehyde in $m = 3$ dimensions showing a manifold of dimension $d = 2$ colored by the blue and orange torsions in Figure 3.1c.

In this example, while the embedding algorithm was able to uncover the manifold structure of the data, finding the physical meaning of the manifold coordinates was done by visual inspection. In general, a scientist scans through many torsions and other functions of the configuration, in order to find ones that can be identified with the abstract coordinates output by a PCA or ML algorithm. Visual inspection for correspondences with features of interest is pervasive in a variety of scientific fields (42; 87). The goal of this chapter is to put this process on a formal basis and to devise a method for automating this identification, thus removing the time consuming visual inspections from the shoulders of the scientist.

In our paradigm, the scientist inputs a **dictionary** $\mathcal{G}$ of functions to be considered as possible collective coordinates. For the examples in Figure 3.1, $\mathcal{G}$ could be a set of candidate torsions. We propose an algorithm that recovers a set of functions $g^1, \dots g^s \in \mathcal{G}$, so that $(g^{1:s})$ is a local diffeomorphism to the output of the embedding algorithm; in other words, so that $(g^{1:s})$ are collective coordinates for the manifold. To keep the approach as general as possible, we do not rely on a particular embedding algorithm, making only the minimal assumption that it asymptotically produces a smooth embedding. We also do not assume a parametric relationship between the embedding and the functions in the dictionary $\mathcal{G}$. Instead, we only assume that these functions are sufficiently smooth.

Our idea is to compose differentials of covariate dictionary functions to reconstruct the differentials of the manifold embedding coordinates. By considering differentials, we map our original non-linear, non-parametric problem to a linear sparse regression robust to non-linearity in both the algorithm and the functional covariates.

The next section defines the problem formally, and Section 3.2 presents assumptions. Section 3.3 introduces gradient estimators that enable the linearized method in Section 3.4. Section 3.5 presents relevant theoretical results, and Section 3.6 presents experimental results of our method on simple synthetic data. Section 3.7 then prepares for application of our method to MDS data, including details about the dictionaries of interpretable functions, adaptions necessary to make our method work in the rotation and translation invariant molecular configuration space, while Section 3.8 presents results on molecular dynamics data.

Section 3.9 discusses a limited amount of related work and suggests some adaptations that will be explored in Chapters 4, 5, and 6. The bulk of discussion is deferred to Chapter 4.

3.2 Problem

We make a number of standard manifold learning assumptions. Observed data $\mathcal{D} = \{\xi_i \in \mathbb{R}^D : i \in 1 \dots n\}$ are sampled i.i.d. from a smooth manifold $\mathcal{M}$ of intrinsic dimension d embedded in a feature space $\mathbb{R}^D$ by the inclusion map. In this chapter, we call smooth any function or manifold of class at least $\mathcal{C}^3$. We assume that the intrinsic dimension d of $\mathcal{M}$ is known; for example, by having been estimated previously by one method in Kleindessner and Luxburg (104). The manifold $\mathcal{M}$ is a Riemannian with Riemannian metric inherited from the ambient space $\mathbb{R}^D$. Furthermore, we assume the existence of a smooth embedding map $\phi : \mathcal{M} \to \phi(\mathcal{M}) \subset \mathbb{R}^m$, where typically $m << D$. That is, ϕ restricted to $\mathcal{M}$ is a diffeomorphism onto its image, and $\phi(M)$ is a submanifold of $\mathbb{R}^m$. We call the coordinates $\phi(\xi_i)$ in this m dimensional ambient space the embedding coordinates $\phi^{1:m}$. In practice, the mapping of the data $\mathcal{D}$ onto $\phi(\mathcal{D})$ represents the output of an embedding algorithm, and we only have access to $\mathcal{M}$ and ϕ via $\mathcal{D}$ and its image $\phi(\mathcal{D})$.

Problem statement We are given a dictionary of user-defined and domain-related smooth functions $\mathcal{G} = \{g^1, \dots g^p, \text{ s.t. } g^j : U \subseteq \mathbb{R}^D \to \mathbb{R}\}$, where U is an open set containing $\mathcal{M}$. Our goal is to determine set $S \subset [p]$ such that $\phi = h \circ g_S$. Given coordinate functions $\phi^{1:m}$ and dictionary functions $g_{1:p}$ of a smooth manifold $\mathcal{M}$, our goal is to recover a subset g^S of $g^{1:p}$ such that $\phi^{1:m} = h \circ g_S$ without knowing h, we propose to recover the subset g^S by solving a set of dependent linear sparse recovery problems, one for each data point. That is, we assume that $\phi(x) = h(g^{j_1}(x), \dots g^{j_s}(x))$, where $h : O \subseteq \mathbb{R}^s \to \mathbb{R}^m$ is a smooth function of s variables, defined on a open subset of $\mathbb{R}^s$ containing the ranges of $g^{j_1}, \dots g^{j_s}$. Let $S = \{j_1, \dots j_s\}$, and $g^S = [g^{j_1}(x), \dots g^{j_s}(x)]^T$. We call this set the **functional support** or **explanation**. In differential geometric terms, g^S is related to finding coordinate systems, charts and parameterizations of $\mathcal{M}$. For example, in the toluene example, the functions in

$\mathcal{G}$ are all the torsions in the molecule, $s = 1$, and $g^S = g^1$ is a chart for the 1-dimensional manifold traced by the configurations. Hence, it is natural to associate $s = d$.

Indeterminacies Since the map ϕ given by the embedding algorithm is determined only up to diffeomorphism, the map h cannot be uniquely determined, and it can therefore be overly restrictive to assume a parametric form for h. Hence, this paper aims to find the support set S while circumventing the estimation of h. Indeterminacies w.r.t. the support S itself are also possible. For instance, the support S may not be unique whenever the relationship $g^1 = t(g^2)$, where t is a smooth function, holds for two functions $g^1, g^2 \in \mathcal{G}$. In Section 3.5 from (130) we give conditions under which S can be recovered uniquely; intuitively, they consist of functional independencies between the functions in $\mathcal{G}$. It is sufficient to assume that that the dictionary $\mathcal{G}$ is a **functionally independent** set, i.e. there is no $g \in \mathcal{G}$ that can be obtained as a smooth function of other functions in $\mathcal{G}$.

3.3 Gradients on manifolds

The main idea of our approach is to exploit that, for any differentiable functions f, g, h, when $f = h \circ g$, the differentials Df, Dh, Dg at any point are in the linear relationship $Df = DhDg$. Thus, the functional relationship $\phi = h \circ g^S$ will be written as the linear relationship $D\phi = DhDg^S$, or, in other words, in terms of gradients $\operatorname{grad}_{\mathcal{M}} \phi^{1:m}$ and $\operatorname{grad}_{\mathcal{M}} g^{1:p}$. Section 3.4 will explain how to identify the functional support using these gradients. Therefore, we review the definition of gradient on a manifold, and explain how to convert the analytically available gradients of the dictionary functions to gradients of functions on the manifold. We also introduce an estimator for the gradients of the embedding coordinates, which are not analytically available for our Diffusion Maps embeddings.

3.3.1 Gradients and coordinate systems

Our algorithm regresses the gradients of the embedding coordinate functions against the gradients of the dictionary functions. Both sets of gradients are with respect to the manifold

$\mathcal{M}$, and so this requires calculating or estimating various gradients in the same d-dimensional coordinate system.

By assumption we have two Euclidean spaces $\mathbb{R}^D$ and $\mathbb{R}^m$, in which manifolds $\mathcal{M}$ and $\phi(\mathcal{M})$ of dimension d are embedded. Denote gradients w.r.t. the Euclidean coordinate systems in $\mathbb{R}^D$ and $\mathbb{R}^m$ by ∇_ξ and ∇_ϕ, respectively. Since our interest is in functions on manifolds, we also recall the gradient of a function on a manifold $\mathcal{M}$ from Definition 2.9. We denote gradients expressed in bases $T_i^{\mathcal{M}}$ in $\mathcal{T}_{\xi_i}\mathcal{M}$ and T_i^{ϕ} in $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$ by $\operatorname{grad}_{T_i^{\mathcal{M}},\mathbf{g}} f$ and $\operatorname{grad}_{T_i^{\phi},\mathbf{g}} f$ respectively. For a manifold $\mathcal{M}$ which is a submanifold of $\mathbb{R}^D$, we denote by $\operatorname{grad}_{T_i^{\mathcal{M}}}(\xi)$ the value of $\operatorname{grad}_{T_i^{\mathcal{M}},\mathbf{id}}(\xi)$ w.r.t. the ambient identity metric $\mathbf{id}$ inherited from $\mathbb{R}^D$.Note that in coordinates, $\operatorname{grad}_{\mathcal{M}} f$ depends on the metric $\mathbf{g}$, but at the same time $\operatorname{grad}_{\mathcal{M}} f$ as a linear operator on $\mathcal{T}_{\xi_i}\mathcal{M}$ is invariant to the metric. Hence, the left hand side must also be invariant to the metric. It follows that $Df(\xi)u = u^T \operatorname{grad}_{T_i^{\mathcal{M}}} f(\xi)$ for any $u \in \mathcal{T}_\xi\mathcal{M}$, and, furthermore, that $\operatorname{grad}_{T_i^{\mathcal{M}},\mathbf{g}} f = G_i^{-1} \operatorname{grad}_{T_i^{\mathcal{M}}} f$ for any other Riemannian metric $\mathbf{g}$.

3.3.2 Calculating the gradients of the dictionary functions

Our goal is construct matrices X_i, for $i = 1, \dots n$, with p columns representing the gradients of the p dictionary functions in basis $T_i^{\mathcal{M}}$:

$$X_i \;:=\; [\operatorname{grad}_{T_i^{\mathcal{M}}} g^j(\xi_i)]_{j=1:p} \in \mathbb{R}^{d\times p}. \tag{3.1}$$

Denote $X = [X_i]_{i=1:n}$ - a $n \times d \times p$ dimensional array. We will index individual gradient vectors in this array as $X_{i.j}$. The gradients $\nabla_\xi g^j(\xi_i)$ are known analytically, by assumption. By definition, in any basis $T_i^{\mathcal{M}} \in \mathbb{R}^{D\times d}$ of $\mathcal{T}_{\xi_i}\mathcal{M}$,

$$\operatorname{grad}_{T_i^{\mathcal{M}}} g^j(\xi_i) \;=\; (T_i^{\mathcal{M}})^T \nabla_\xi g^j(\xi_i). \tag{3.2}$$

In other words, $\operatorname{grad}_{T_i^{\mathcal{M}}} g^j$ is the projection of $\nabla_\xi g^j$ on the orthonormal basis $T_i^{\mathcal{M}}$. We estimate these bases with the WLPCA algorithm described in Section 2.3.

3.3.3 Estimating gradients of coordinate functions

In contrast, the gradients of ϕ^k are often not analytically available, and ϕ^k is known only through its values at the data points. We introduce an estimator of these gradients based on the notion of vector pull-back between tangent spaces. Instead of estimating these gradients merely from differences $\phi^k(\xi_i) - \phi^k(\xi_{i'})$ between neighboring points, we first estimate their values in $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$, where they have a simple expression, and then pull them back in the coordinate system $T_i^{\mathcal{M}}$. This estimation method is novel, and of some independent interest.

The PullBackDPhi Algorithm takes as inputs the local neighborhoods Ξ_i, Φ_i of point ξ_i in the original and embedding spaces, respectively, the basis $T_i^{\mathcal{M}}$ of $\mathcal{T}_{\xi_i}\mathcal{M}$, and the row of the Laplacian matrix corresponding to i, $L_{i,\mathcal{N}_i}$. From this local information, the algorithm first computes a basis for the tangent space $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$, obtains the gradients of the coordinate functions ϕ in this basis by projection, and finally pulls them back in the coordinate system given by $T_i^{\mathcal{M}}$ by solving a least squares regression.

Estimating $\mathcal{T}\phi(\mathcal{M})$ using RMetric As an alternative to wlpca, we estimate the $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$ using the Riemannian metric $\mathbf{g}$, expressed as G_i in the coordinates ϕ at ξ_i (**?**). Recall our assumption that $(\mathcal{M}, \mathbf{id})$ is a Riemannian manifold, with the metric $\mathbf{id}$ induced from $\mathbb{R}^D$. With this we associate to $\phi(\mathcal{M})$ a Riemannian metric $\mathbf{g}$ which preserves the geometry of $(\mathcal{M}, \mathbf{id})$. This metric - the pushforward metric - is defined by

$$\langle u, v\rangle_{\mathbf{g}} \;=\; \langle D\phi^{-1}(\xi)u, D\phi^{-1}(\xi)v\rangle \quad \text{for all } u, v \in \mathcal{T}_{\phi(\xi)}\phi(\mathcal{M}). \tag{3.3}$$

In the above, $D\phi^{-1}(\xi)$ is the pull-back operator that maps vectors from $\mathcal{T}_{\phi(\xi)}\phi(\mathcal{M})$ to $\mathcal{T}_{\xi}\mathcal{M}$, and $\langle,\rangle$ the Euclidean scalar product. The matrices G_i can be estimated by the algorithm RMetric given in Section 2.3. The algorithm uses only local information, and thus can be run efficiently using the Laplacian, the neighborhood graph, and local embedding coordinate matrices. The theoretical rank of G_i equals d and the d principal eigenvectors of G_i represent an orthonormal basis $T_i^{\phi} \in \mathbb{R}^{m\times d}$ of $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$.

We use the output of this algorithm to estimate $D\phi(\xi)$. Trivially, the gradients of $\phi^{1:m}$ in the embedding space $\mathbb{R}^m$, are equal to the m basis vectors of $\mathbb{R}^m$, i.e. $\nabla_\phi \phi^{1:m} = I_m$. Therefore

$$[\mathrm{grad}_{T_i^\phi}\, \phi_k(\xi_i)]_{k=1}^m \;=\; (T_i^\phi)^T I_m. \tag{3.4}$$

In order to bring these gradients into the same coordinate system as our dictionary functions, we define the following matrices, with $\mathrm{Proj}_T\, v$ denoting the Euclidean projection of vector v onto subspace T.

$$\begin{aligned}
Y_i &= [y_{ik}]_{k=1}^m \;=\; [\mathrm{grad}_{T_i^{\mathcal{M}}}\, \phi_k(\xi_i)]_{k=1}^m \in \mathbb{R}^{d\times m}, && (3.5)\\
A_i &= \Big[\mathrm{Proj}_{\mathcal{T}_{\xi_i}\mathcal{M}}(\xi_{i'} - \xi_i)\Big]_{i'\in\mathcal{N}_i} \in \mathbb{R}^{d\times k_i}, && (3.6)\\
B_i &= [\phi(\xi_{i'}) - \phi(\xi_i)]_{i'\in\mathcal{N}_i} \in \mathbb{R}^{m\times k_i}, && (3.7)\\
\tilde{B}_i &= \Big[\mathrm{Proj}_{\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})}\, [\phi(\xi_{i'}) - \phi(\xi_i)]\Big]_{i'\in\mathcal{N}_i} \in \mathbb{R}^{d\times k_i}. && (3.8)
\end{aligned}$$

The columns of A_i and Y_i are vectors in $\mathcal{T}_{\xi_i}\mathcal{M}$, the columns of B_i are in $\mathbb{R}^m$ and the columns of $\tilde{B}_i$ are in $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$. Note that when $m = d$, $B_i = \tilde{B}_i$. The columns of A_i and $\tilde{B}_i$ are in correspondence because they represent namely the logarithmic maps of point i' with respect to point i (approximately) the same vectors in two coordinate systems of $\mathcal{T}_{\xi_i}\mathcal{M}$ and $\phi(\mathcal{M})$. The accuracy of this approximation is given by the following proposition Meila et al. (130).

Proposition 30.

$$(\mathrm{grad}_{T_i^{\mathcal{M}}}\, \phi^k(\xi_i))^T A_i \;=\; (\mathrm{grad}_{T_i^\phi}\, \phi^k(\xi_i))^T \tilde{B}_i + o(\epsilon). \tag{3.9}$$

Our estimator uses this correspondence in order to pull back the gradient of the coordinate function ϕ^k into the coordinates $T_i^{\mathcal{M}}$. We calculate the value of the differential $D\phi^k$ on the columns of $\tilde{B}_i$ in the coordinate system given by ϕ, and equate these values with $\mathrm{grad}_{T_i^{\mathcal{M}}}\, \phi^k$ applied to the columns of A_i. In coordinates, $A_i = (T_i^{\mathcal{M}})^T(\Xi_i - \xi_i \mathbf{1}_{k_i}^T)$ and $\tilde{B}_i = (T_i^\phi)^T(\Phi_i - \phi(\xi_i)\mathbf{1}_{k_i}^T)$. These matrices are computed by Steps 2 and 3 of Algorithm PullBackDPhi, while Y_i contains the gradients we want to estimate. Recalling that $Y_i = [\mathrm{grad}_{T_i^{\mathcal{M}}}\, \phi_k(\xi_i)]_{k=1:m}$ we obtain

$$Y_i^T A_i \;=\; [(T_i^\phi)^T I_m]^T (T_i^\phi)^T B_i + o(r_N). \tag{3.10}$$

We solve this linear system in the least squares sense

$$Y_i \;=\; \arg\min_{Y\in\mathbb{R}^{d\times m}} \|A_i^T Y - B_i^T T_i^{\phi}(T_i^{\phi})^T\|_2^2 \tag{3.11}$$

to obtain

$$Y_i \;=\; A_i^{\dagger} B_i^T T_i^{\phi}(T_i^{\phi})^T. \tag{3.12}$$

This solution is the regression of the columns of $B_i T_i^{\phi}(T_i^{\phi})^T$ on the columns of A_i at each data point ξ_i. We call estimator (3.12) the **pullback gradient estimator** because of its implicit invocation of the notion of vector pullback.

To justify this name, we note that by equation (3.3), for any function $f : \phi(\mathcal{M}) \to \mathbb{R}$,

$$\langle D\phi^{-1}(\phi(\xi))u, D\phi^{-1}(\phi(\xi))\,\mathrm{grad}_{\phi(\mathcal{M})} f(\xi)\rangle \;=\; \langle u, \mathrm{grad}_{\phi(\mathcal{M})} f(\phi(\xi))\rangle_{\mathbf{g}}, \quad \text{for all } u \in \mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M}) \tag{3.13}$$

where $\mathbf{g}$ is the push-forward metric associated with ϕ. Using this fact, and the invariance of gradient to metric, we have that, for any $w \in \mathcal{T}_{\xi_i}\mathcal{M}$, $D\phi^{-1}(\phi(\xi))\,\mathrm{grad}_{\phi(\mathcal{M})} f(\phi(\xi)) = \mathrm{grad}_{\mathcal{M}}(f \circ \phi)(\xi)$ for any smooth function $f : \phi(\mathcal{M}) \to \mathbb{R}$. The above claims give us $\langle D\phi^{-1}(\phi(\xi))u, \mathrm{grad}_{\mathcal{M}}(f\circ\phi)(\xi)\rangle = \langle u, \mathrm{grad}_{\phi(\mathcal{M})} f(\phi(\xi))\rangle$ where $u \in \mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$ is an arbitrary tangent vector. In coordinates T_i^{ϕ} and $T_i^{\mathcal{M}}$, we can write this equivalence as

$$\langle D\phi^{-1}(\phi(\xi))u, \mathrm{grad}_{T_i^{\mathcal{M}}}(f\circ\phi)(\xi)\rangle = \langle u, \mathrm{grad}_{T_i^{\phi}} f(\phi(\xi))\rangle. \tag{3.14}$$

If we then replace values of $(T_i^{\phi})^T e^k$, $(T_i^{\mathcal{M}})^T(\xi_{i'}-\xi_i)$ and $(T_i^{\phi})^T(\phi(\xi_{i'})-\phi(\xi_i))$ for $\mathrm{grad}_{T_i^{\phi}}\phi^k(\phi(\xi_i))$, $D\phi^{-1}(\phi(\xi_i))u$ and u, respectively, we obtain (3.10).

```
PullBackDPhi( local data $\Xi_i$, local embedding coordinates $\Phi_i$, basis $T_i^{\mathcal{M}}$ (Optional: $T_i^{\phi}$ or Laplacian row $L_{i,\mathcal{N}_i}$, intrinsic dimension $d$))
Compute pushforward metric eigendecomposition $G_i, T_i^{\phi} \leftarrow$ RMetric($L_{i,\mathcal{N}_i}, \Phi_i, d$).
Compute $B_i \leftarrow (\Phi_i^T - \phi(\xi_i)\mathbf{1}_{k_i}^T)$
Compute $A_i \leftarrow (T_i^{\mathcal{M}})^T(\Xi_i^T - \xi_i\mathbf{1}_{k_i}^T)$
Calculate $Y_i \leftarrow A_i^{\dagger}B_i^T T_i^{\phi}(T_i^{\phi})^T$ by solving linear system (3.11)
Output $Y_i$
```

3.4 The ManifoldLasso *algorithm*

Gradient estimators form the preliminary steps of the main ManifoldLasso algorithm, which takes as input data $\mathcal{D}$ sampled from an unknown manifold $\mathcal{M}$, a dictionary $\mathcal{G}$ of functions defined on $\mathcal{M}$ (or alternatively on an open subset of the ambient space $\mathbb{R}^D$ that contains $\mathcal{M}$), and an embedding $\phi(\mathcal{D}) \subset \mathbb{R}^m$, and outputs a set S of indices in $\mathcal{G}$, representing the functions in $\mathcal{G}$ that explain $\mathcal{M}$. The first part of the algorithm contains preparatory steps for geometric analysis covered in Section 2.3. Steps 1 and 2 construct the neighborhood graph, kernel matrix, and Laplacian matrix used for manifold learning and tangent space estimation. The second part of ManifoldLasso calculates the necessary gradients; this comprises Steps 8–10. In Step 8, we estimate orthogonal bases of tangent subspaces by the wlpca algorithm described in Section 2.3. The gradients of the dictionary functions w.r.t. the manifold in bases $T_i^{\mathcal{M}}$ are then obtained as columns of the $d \times p$ matrix X_i in Steps 4, 5, and 9. These operations are described in detail in Section 3.3.2. In Step 10, the gradients at ξ_i of the coordinates $\phi^{1:m}$, also in bases $T_i^{\mathcal{M}}$, are calculated as columns of the $d \times m$ matrix Y_i by the PullBackDPhi algorithm described in Section 3.3.3. In the last part of ManifoldLasso, Step 14 finds the support S by solving the sparse linear regression. A GroupLasso algorithm is called to perform the sparse regression of the manifold coordinates' gradients $Y_{1:n}$ on the gradients of the dictionary functions, represented by $X_{1:n}$. The indices of those dictionary functions whose β coefficients are not identically null represent the support set $S = \operatorname{supp} \beta$. This is described in Section 3.4.1. Scaling of functions is addressed through normalization in Steps 5 and 12; this procedure is described in more detail in Section 3.4.3.

Algorithm 4 ManifoldLasso(Dataset $\mathcal{D}$, dictionary $\mathcal{G}$, embedding coordinates $\phi(\mathcal{D})$, intrinsic dimension d, kernel bandwidth ϵ, neighborhood cutoff size r, regularization parameter λ)

```
 1: Construct $\mathcal{N}_i$ for $i = 1 : n$; $i' \in \mathcal{N}_i$ iff $\|\xi_{i'} - \xi_i\| \leq r$, and local data matrices $\Xi_{1:n}$
 2: Construct kernel matrix and Laplacian $K, L \leftarrow$ Laplacian($\mathcal{N}_{1:n}, \Xi_{1:n}, \epsilon$)
 3: [Optionally compute embedding: $\phi(\xi_{1:n}) \leftarrow$EmbeddingAlg($\mathcal{D}, \mathcal{N}_{1:n}, m, \ldots$)]
 4: for $j = 1, 2, \ldots p$ do
 5:     Compute $\nabla_\xi g^j(\xi_i)$ for $i = 1, \ldots n$
 6:     Compute $\zeta_j^2$ by (3.21) and normalize $\nabla_\xi g^j(\xi_i) \leftarrow (1/\zeta_j)\nabla_\xi g^j(\xi_i)$ for $i = 1, \ldots n$
 7: end for
 8: for $i = 1, 2, \ldots n$ do
 9:     Compute basis $T_i^{\mathcal{M}} \leftarrow$wlpca($\Xi_i, K_{i,\mathcal{N}_i}, d$)
10:     Project $X_{i..} \leftarrow (T_i^{\mathcal{M}})^T \nabla_\xi g^{1:p}$
11:     Compute $Y_{i..} \leftarrow$PullBackDPhi($\Xi_i, \Phi_i, T_i^{\mathcal{M}}, L_{i,\mathcal{N}_i}, d$)
12: end for
13: Compute $\zeta_k^2 \leftarrow \frac{1}{n}\sum_{i=1}^n \|Y_{i.k}\|^2$ (i.e. (3.20)), for $k = 1, \ldots m$ and
    normalize $Y_i \leftarrow Y_i\,\mathrm{diag}\{1/\zeta_{1:m}\}$, for $i = 1, \ldots n$.
14: $\beta \leftarrow$ GroupLasso($X, Y, \lambda\sqrt{mn}$)
15: Output $S = \mathrm{supp}\,\beta$
```

There are several optional steps and substitutions in this algorithm. An embedding can be computed in Step 3, or input separately by the user - we denote this step generically as EmbeddingAlg. Also, although we explicitly describe tangent space estimation methods of both $\mathcal{T}_{\xi_i}\mathcal{M}$ and $\mathcal{T}_{\phi(\xi_i)}\phi(\mathcal{M})$ in our algorithms, other approaches to estimate them may be used. Further variations and extensions are discussed in Section 3.9.

3.4.1 The GroupLasso *formulation*

We resolve the functional support problem by applying a group lasso to the estimated gradients. Recall that X_i defined in (3.1) contains the gradients of the dictionary functions g^j, and that $Y_{i.k} \in \mathbb{R}^d$, the k-th column of $Y_{i..}$, represents the coordinates of $\operatorname{grad}_{\mathcal{M}} \phi^k(\xi_i)$ in the chosen basis of $\mathcal{T}_{\xi_i}\mathcal{M}$. Further, given the assumption that $\phi = h \circ g^S$, let h^k be the k-th component of the vector valued function h, and denote

$$\beta_{ijk} = \frac{\partial h^k}{\partial g^j}(g^j(\xi_i)) \tag{3.15}$$

$$\beta = [\beta_{ijk}]_{i,k,j=1}^{n,pm} \tag{3.16}$$

$$\tag{3.17}$$

Then, from the identity $\operatorname{grad}_{\mathcal{M}} \phi^k = \operatorname{grad}_{\mathcal{M}}(h^k \circ g^S)$ and the chain rule, one obtains the following linear model.

$$Y_{ik} \;=\; \sum_{j=1}^{p} X_{i.j}\beta_{ijk} + \epsilon_{ik} \;=\; X_{i..}\beta_{i.k} + \epsilon_{ik} \text{ for all } i = 1:n, \text{ and } k = 1:m. \tag{3.18}$$

In the above regression of $Y_{1:n}$ on $X_{1:n}$, $\beta_{i.k}$ is the set of regression coefficients of y_{ik} onto X_i. If there is some h such that $\phi = h \circ g_S$, then the non-zero β_{ijk} coefficients are estimates of $\frac{\partial h^k}{\partial g^j}(\xi_i)$ for $j \in S$. Further, $\beta_{.j.}$ represents the regression coefficients corresponding to the effect of function g^j; therefore, the zero $\beta_{.j.}$ matrices indicate that $j \notin S$. Hence, in each $\beta_{i.k}$, only $|S|$ elements are non-zero. The term ϵ_{ik} is added to account for noise or model misspecification.

The key characteristic of the functional support that we leverage is that the same set S of coefficients will be non-zero for all i and k. Since solving Equation 3.15 for β is underdetermined, we use a sparsity inducing regularization that simultaneously zeros out entire β_j vectors. Thus, our problem can be naturally expressed as a Group Lasso (197), with p groups of size mn, consisting of the $\beta_{1:p}$ groups of coefficients of $\operatorname{grad}_{\mathcal{M}} g^{1:p}$. To solve it we minimize the following objective function w.r.t. β:

$$J_{M,\lambda}(X, Y, \beta) \;=\; \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{m} \|Y_{i.k} - X_{i..}\beta_{i.k}\|_2^2 + \frac{\lambda}{\sqrt{mn}}\sum_{j=1}^{p} \|\beta_{.j.}\|_F. \tag{3.19}$$

The first term of the objective is the least squares loss of regressing $Y_{1:n}$ onto $X_{1:n}$. The second is a regularization term, which penalizes each group β_j by its Euclidean norm. This encourages most $\beta_{.j.}$ groups to be identically 0. The normalization of the regularization coefficient λ by the group size mn follows Yuan and Lin (197) and takes into account that the least squares loss also grows proportionally to mn. The use of Group Lasso for sparse functional regression was introduced in Meila et al. (130).

Note that we can consider objective 3.19 to be a group lasso problem with block diagonal X and Y. It is convex in β and invariant to the change of basis $T_i^{\mathcal{M}}$. Let $\tilde{T}_i^{\mathcal{M}} = T_i^{\mathcal{M}} U$ be a different basis, with $U \in \mathbb{R}^{d \times d}$ a unitary matrix. Then, $\tilde{Y}_{ik.} = U^T Y_{ik.}$, $\tilde{X}_i = U^T X_i$, and $\|\tilde{Y}_{ik} - \tilde{X}_{i..}\beta_{i.k}\|_2^2 = \|Y_{i.k} - X_{i..}\beta_{i.k}\|^2$ for any $\beta_{i.k} \in \mathbb{R}^p$.

3.4.2 Computation

The first two steps of ManifoldLasso are construction of the neighborhood graph and estimation of the Laplacian L. As stated in Section 2.3, L is a sparse matrix, hence RMetric can be run efficiently by only passing values corresponding to one neighborhood at a time. Note that in our examples and experiments, Diffusion Maps is our chosen embedding algorithm, so the neighborhoods and Laplacian are already available, though in general this is not the case. The second part of the algorithm estimates the gradients and constructs arrays $Y_{1:n}, X_{1:n}$. The gradient estimation runtime, with Cholesky decomposition-based solvers, is $O(qd^2 + nd^3)$ where $q = \sum_{i=1}^n k_i$ is the number of edges in the neighborhood graph. The last major step is a call to the GroupLasso solver, which estimates the support S of ϕ. The computation time of each iteration in GroupLasso is $O(nmpd)$. Note that when using a standard group lasso solver, the computation time is $O(n^2m^2pd)$ due to the block-diagonal structure of the problem implicit in flattening the n by p by d covariate tensor. We therefore use our own implementation of proximal FISTA to solve this problem (32; 96). Finally, we typically perform the 'for' loop over a subset $\mathcal{I} \subset [n]$ of the original data. This replaces the n in the computation time with the smaller factor $|\mathcal{I}|$, while still enabling the embedding and the tangent spaces to be learned from the entire dataset. embedding. For each i, computing the local mean is $O(k_i D)$ and the tangent space is $O(k_i D^2 + k_i^3)$.

3.4.3 Normalization

As with many sparse regression methods, normalization is necessary to balance the relative influence of dictionary elements and embedding coordinates. Multiplying g^j by a non-zero constant and dividing its corresponding $\beta_{.j.}$ by the same constant leaves the reconstruction error of all y's invariant, but affects the norm $\|\beta_{.j.}\|_F$. Therefore, the relative scaling of the dictionary functions g^j can influence the recovered support S, by favoring the dictionary functions whose columns have larger norm. A similar effect is present if a particular embedding coordinate ϕ^k is rescaled by a constant. For example, multiplying a certain ϕ^k by a number close to zero will cause the penalty accrued by learned coefficients for that coordinate to be smaller than for the other coefficients, and for that ϕ^k to dominate support recovery.

We therefore normalize all $\operatorname{grad}_{T_i^{\mathcal{M}}} \phi^{1:m}$ and $\operatorname{grad}_{T_i^{\mathcal{M}}} g^{1:p}$ as follows. Denote f a function on $\mathcal{M}$, which can be either a coordinate function or a dictionary function. When f is defined on $\mathcal{M}$, but not outside $\mathcal{M}$, we calculate the **normalizing constant**

$$\zeta^2 \;=\; \frac{1}{n}\sum_{i=1}^{n} \|\operatorname{grad}_{T_i^{\mathcal{M}}} f(\xi_i)\|_2^2. \tag{3.20}$$

Then we set $f \leftarrow f/\zeta$. The above ζ is the finite sample version of $\|\operatorname{grad}_{\mathcal{M}} f\|_{L_2(\mathcal{M})}$, integrated w.r.t. the data density on $\mathcal{M}$. We apply this normalization to coordinate functions ϕ^k, but it could also be applied to functions g^j when they are defined only on $\mathcal{M}$.

When function f is defined on a neighborhood around $\mathcal{M}$ in $\mathbb{R}^D$, we compute the normalizing constant with respect to $\nabla_\xi f$. That is,

$$\zeta^2 \;=\; \frac{1}{n}\sum_{i=1}^{n} \|\nabla_\xi f(\xi_i)\|^2. \tag{3.21}$$

Then, once again, we set $f \leftarrow f/\zeta$. We apply this normalization to dictionary functions g_j. This favors dictionary functions whose gradients are nearly tangent to the manifold $\mathcal{M}$, and penalizes the g^j's which have large gradient components perpendicular to $\mathcal{M}$.

3.4.4 Tuning

Tuning parameters are often selected by cross-validation in Lasso-type problems. However, in this setting, the recovered support generally span the tangent space, and as discussed in Section 3.5, we are theoretically motivated to identify a size d support. Since the cardinality of the support decreases as the tuning parameter λ is increased, we thus base our choice of λ on matching the cardinality of the support to d. Sufficient conditions for this estimation strategy are given in Section 3.5. We perform a binary search over λ in the range $[0, \lambda_{\max}]$ to identify the optimal λ, which we call λ_0.

Proposition 31. *Let* $\lambda_{max} := \min_{\lambda > 0} \arg\min J_{M,\lambda}(X, Y, \beta) = 0$*, the theoretical maximum* λ *value.*

$$\lambda_{max} = \max_{j \in [p]} (\sum_{i=1}^{n} \sum_{k=1}^{m} (\operatorname{grad}_{T_i^{\mathcal{M}}} g^j(\xi_i))^T (\operatorname{grad}_{T_i^{\mathcal{M}}} \phi^m(\xi_i)))^{1/2}. \tag{3.22}$$

Proof. Consider partial derivatives $\beta' := \frac{\partial J_{M,\lambda}(X,Y,\beta)}{\beta}$. At all $\lambda \geq \lambda_{max}$, the minimizer of objective $J_{M,\lambda}(X, Y, \beta)$ satisfies $\beta = 0$, by construction. Differentiating $J_{M,\lambda}(X, Y, \beta)$, we see that $\|\beta'_{.j.}\| = (\sum_{i=1}^{n} \sum_{k=1}^{m} (\operatorname{grad}_{T_i^{\mathcal{M}}} g^j(\xi_i))^T (\operatorname{grad}_{T_i^{\mathcal{M}}} \phi^m(\xi_i)))^{1/2} + \lambda$. Thus, when $\lambda < \sum_{i=1}^{n} \sum_{k=1}^{m} (\operatorname{grad}_{T_i^{\mathcal{M}}} g^j(\xi_i))^T (\operatorname{grad}_{T_i^{\mathcal{M}}} \phi^m(\xi_i)))^{1/2}$, $\|\beta_{.j.}\| > 0$. □

3.5 *Theory*

Theoretical analyses motivating the pullback estimator, sufficient conditions for functional support recovery, and convergence rate analyses are provided in (130). This section reviews and comments on these results. These comments motivate the alternative sparse estimation methodology in Chapter 5, as well as the gradient and tangent space estimators intoduced in Chapter 6.

3.5.1 Approximating the logarithmic map by orthogonal projection

The logarithmic map $\log_\xi \xi'$ of a neighboring point $\xi' \in \mathcal{M}$ w.r.t. ξ plays a central role in gradient estimation since it gives the mathematical framework for relating distances to neighbors sampled from $\mathcal{M}$ to vectors in $\mathcal{T}_\xi\mathcal{M}$. In (130), the accuracy of the approximation to logarithmic map by orthogonal projection in section 3.3.3 is shown to be a corollary of

Proposition 32. *For all ξ not on the boundary of $\mathcal{M}$ and all ξ' such that $\|\xi' - \xi\| \leq r$ for some $r = C\epsilon$, it holds that*

$$\| \operatorname{Proj}_{\mathcal{T}_\xi\mathcal{M}}(\xi' - \xi) - \log_\xi \xi'\| = o(\epsilon) \tag{3.23}$$

$$\| \operatorname{Proj}_{\mathcal{T}_{\phi(\xi)}\phi(\mathcal{M})}(\phi(\xi') - \phi(\xi)) - \log_{\phi(\xi)} \phi(\xi')\| = o(\epsilon). \tag{3.24}$$

However, not all gradient estimators require projection onto the tangent space in all the senses we have required. Given coordinates $T_{\phi(\xi)}M$, just as the projection $T^{\phi}_{\phi(\xi)}{}^T(\phi(\xi') - \phi(\xi))$ is an approximation to $\log_{\phi(\xi)} \phi(\xi')$ in the coordinates of $T^{\phi}_{\phi(\xi)}$, $\phi(\xi') - \phi(\xi)$ is itself an approximation to $T^{\phi}_{\phi(\xi)} \log_{\phi(\xi)} \phi(\xi')$. Since this is actually what is being approximated by columns of B_i, the accuracy of this approximation can be bounded instead. We therefore show

Proposition 33. $\|(\phi(\xi') - \phi(\xi)) - T^{\phi}_{\phi(\xi)} \log_{\phi(\xi)} \phi(\xi')\| = o(\epsilon)$.

Proof. Since $\phi(\mathcal{M})$ is a submanifold of $\mathbb{R}^m$, within a neighborhood, we can write it in coordinates as $(u, p(u))$ where $u = [u^1 \dots u^d]$ are coordinates of the tangent space $\mathcal{T}_{\phi(\xi)}\phi(\mathcal{M})$ within $\mathbb{R}^m$ and $p(u)$ is a map from $\mathbb{R}^d \to \mathbb{R}^{m-d}$. Within the same neighborhood define $\tilde{p}(u) = [u, p(u)]$ as well. Then,

$$\phi(\xi') = \tilde{p}(T^{\phi(\mathcal{M})}_{\phi(\xi)}{}^T \phi(\xi')). \tag{3.25}$$

Now each of the m component functions of $\tilde{p}$ has Taylor expansion

$$\tilde{p}^k(\operatorname{Proj}\phi(\xi')) = \phi(\xi) + \nabla\tilde{p}^k(\phi(\xi))^T(T^{\phi(\mathcal{M})}_{\phi(\xi)}{}^T(\phi(\xi') - \phi(\xi))) + o(\epsilon) \tag{3.26}$$

Thus,

$$\begin{aligned}
&\|\phi(\xi') - T^{\phi(\mathcal{M})}_{\phi(\xi)} T^{\phi(\mathcal{M})}_{\phi(\xi)}{}^T \phi(\xi')\| && (3.27)\\
&= \|\phi(\xi) + D\tilde{p}(\phi(\xi))(T^{\phi(\mathcal{M})}_{\phi(\xi)}{}^T(\phi(\xi') - \phi(\xi))) - T^{\phi(\mathcal{M})}_{\phi(\xi)} T^{\phi(\mathcal{M})}_{\phi(\xi)}{}^T \phi(\xi')\| + o(r) && (3.28)\\
&= \|D\tilde{p}(\phi(\xi))(T^{\phi(\mathcal{M})}_{\phi(\xi)}{}^T(\phi(\xi'))) - T^{\phi(\mathcal{M})}_{\phi(\xi)} T^{\phi(\mathcal{M})}_{\phi(\xi)}{}^T \phi(\xi')\| + o(r) && (3.29)\\
&= o(r). && (3.30)
\end{aligned}$$

To show the proposition, we combine this result with the following using the triangle inequality. Assume w.l.o.g. that $\log_{\phi(\xi)} \phi(\xi')$ is given in the coordinates of $T^{\phi(\mathcal{M})}_{\phi(\xi)}$. Following the proof of in Appendix A of (130), $\|{T^{\phi}_{\phi(\xi)}}^T(\phi(\xi') - \phi(\xi)) - \log_{\phi(\xi)} \phi(\xi')\| = o(r)$. Thus, $\|T^{\phi}_{\phi(\xi)}{T^{\phi}_{\phi(\xi)}}^T(\phi(\xi') - \phi(\xi)) - T^{\phi}_{\phi(\xi)} \log_{\phi(\xi)} \phi(\xi')\| = o(r)$. Since $T^{\phi}_{\phi(\xi)}{T^{\phi}_{\phi(\xi)}}^T \phi(\xi) = \phi(\xi)$, this can be rewritten as $\|T^{\phi}_{\phi(\xi)}{T^{\phi}_{\phi(\xi)}}^T \phi(\xi') - \phi(\xi) - T^{\phi}_{\phi(\xi)} \log_{\phi(\xi)} \phi(\xi')\| = o(r)$.

□

In Chapter 6, we will discuss a related gradient estimator that does not project onto $\mathcal{T}_{\phi(\xi)}\phi(M)$ prior to solving a linear system (17).

3.5.2 Functional Support

Our definition of functional support raises the question of what conditions guarantee existence and uniqueness of such as g^S within $\mathcal{G}$. These conditions are functional dependence conditions, defined as follows.

Definition 34. *We say that a set of functions g^S on a metric space X is C^ℓ* ***functionally dependent*** *at ξ if there is a subset $S' \subset S, S' \neq S$, a function $\tau : \mathbb{R}^{|S'|} \to \mathbb{R}^{|S|}$ and a neighborhood U around ξ such that*

- *$g^S = \tau \circ g^{S'}$ on U*
- *τ is C^ℓ (smooth) globally on $g^{S'}(U) \subset \mathbb{R}^{|S'|}$*
- *$y - \tau(y^{S'}) \not\equiv 0$ for y in any neighborhood $U \subset \mathbb{R}^{|S|}$ containing $g^S(\xi)$, where $y = (y^1, \cdots, y^{|S|}) \in \mathbb{R}^{|S|}, y^{S'} = (y^i)_{i \in S'} \in \mathbb{R}^{|S'|}$.*

We call a set of functions g^S ***functionally independent*** *if it does not contain any subset S' s.t. $g^S = \tau(g^{S'})$.*

If $\phi = h \circ g^S$ and g^S is functionally independent, then it is a minimal explanation for the manifold. For our problem, this definition applies in the following way.

Proposition 35. *Let $\mathcal{G}$ and g^S be defined as before. $\mathcal{M}$ is a smooth manifold with dimension d embedded in $\mathbb{R}^D$. Suppose that $\psi : \mathcal{M} \subset \mathbb{R}^D \to \mathbb{R}^m$ is also an embedding of $\mathcal{M}$ and has a decomposition $\psi(\xi) = h \circ g^S(\xi)$ for every $\xi \in \mathcal{M}$ where h is smooth. If the dictionary g^S contains d functions denoted by $g^{S'}$, that are smooth functionally independent on $\mathcal{M}$, then there exists a $\widetilde{h}$ such that $\psi = \widetilde{h} \circ g^{S'}$ on every $\xi \in \mathcal{M}$. Here, the function $\widetilde{h}$ is smooth almost everywhere in the range of $g^{S'}$.*

The first important feature of this proposition is that, if we have a cardinality d functionally independent set $g^{S'}$ within our dictionary, we can write $\phi(\xi) = h(g^{S'}(\xi))$. By the definition of functional independence, there does not exist a lower-cardinality subset $g^{S''}$ such that $g^{S'} = \tau \circ g^{S''}$. Therefore, there cannot exist a $h' = h \circ \tau$ satisfying $\phi = h' \circ g^{S''}$. Thus, the set $g^{S'}$ is minimal.

The other important feature of the proposition is the cardinality d of the set $g^{S'}$. The reason that this cardinality is required to be equal to the dimension $\mathcal{M}$ is that the gradients of $g^{S'}$ must span $\mathcal{T}_\xi\mathcal{M}$ in order for the implicit function theorem to show existence of h.

Proposition 36. *Suppose $\mathcal{M}$ is a $d-$dimensional smooth manifold and $g^S : \mathcal{M} \to \mathbb{R}^d$ are d C^ℓ functions. Suppose $g^S(\mathcal{M})$ has a positive measure in $\mathbb{R}^d$. Then they are functionally independent on $\mathcal{M}$ iff $\operatorname{rank} Dg^S(\xi) = d$ everywhere on $\mathcal{M}$ except for a closed subset $W \subset \mathcal{M}$ with no interior point.*

This requirement gives us a practical condition for checking for the presence of such a set. Since, by Sard's Theorem, the measure of W is 0, so if we estimate $Dg^S = [\operatorname{grad}_M g^1, \ldots \operatorname{grad}_M g^d]$, we can check if this rank condition is satisfied for any $g^S \subset \mathcal{G}$. The existence of such functions g^S with rank d almost everywhere is guaranteed by the fact that a single coordinate chart can cover any compact manifold except for a set of measure zero, known as the cut-locus of the chart (169; 25). One can, for example, find one function explaining the whole circle S^1 embedded in $\mathbb{R}^2$ except one point. Thus, these theoretical results should be considered conditions on the dictionary, rather than the manifold itself.

3.5.3 Convergence analysis

In addition to such asymptotic infinite sample existence results, one can derive sufficient conditions for finding a suitable functional support g_S using Program (3.19). Since, by Proposition 36, gradients $[\operatorname{grad}_{T_i^{\mathcal{M}}} g^1(\xi_i) \dots \operatorname{grad}_{T_i^{\mathcal{M}}} g^d(\xi_i)]$ must span the tangent space $\mathcal{T}_{\xi_i}\mathcal{M}$, these conditions are related to the conditions for which Program 2.15 recovers a minimizer of 2.14, and Lasso theory for finding such a solution can be adapted to our setting Wainwright (187); Tibshirani (180). These highlight the connections of dictionary functions collinearity to error resulting from estimation of $\operatorname{grad}_{T_i^{\mathcal{M}}} g^j$ and $\operatorname{grad}_{T_i^{\mathcal{M}}} \phi^k$.

According to (144), a particular group S will be recovered by Group Lasso methods, if (i) it is close to perpendicular to the linear subspace generated by all other groups, and (ii) group features in S are close to orthogonal. The first condition will be discussed later in this Section. As for condition (ii), we note that if a set S' is not full rank on $\mathcal{M}$, the Jacobian $Dg^{S'}$ will be ill-conditioned at the data near the critical points, which will result in very large β_{ijk} values. Hence, such a subset will be heavily penalized. Similarly, features g^j which vary much in a direction normal to $\mathcal{M}$ will have, due to the gradient normalization, smaller values for $\operatorname{grad}_{T_i^{\mathcal{M}}} g^j$; therefore their β_j coefficients will be large relatively to the coefficients of functions whose gradients are tangent to $\mathcal{M}$.

We introduce several relevant quantitites. As an abuse of notation, rearrange X into $\mathbb{R}^{n \times p \times d}$. The **incoherence** of $\mathcal{G}$ is defined as

$$\mu = \max_{i=1:n, j \in [p], j' \in S, j \neq j'} \frac{|X_{i.j}^T X_{i.j'}|}{\|X_{i.j}\| \|X_{i.j'}\|}. \tag{3.31}$$

Note that this definition differs in several ways from the standard definition of incoherence as $\max_i \max_{j,j' \in [p]} |X_{i.j}^T X_{i.j'}|$. Since the values of $X_{ij.}$ consist of gradients after projection, we assume the normalizations in Section 3.4.3. Second, as we have conditioned on the set S, it is not necessary to require that the gradients outside the support S be incoherent. Finally, the due to the group structure being shared across data points, μ is a maxima across $[n]$.

Assuming that that the data $Y_{1:n} \in \mathbb{R}^d$ satisfy the noise model

$$Y_i = \sum_{j=1}^{p} \beta^*_{ij} X_{i.j} + \epsilon_i \text{ for } i \in [n]. \tag{3.32}$$

for error associated with estimation of the embedding and tangent spaces, we also define the **noise level** σ

$$\max_{i \in [n]} \|\epsilon_i\|_2^2 = d\sigma^2. \tag{3.33}$$

We now claim the following results relating coherence and colinearity with support recovery and coefficient estimation.

Proposition 37. *Assume that Model (3.32) holds, and that* $\sum_{i=1}^n ||X_{i.j}||^2 = \gamma_j^2$ *for all* $j = 1 : p$*. Let* $\gamma_{max} = \max_{j \notin S} \gamma_j$*,* $\kappa_S = \max_{i=1:n} \frac{\max_{j\in S} \|X_{i.j}\|}{\min_{j \in S} \|X_{i.j}\|}$*. Denote by* $\bar{\beta}$ *the minimizer of Objective 3.19 for some* $\lambda > 0$*. If* $1 - (s-1)\mu > 0$ *and*

$$\gamma_{max} \left(\frac{\mu}{1-(s-1)\mu} \frac{\kappa_S}{\min_{i=1}^n \min_{j' \in S} \|X_{i.j'}\|} + \frac{\sigma\sqrt{d}}{\lambda\sqrt{n}} \right) \ \leq \ 1 \tag{3.34}$$

then $\bar{\beta}_{ij} = 0$ *for* $j \notin S$ *and all* $i = 1, \ldots n$*.*

The factor $\frac{\mu}{1-(s-1)\mu}$ measures the near-orthogonality of the gradients in S, while the factors $(\min_{i=1}^n \min_{j' \in S} \|X_{i.j'}\|)^{-1}$ and κ_S measure the conditioning of S with respect to the gradient norms. They are optimal when all gradients in S are bounded away from 0, and when their sizes are relatively equal. The second term depends on the noise amplitude, and can be made arbitrarily small by increasing the regularization coefficient λ. The drawback of increasing λ in this manner is that coefficient estimates are increasingly biased. However, given the following conditions, exact support recovery is guaranteed.

Proposition 38. *Assume that Model 3.32 and condition (3.34) hold. Let* $\kappa = \frac{\mu}{1-(s-1)\mu} \frac{\kappa_S}{\min_{i=1}^n \min_{j' \in S} \|X_{i.j'}\|}$ *and* $\gamma_S = \|X_{..S}\|_F$*. Denote by* $\hat{\beta}$ *the minimizer of Objective 3.19 for some* $\lambda > 0$*. If (1)* $\lambda = c\frac{\gamma_{max}\sigma\sqrt{d}}{1-\kappa\gamma\,\mathrm{max}}$*,* $c > 1$*, and (2)* $\|\beta^*_{.j.}\| > \sigma\sqrt{d}(\gamma_{max} + \gamma_S) + \lambda(1+\sqrt{s})$ *for all* $j \in S$*, then the support* S *is recovered exactly and*

$$\|\hat{\beta}_{.j.} - \beta^*_{.j.}\|_F < \sigma\sqrt{d}(\gamma_{max} + \gamma_S) + \lambda(1+\sqrt{s}) = \sigma\sqrt{d}\gamma_{max}\left[1 + \gamma_S/\gamma_{max} + c\frac{1+\sqrt{s}}{1-\kappa\gamma_{max}}\right] \text{ for all } j \in S. \tag{3.35}$$

These results are similar to literature on undercomplete sparse coding, in which a typical theoretical objective is guaranteeing so-called sparsistency and consistency with respect to the oracle support. However, in practice these results can be challenged by violations of support recovery conditions. We address these issues in Chapter 5.

3.6 Experiments 1: Set-up and Swiss Roll

We demonstrate the ability of ManifoldLasso to identify explanations of manifolds and their embedding coordinates in several toy and scientific manifold learning problems. This section describes the general experimental procedure for all experiments, as well as results for the classic **SwissRoll** dataset. Subsequently, Section 3.7 describes adjustments necessary for analyzing MD data, and Section 3.8 gives toy and scientific molecular data analyses. [1]

3.6.1 Experimental setup

For all experiments, the data consist of n data points in D dimensions, as well an embedding $\phi^{1:m}(\mathcal{D})$. We assume access to the manifold dimension d, a kernel bandwidth ϵ used in the estimation of the tangent spaces, and p dictionary functions. Except where otherwise specified, m and ϵ are used in the preliminary step of generating embeddings $\phi^{1:m}$ using the Diffusion Maps algorithm as EmbeddingAlg. ManifoldLasso is applied to a uniformly random subset of size $n' = |\mathcal{I}|$ and this process is repeated ω number of times. These parameters are passed to the Laplacian, wlpca, RMetric, and PullBackDPhi algorithms, and are summarized in Table 3.1. The regularization parameter λ ranges over $[0, \lambda_{\max}]$ as described in Section 3.4.4.

3.6.2 Swiss roll

We begin our experimental study by demonstrating that ManifoldLasso is invariant to the choice of embedding algorithm on the classic unpunctured **SwissRoll** dataset. This dataset

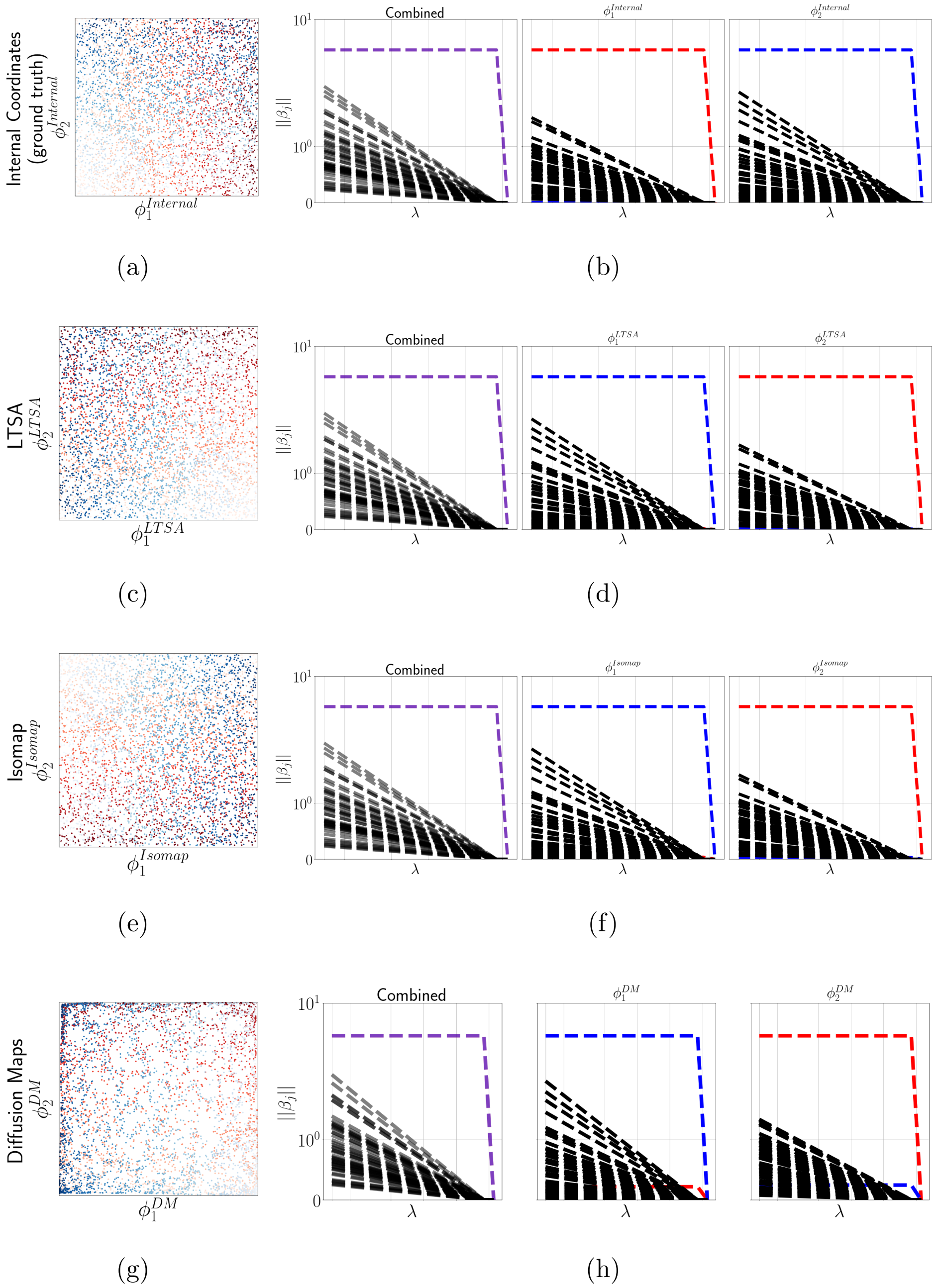
[1] Code is available at `https://github.com/sjkoelle/montlake/tree/master/montlake`

Dataset	n	N_a	D	d	ϵ_N	m	n'	p	ω
SwissRoll	10000	NA	49	2	.18	2	100	51	1
RigidEthanol	10000	9	50	2	3.5	3	100	12	25
Ethanol	50000	9	50	2	3.5	3	100	12	25
Malonaldehyde	50000	9	50	2	3.5	3	100	12	25
Toluene	50000	15	50	1	1.9	2	100	30	25
Ethanol	50000	9	50	2	3.5	3	100	756	25
Malonaldehyde	50000	9	50	2	3.5	3	100	756	25

Table 3.1: Summary of experiments. **SwissRoll** and **RigidEthanol** are toy data, while **Toluene**, **Ethanol**, and **Malonaldehyde** are from quantum molecular dynamics simulations by (47). The columns list the following experimental parameters: n is the sample size for manifold embedding, N_a is the number of atoms in the molecule, D is the dimension of ξ, d is the intrinsic dimension, ϵ_N is the kernel bandwidth, m is the embedding dimension, n' is the size of the subsample used for ManifoldLasso, p is the dictionary size, and ω is the number of independent repetitions of ManifoldLasso. More details are in Section 3.6.1

consists of points sampled from a two dimensional rectangle and rolled up along one of the two axes, then randomly rotated in $D = 49$ dimensions. We learn the manifold using three techniques: Local Tangent Space Alignment, Diffusion Maps, and Isomap, shown in Figures 3.2c, 3.2e and 3.2g. For comparison, we also analyze the "trivial embedding" consisting of coordinates given by projection onto the rectangle edges (Figure 3.2a). These intrinsic rectilinear coordinates are colored in red and blue, and show clear associations with individual embedding coordinates.

The dictionary $\mathcal{G}$ consists of $g^{1,2}$, the two intrinsic coordinates, as well as $g^{j+2} = \xi_j$, for $j = 1, \ldots 49$, the coordinates of the feature space. Applying ManifoldLasso to the embeddings identifies the set $S = \{g^1, g^2\}$ as the manifold explanation, and identifies the association of the recovered support with individual embedding coordinates $\phi^{1,2}$. By visual inspection of Figures 3.2a, 3.2c, 3.2e, and 3.2g, we see that all embedding algorithms recover the original manifold, although the embeddings $\phi^{\mathrm{Iso}}, \phi^{DM}, \ldots$ are not isometric (this is more noticeable with Diffusion Maps), and sign changes are possible. However, Figures 3.2b, 3.2d, 3.2f and 3.2h demonstrate that ManifoldLasso recovers the two manifold-specific coordinate functions in each case, while the coefficients $\beta_{.,3:51,.}$ decay rapidly to 0 with λ. Furthermore, each of $g^{1,2}$ is always mapped to the correct embedding coordinate. The regularization paths are virtually identical for all embeddings, even though the embeddings are not isometric.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 3.2: Results for **SwissRoll** embedded using several ML algorithms. 3.2a: the data mapped w.r.t. the edges of the rectangle. 3.2c, 3.2e, and 3.2g: embeddings colored by intrinsic coordinates in red and blue. 3.2b, 3.2d, 3.2f, 3.2h: regularization paths of ManifoldLasso for these embeddings. Combined norms $\|\beta_{.j.}\|_F$ used in ManifoldLasso are given on the left. Norms for individual embedding coordinates $\|\beta_{.jk}\|_2$ are on the right.

3.7 Molecular dynamics

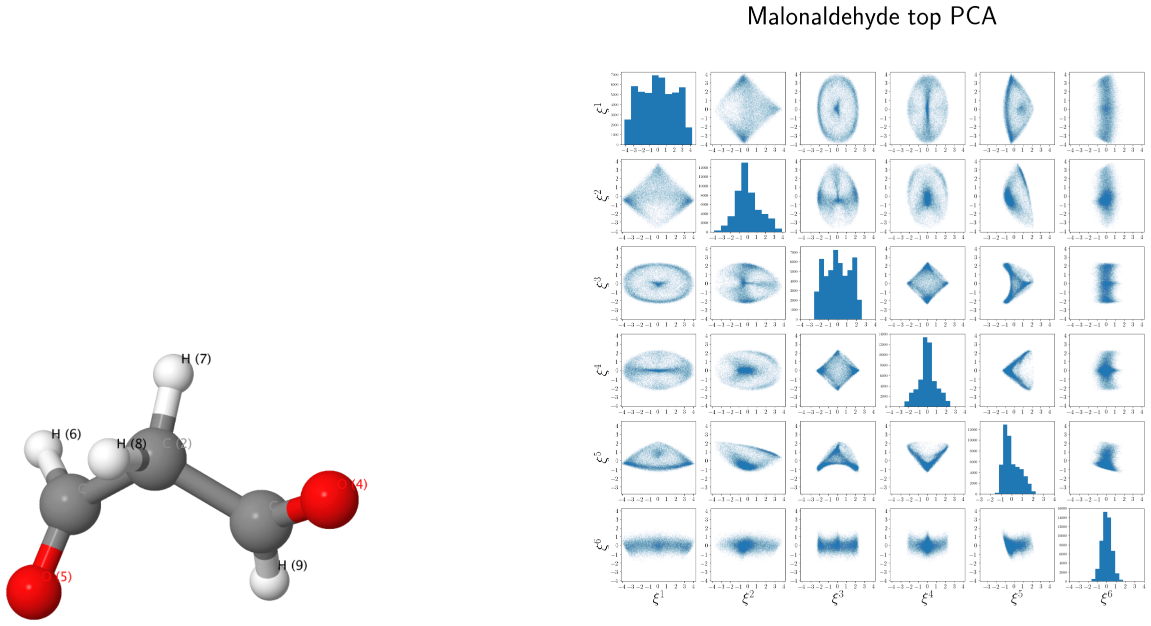
From the machine learning point of view, MD data from well-studied molecules are an excellent testbed for ManifoldLasso. Not only are MD data challenging problems for manifold learning, interpretations of learned manifolds in terms of physical features are critical for downstream tasks. High quality MD data are highly expensive to generate, taking weeks or months of supercomputer time (31; 69). Every new simulation represents a new manifold, and a new manifold explanation problem, and so fast automated analysis of these data by identification of so called collective variables serves both in the scientific understanding of the data and in acceleration sampling methods (160).

Our application of ManifoldLasso to MDS data entails several preliminary steps. These include the choice of a featurization in which to learn the manifold, the design of the dictionary, and the expression of the gradients of the dictionary functions in the translation and rotation-invariant molecular shape space. This section explains our approaches to these steps.

3.7.1 Representing molecular configurations

Our MD data are quantum-simulations from (47). The raw data consists of X, Y, Z coordinates for each of the N_a atoms of the chosen molecule. For a single observation, we denote these by $r_i \in \mathbb{R}^{3N_a}$. The first step in our data analysis pipeline is to featurize the configuration in a way that is invariant to rotation and translation. In the present experiments, we follow (44) and represent a molecular configuration as a vector $a_i \in \mathbb{R}^{3\binom{N_a}{3}}$ of the *planar angles* formed by triplets of atoms. We then perform an SVD on this featurization, and project the data onto the top $D = 50$ singular vectors to remove linear redundancies; we denote the new data points by $\xi_{1:n}$. The EmbeddingAlg and wlpca algorithms work directly with ξ in dimension D. Other possible representations such as applying a Procrustes transform to each configuration to align it with the first one give similar results, and empirically no matter which low level representation we choose, large-scale conformational changes are described by the relative rotations of groups of atoms - the bond torsions illustrated in Figure 3.1 (44).

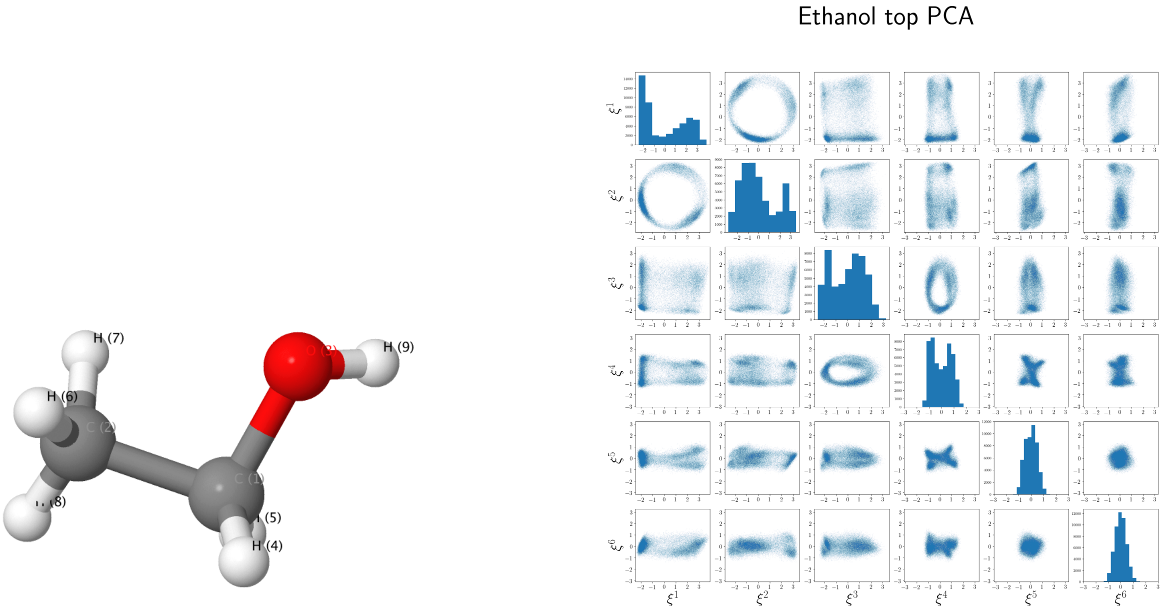
We display scatterplots of pairs of the top features in feature space $\mathbb{R}^D$ containing data points ξ in Figures Recall that PCA is applied as a preprocessing step prior to Manifold-Lasso, and so the PCA coordinates therefore form our feature space. PCA coordinates have a natural ordering given by their corresponding eigenvalues, and so we are able to plot the 'top' coordinates. Multiscale non-i.i.d. noise and non-trivial topology and geometry of data are present the PCA feature space. Note also that the manifolds are relatively thin in comparison to some noise dimensions; in other words the manifold reach is of the same scale as the noise. This "noise" is non-uniformly distributed and highly correlated with position on the manifold, varying, for example, with position along the circle embedded in the first and second coordinates.



(a)

(b)

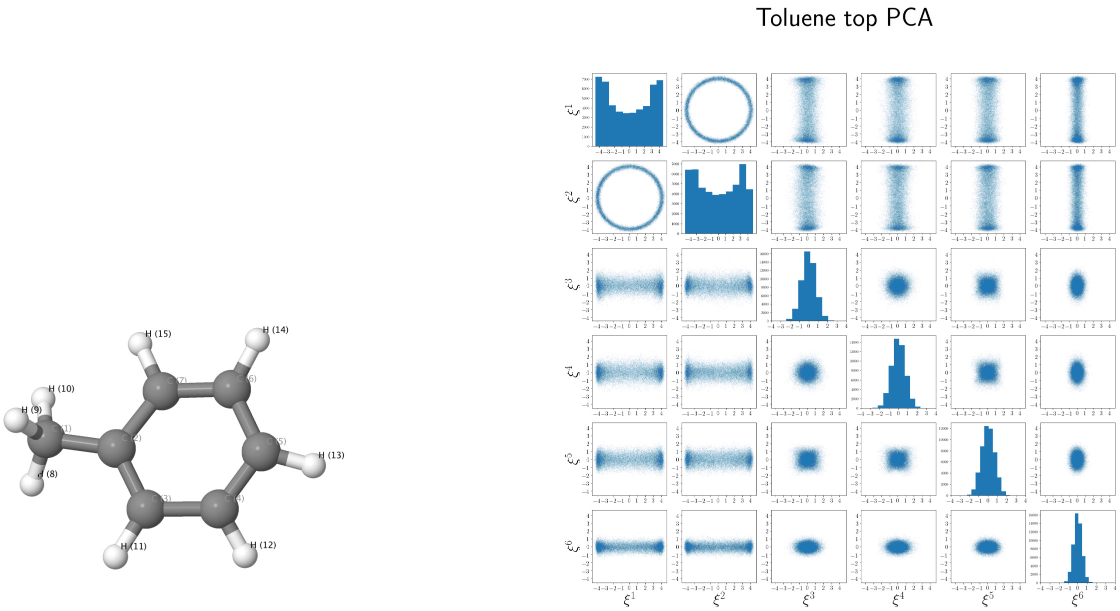
Figure 3.3: Bond diagram and first 6 coordinates of PCA feature space of malonaldehyde.



(a)

(b)

Figure 3.4: Bond diagram and first 6 coordinates of PCA feature space of ethanol.



(a) (b)

Figure 3.5: Bond diagram and first 6 coordinates of PCA feature space of toluene.

3.7.2 Dictionaries for MD data

For molecular dynamics data analyses, our dictionary $\mathcal{G}$ consists of bond torsions g (see Figure 3.1). The bonds represented in the diagrams shown in Figures 3.3, 3.4, 3.5 are stable covalent attractions between two atoms or molecules caused by the sharing of electrons. That is, two atoms that are adjacent in the diagram share electrons, and therefore are stably attracted to each other in the sense that they tend to remain adjacent.

We can associate features of the atomic geometry to bonds. For example, three contiguous bonds in 3.4 - 3.5 define a bond torsion around the central bond. In particular, to each ordered atom 4-tuple (d_1, c_1, c_2, d_2) consisting of two distal and two central points (atoms) in $\mathbb{R}^3$, we associate a torsion $\tau(d_1, c_1, c_2, d_2)$ (where c_1 and c_2 are central, and d_1 and d_2 distal) This torsion is not unique. There is an equivalence

$$\tau(d_1, c_1, c_2, d_2) = \tau(d_1, c_2, c_1, d_2) = \tau(d_2, c_2, c_1, d_1) = \tau(d_2, c_1, c_2, d_1).$$

For example, if $[9, 3, 1, 5]$ is explicitly included in our dictionary, then $[5, 1, 3, 9]$ should not be, since these are in fact the same function. Thus, each set of 4 atoms defines 6 torsions upon ordering, since we have $4! = 24$ ordered 4-tuples, and equivalences of groups of 4. This is understandable geometrically by the fact that a tetrahedron (the shape defined by 4 points) has 6 edges, and therefore 6 torsions. In particular, a torsion $\tau_{d_1c_1c_2d_2}$ is a function of the angles of the triangles $d_1c_1c_2$, $c_1c_2d_2$, $d_1c_1d_2$, and $d_1c_2d_2$. Given four atoms d_1, c_1, c_2, d_2, the torsion - the angle of the planes containing d_1, c_1, c_2 and c_1, c_2, d_2, is

$$\tau(d_1, c_1, c_2, d_2) = \cos^{-1} \tag{3.36}$$

$$\left(\frac{\|((d_2 - \frac{\langle d_2-c_2, c_1-c_2\rangle}{\|c_1-c_2\|_2^2} \odot (c_1 - c_2) - c_2\|_2^2 - \|(d_2 - d_1 + (\frac{\langle d_2-c_2, c_1-c_2\rangle}{\|c_1-c_2\|_2^2} + \frac{\langle d_1-c_1, c_2-c_1\rangle}{\|c_2-c_1\|_2^2} - 1) \odot (c_2 - c_1))\|_2^2 + \|(d_1 - \frac{\langle d_1-c_1, c_2-c_1\rangle}{\|c_2-c_1\|_2^2} \odot (c_2 - c_1)) - c_1\|_2^2}{2\|((d_2 - \frac{\langle d_2-c_2, c_1-c_2\rangle}{\|c_1-c_2\|_2^2} \odot (c_1 - c_2) - c_2\|_2 \|(d_1 - \frac{\langle d_1-c_1, c_2-c_1\rangle}{\|c_2-c_1\|_2^2} \odot (c_2 - c_1)) - c_1\|_2}\right). \tag{3.37}$$

We apply ManifoldLasso to two types of dictionaries. First, we apply it to dictionaries implicitly defined by the bond diagram. That is, we include all torsions where there is a bond (d_1, c_1), (c_1, c_2), and (c_2, d_2). However, since the distribution of electrons is in reality governed by probabilistic quantum mechanics, we also replicate the analysis on a dictionary consisting of all $\binom{6(N_a}{4)}$ torsions in the molecule.

We compute the gradients of the torsions by automatic differentiation (149). Recall that our original featurization of the molecular geometry prior to application of EmbeddingAlg make use of **planar angles**. Given three atoms d_1, c_1, d_2, the planar angle

$$\alpha(d_1, c_1, d_2) = \cos^{-1}(\frac{\|d_1 - c_1\|_2^2 + \|d_2 - c_1\|_2^2 - \|d_2 - d_1\|^2}{2\|d_1 - c_1\|_2 \|d_2 - d_1\|}). \tag{3.38}$$

We are interested in obtaining the gradient of a torsion as a function of the angles. We can in principal write out the torsion as a function of the angles, but one cannot use the obtained gradients directly in ManifoldLasso, since the angular features overparameterize the molecular shape space $\Sigma_3^{N_a}$ (8; 101) of dimension $D' = 3N_a - 7$, and off-manifold gradients are therefore not well-defined. For example, whether one chooses to use angles from triangles defined by (in torsion notation) $\{d_1, c_1, c_2\}$, $\{d_1, c_1, d_2\}$, and $\{d_1, c_2, d_2\}$, or $\{c_1, c_1, d_2\}$ to compute $\tau(d_1, c_1, c_2, d_2)$ has no effect on the value of $\tau(d_1, c_1, c_2, d_2)$, but changes the value of the gradient in the planar angular space. We therefore project the gradients prior to normalization on the tangent bundle of the shape space as it is embedded in $\mathbb{R}^D$.

3.7.3 The shape space

Appropriate treatment of the shape space is essential for running ManifoldLasso. This space is defined in Section 2.5. Here, we show how to obtain the gradient of a torsion τ of a molecular configuration in the tangent bundle of this space at a non-singular point. This material is taken from Addicoat and Collins (8).

We are given a planar angle $\alpha \in \mathbb{R}^{3\binom{N_a}{3}}$, and we are interested in obtaining $\operatorname{grad}_{\Sigma_3^{N_a}} \tau_\alpha(\alpha_i)$, the gradient of the torsion τ w.r.t. the submanifold $\Sigma_3^{N_a} \subset \mathbb{R}^{3\binom{N_a}{3}}$. Here, we have written τ_α to emphasize that τ is a function of the planar angle feature set. We can compute the differential $D_x^\alpha \in \mathbb{R}^{3N_a \times \mathbb{R}^{3\binom{N_a}{3}}}$ consisting of gradients of each planar angle w.r.t. the Euclidean coordinates of the atoms. This map is rank $3N_a - 7$ since codomain of this map is an embedding of the shape space in $R^{3\binom{N_a}{3}}$. A deductive explanation for the rank of W_i is that translation, rotation, and dilation correspond to a total of 7 degrees of freedom. We can also compute the differential $\nabla_x \tau \in \mathbb{R}^{3N_a}$ consisting of gradients of τ w.r.t. the Euclidean coordinates of the atoms. Then

$$\operatorname{grad}_{\Sigma_3^{N_a}} \tau_\alpha(\alpha_i) = (D_x^\alpha(\alpha_i))^\dagger \nabla_x \tau(\alpha_i). \tag{3.39}$$

As an additional but somewhat unrelated preprocessing step, we also apply Principal Component Analysis (PCA) to the angular features matrix $\alpha_{1:n} \in \mathbb{R}^{n \times D}$ prior to running Diffusion Maps. To perform PCA, we use Singular Value Decomposition:

$$\alpha_{1:n} = M \Pi N^T.$$

Denote by P the matrix formed with the first D columns of N; P projects the angular features into a lower dimension space that reduces redundancy while capturing the vast majority of the variability. That is,

$$\xi_i = \alpha_i P, \text{ for } i = 1, \dots n.$$

The gradient of τ with respect to coordinates ξ is thus given by

$$\operatorname{grad}_{\xi} \tau_\alpha(\xi_i) = P^T \operatorname{grad}_{\Sigma_3^{N_a}} \tau_\alpha(\alpha_i).$$

We use $\operatorname{grad}_{\xi} \tau_\alpha(\xi_i)$ as $\nabla_{\xi} \tau_\alpha(\xi_i)$ in ManifoldLasso.

3.8 Experiments: MDS data

In this section, we first demonstrate the workings of ManifoldLasso in a controlled setting by applying it to a simple non-dynamical simulation of a rigidly-rotating ethanol molecule. We then use ManifoldLasso to identify torsions that govern the dynamics of the molecules in Figure 3.1.

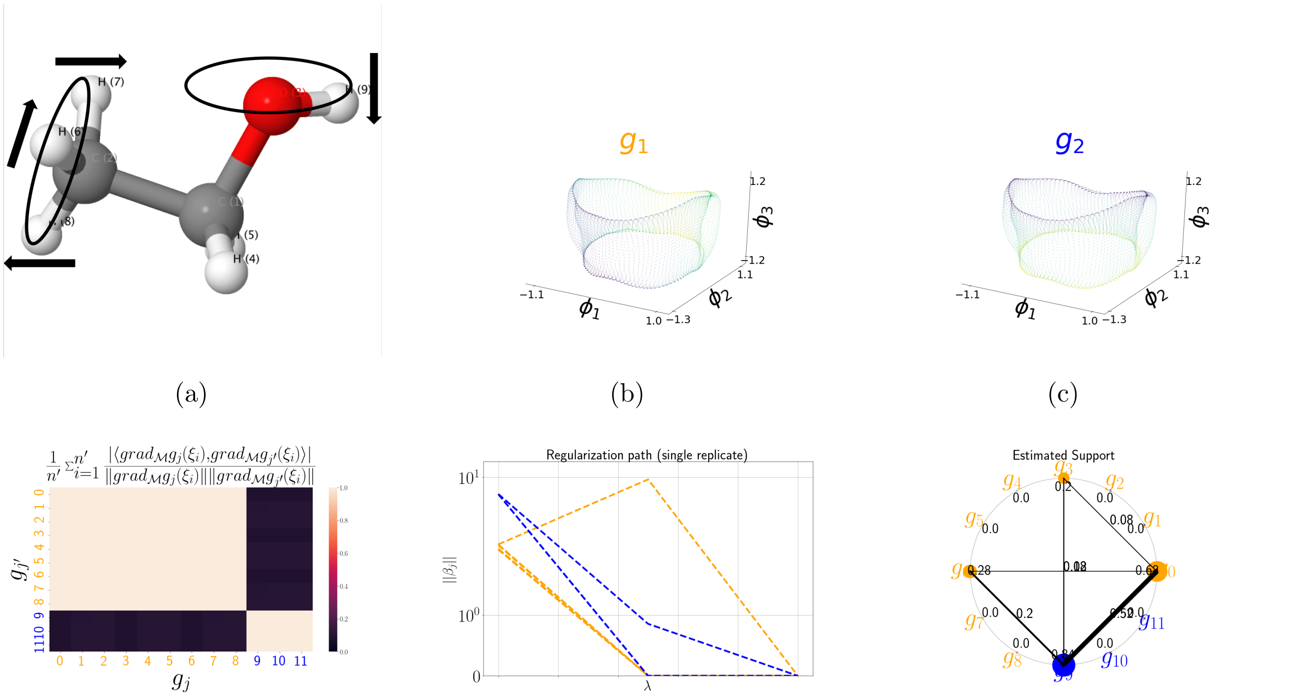
3.8.1 ManifoldLasso *on a Rigid Ethanol skeleton*

We construct an ethanol skeleton composed of the atoms shown in Figure 3.6a. We then sample configurations as we rotate the atoms around the C-C and C-O bonds. In contrast with the MD trajectories, which are simulated according to quantum dynamics, these two angles are distributed uniformly over a grid, and Gaussian noise is added to the position of each atom. We call the resultant dataset **RigidEthanol**. As expected given our two a priori known degrees of freedom, Figures 3.6b and 3.6c show that the estimated manifold is a two-dimensional surface with a torus topology similar to that observed for the MD **Ethanol** in Figure 3.1. In particular, it is parameterized by bond torsions g^1 and g^2

The dictionary consists of the twelve torsions implicitly defined by the bond diagram [2] in Figure 3.6a. All of these torsions circumscribe one of the central C-C and C-O bonds. Counting permutations of peripheral hydrogens, we can see that there are 9 of the former, and 3 of the latter, which we denote by $g^{0:8}$ and $g^{9:11}$ in Figure 3.6d. Hence, any pair $\{g^j, g^{j'}\}$ with $j \in \{0 : 8\}, j' \in \{9 : 11\}$ is an equally correct coordinate system for this manifold. This is shown in Figure 3.6d by the incoherences $\mu_{jj'}$, i.e. mean pairwise cosines of the dictionary functions. Comparing the row and column labels of Figure 3.6d with Figure 3.6a shows that the collinearities of gradients clearly cluster by central bond. Thus, we expect ManifoldLasso to recover one torsion from each group. Indeed, in the regularization path of an individual replicate of ManifoldLasso shown in Figure 3.6e, collinear torsions are

[2] These are all 4-tuples of atoms connected by a path in the figure, modulo the natural equivalence relation on torsions previously described.

killed off, and a representative torsion is selected from each group. Finally, Figure 3.6f shows that ManifoldLasso selects such orthogonal pairs in 18 out of 25 random replicates of the n' points.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.6: Results of ManifoldLasso for **RigidEthanol**. Figure 3.6a shows the simplified dynamics of our rigid molecular simulation. Atoms in the rigid ethanol skeleton are articulated around the C-O and C-C bonds by a torus of rotations. Figure 3.6b shows the learned torus, colored by C-C torsion g^1 from Figure 3.1. Figure 3.6c shows the same torus, colored by the C-O torsion g^2 from Figure 3.1. Figure 3.6d displays the incoherences, i.e. pairwise collinearities of dictionary gradients; C-C torsions are in orange, C-O torsions in blue. Figure 3.6e shows regularization paths $\|\beta_{.j.}\|_F$ vs. λ for a single replicate. The chord diagram in Figure 3.6f represents the frequency of selecting each pair of torsions in replicate experiments. The listed frequencies with which individual torsions are selected are given by the sizes of the perimeter dots corresponding to each dictionary element, while the frequencies with which pairs of torsions are selected are given by the line widths connecting the dots. Frequencies are also given by the numbers next to the respective graphical indicators.

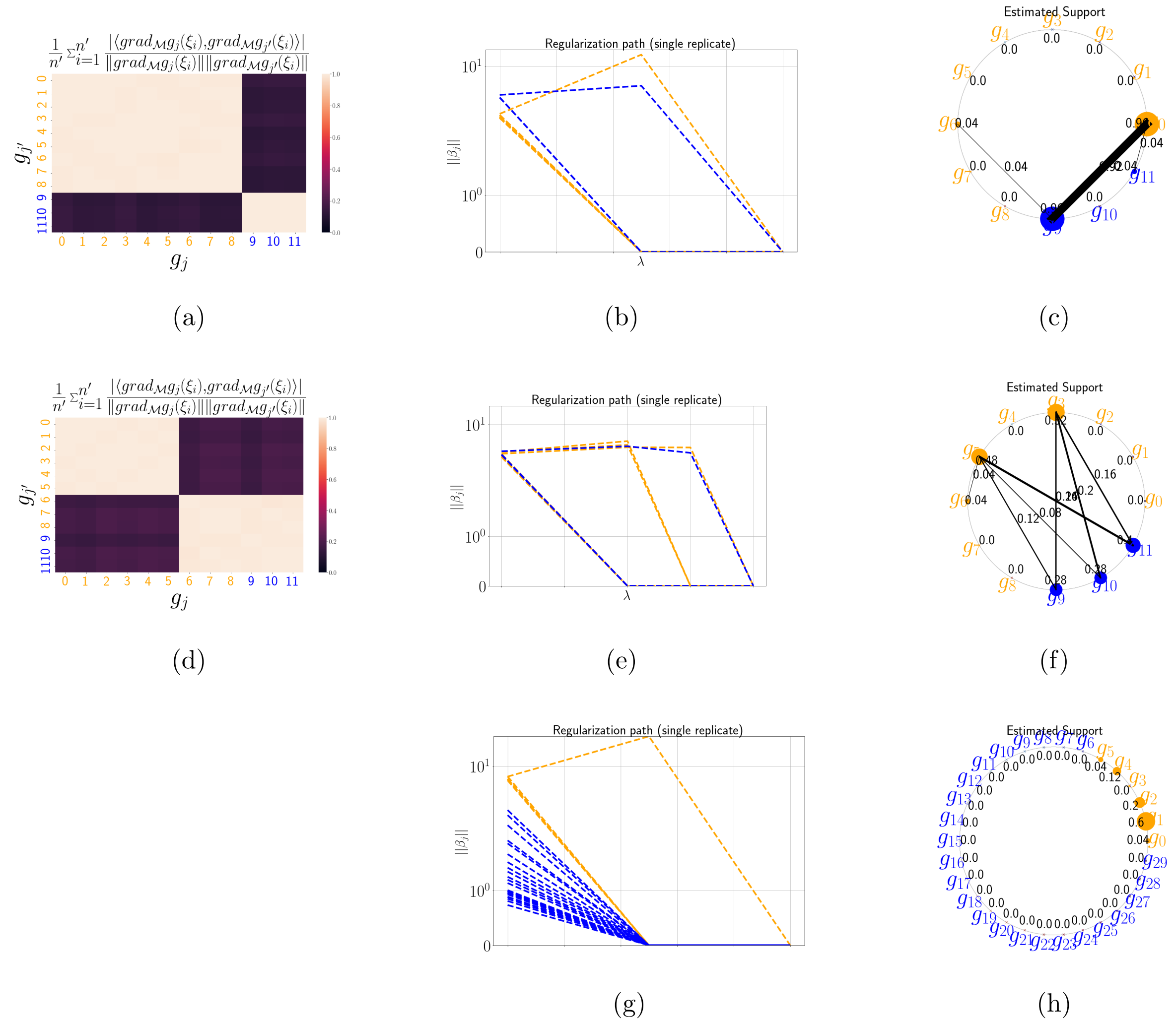
3.8.2 Molecular Dynamics results

We first show that ManifoldLasso can distinguish groups that correspond to the chemical bonds in Figure 3.1, as would typically be done by a scientist using prior domain knowledge. Next, we repeat the analysis with no prior knowledge, including all distinct 4-tuples of atoms in the dictionary. Note that as the data ξ does not lie exactly on $\mathcal{M}$, the values of $\operatorname{grad}_{T_i\mathcal{M}} g^j$ will necessarily be noisy as well.

3.8.3 Dictionaries based on bond diagrams

Bond diagrams such as the ones in Figure 3.1 are based on a priori information about molecular structure garnered from historical work. Building a dictionary based on this structure is akin to many other methods in the field (110; 194). As in the case of **RigidEthanol**, our dictionaries consist of all equivalence classes of 4-tuples of atoms implicitly defined by bond diagrams, and the incoherence plots for **Ethanol** and **Malonaldehyde** in Figures 3.7a and 3.7d show two groups of highly dependent torsions, corresponding to the two bonds between heavy atoms in the molecules. Therefore, success means recovering a pair of incoherent torsions out of these dictionaries. For **Toluene**, the manifold dimension is $d = 1$ and success means recovering one of the 6 torsions associated with the peripheral methyl group bond. For this molecule, there are also $p - 6 = 24$ torsions that do not explain the data manifold. We apply ManifoldLasso with these dictionaries to the embeddings shown in Figure 3.1.

As Figure 3.7 shows, ManifoldLasso is always able to identify torsions corresponding to the expected labelled bonds. Figures 3.7b, 3.7e, and 3.7g show regularization paths for single replicates of ManifoldLasso, and Figures 3.7c, 3.7f and 3.7h show frequencies of support recovery of sets of size d over $w = 25$ replicates. ManifoldLasso finds that the toroidal **Ethanol** manifold is explained by pairs of torsions from the C-O and C-C bonds, while **Malonaldehyde** is explained by one of each of the two central bonds. **Toluene** is explained by the torsion of the peripheral methyl group. These agree with our domain-expert validated parameterizations from Figure 3.1.



(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 3.7: Results for MD data with a priori dictionaries given by the bond diagrams in Figure 3.1. The three rows correspond to **Ethanol**, **Malonaldehyde**, and **Toluene**, respectively. Figures 3.7a and 3.7d display pairwise collinearities of dictionary gradients, colored by bond as in Figure 3.1. Toluene, a $1-d$ manifold, has trivial cosines, and so these are not shown. Figures 3.7b, 3.7e, and 3.7g show overall regularization paths of $\|\beta_{.j.}\|_F$ for single replicates. Figures 3.7c, 3.7f, and 3.7h show chord diagrams displaying frequency of support recovery of sets of size d for 25 replicates. As for **RigidEthanol**, two-dimensional support recovery frequency is denoted by chord width, and one-dimensional support recovery frequency is denoted by size of perimeter dot. Note that ’blue’ in toluene corresponds to torsions in the benzene ring.

Coordinate-association We show the association of individual embedding coordinates to dictionary functions in **Ethanol** and **Malonaldehyde**. In general, manifold coordinates may not have individual meaning, so it will not always be possible to find a good explanation for a single ϕ^k. However, in contrast to **Malonaldehyde**, but similar to **SwissRoll**, Figure 3.8 shows that **Ethanol** has a distinct association of embedding coordinates with dictionary functions. In particular, ϕ^3 is associated with torsions from different groups as those associated with ϕ^1 and ϕ^2. This is clearly evident in Figure 3.1. In **Malonaldehyde**, there is no clear association with individual embedding coordinates. Note that this would also be true for **Toluene**, as Figure 3.1 clearly shows a circular manifold symmetric in ϕ^1 and ϕ^2.

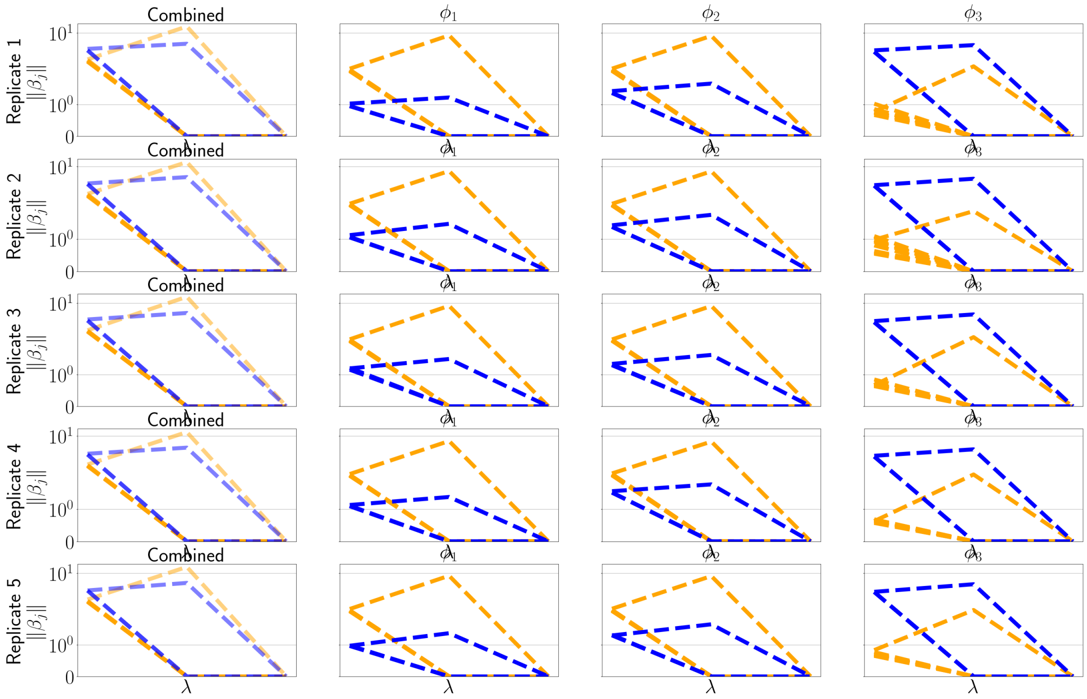


Figure 3.8: Combined and coordinate-specific regularization paths in five replicates of ManifoldLasso for **Ethanol** with dictionary given by the bond diagram. The blue torsion associates with ϕ^3, and the orange with $\phi^{1,2}$.

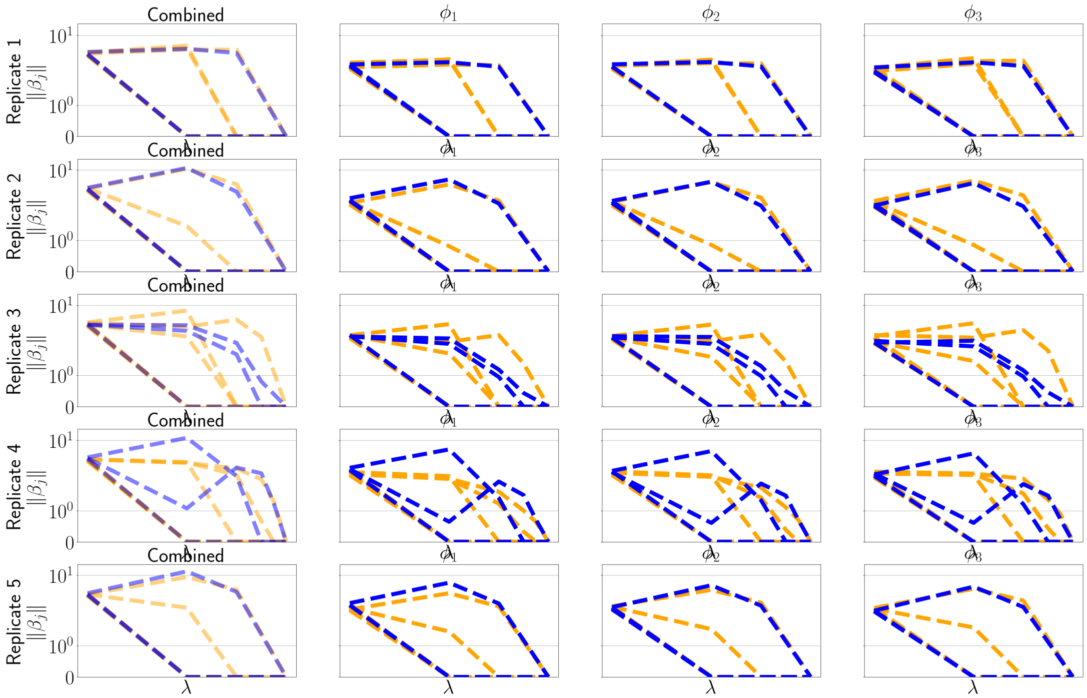


Figure 3.9: Combined and coordinate-specific regularization paths in five replicates of ManifoldLasso for **Malonaldehyde** with dictionary given by the bond diagram. There is no clear association of embedding coordinates and covariates.

3.8.4 Results from full dictionary

Actual interatomic interactions are often more complex than exhibited in a bond diagram; thus, the analyzed molecules have more interactions than those represented in Figures and 3.1a-3.1c, and potentially more interesting bond torsions. Indeed, a motivating application of quantum dynamics simulations is to uncover molecular behavior that is not encapsulated by the simplified bond diagrams. This motivates us to test ManifoldLasso in the case when the dictionary consists off all possible torsions, i.e. all $\binom{N_a}{4}$ 4-tuples modulo equivalence.

For **Ethanol** and **Malonaldehyde** we obtain $p = 756$ torsions. [3] Such a large p is challenging for l_1 regularized estimation, due to the bias mentioned in Section 6.5.1 for large λ. Moreover, examining Figures 3.10a and 3.10e, we see that, besides the two groups of collinear torsions in the previous dictionary, there are other torsions, about a fourth of the 756, that are coherent with both groups. While we do not necessarily expect ManifoldLasso to succeed, or to be used in such a way in practice, this experiment will inform us on the robustness of ManifoldLasso in a situation that is challenging for any type of sparsity inducing regularization.

The results of ManifoldLasso with the full dictionary for **Ethanol** and **Malonaldehyde** are displayed in Figure 3.10. For consistency between replications, we choose a priori the ground truth to be represented by torsions $g_{74,176}$ and $g_{0,8}$, which are representative torsions for **Ethanol**, respectively for **Malonaldehyde**, depicted in Figure 3.1. We can evaluate the selected $d = 2$ functions for coherence with this ground truth. In this most challenging setting, ManifoldLasso identifies supports with mean incoherences with the true support of $.68 \pm .32$ and $.95 \pm .1$ for **Ethanol** and for **Malonaldehyde**, respectively. This is apparent from comparing selected torsions in Figures 3.10c and 3.10g with their collinearities in Figures 3.10d and 3.10h. Thus, we can see that ManifoldLasso performs very well on **Malonaldehyde** but more poorly on **Ethanol**.

In the latter figures, collinearities of the selected supports with example functions from the

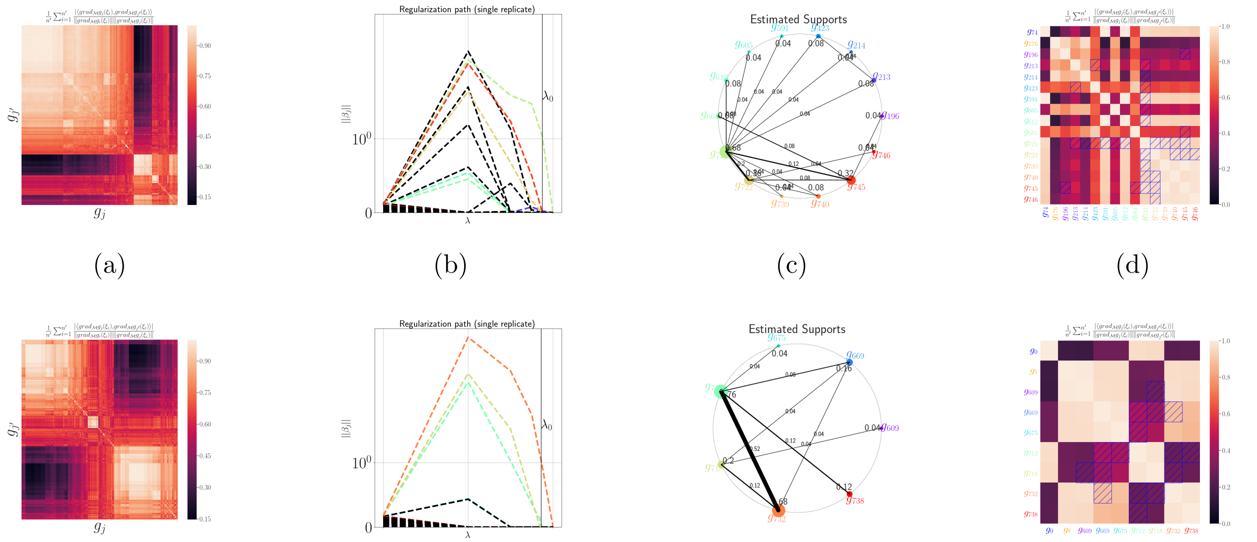
[3] We do not analyze **Toluene**, because for $d = 1$ the solution is available analytically, making this example somewhat trivial.

representative true support are also plotted. We can see that the selected support functions are often strongly coherent with the ground truth functions, both when the selected support is almost orthogonal, and when the selected support functions are not. In the latter case, both selected support functions are strongly coherent with only one of the ground truth functions. Note that when both selected functions are more coherent with a single element of the true support, we use the pairwise coherences with higher mean. The results are visualized in Figure 3.10, which shows the embeddings colored by the selected torsions. The identities of the selected torsions can be compared with the bond diagrams in Section 3.7.2. There is a clear visual correspondence between coherences between torsions and their colorings of the manifolds learned from **Ethanol** and **Malonaldehyde**; thus, when orthogonal pairs are selected, we capture information that would otherwise necessarily be obtained visually from the embeddings. However, the **Malonaldehyde** plots also demonstrate that even for this simple manifold, associating manifold coordinates to dictionary functions by visual inspection is delicate work. From a chemistry perspective, orthogonal recovered torsions generally flank pairs of hydrogens of which each is attached to one of the central atoms in the putatively true bonds. Thus, it makes sense that these peripheral torsions could geometrically describe the same motion as the putative true support.

	$\bar{\mu}$	σ_μ	$\bar{\kappa_S}$	σ_{κ_S}	$\gamma_{\max}$	$\sigma_{\gamma_{\max}}$	$\bar{\min_{i=1}^n \min_{j\in S} \|\|x_{ij}\|\|}$	$\sigma_{\min_{i=1}^n \min_{j\in S} \|\|x_{ij}\|\|}$
Ethanol (a priori)	∼1.0	8.348332e-08	10.029410	9.815921	44.110696	1.825407	0.970851	0.530601
Malonaldehyde (a priori)	∼1.0	9.752936e-08	2.220002	0.771986	26.189684	0.477759	2.709132	0.464913
Toluene (a priori)					15.576112	0.407799	1.449570	0.914012
Ethanol (agnostic)	∼1.0	4.801062e-11	4.138372	2.113210	57.300602	1.360244	1.932001	1.605302
Malonaldehyde (agnostic)	∼1.0	2.016285e-09	2.204895	0.589812	66.019168	1.044451	3.955157	0.853646

Table 3.2: Mean and standard deviation of theoretical quantities across replicates

For all quantum MD experiments, we examine the support recovery condition Theorem37. In practice, we do not know theoretical quantities like μ, $\gamma_{\max}$, κ_S, and $\min_{i=1}^n \min_{j'\in S} \|X_{ij.}\|_2$ since we do not have access to S. However, we are able to calculate these quantities using the putative true support. The quantities in Table 3.2 indicate that the assumptions of our support recovery guarantees are not satisfied. We first note that Figures 3.7a and 3.7d show that even without foreknowledge of a unique true support, the incoherence parameter μ must be quite close to 1, since it is a maximum over set of cosines whose mean is plotted. The high values of the incoherence parameter μ and otherwise unfavorable empirical support recovery parameters listed in the table indicate that we cannot expect a unique recovery. However, ManifoldLasso is still successful in obtaining representative torsions from the desired bonds in Figure 3.1. The similarity between the results on this real data, with more challenging noise and variable sampling density, and the result on the synthetic **RigidEthanol** are witness to the robustness of the ManifoldLasso method. When ManifoldLasso fails to select orthogonal functions, for example due to the documented support recovery instability and bias at high values of λ for Lasso methods in general (132; 88; 90), we consider a two-stage variable selection procedure in which a secondary variable selection step is applied after initial pruning, as in Hesterberg et al. (88). This approach is described in Chapter 5.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 3.10: Results for MD data with full dictionaries consisting of all possible torsions. The top and bottom rows show results for **Ethanol** and **Malonaldehyde**, respectively. Figures 3.10a and 3.10e show mean cosine collinearity of dictionary gradients ordered by heirarchical clustering. Figures 3.10b and 3.10f show examples of regularization paths for single replicates that select relatively orthogonal functions. The tuning parameter at which $|S| = d$ is indicated as λ_0. Functions are colored if they are selected in any replicate. Figure 3.10c and 3.10g shows support recoveries given by ManifoldLasso over different replicates. Figure 3.10d and 3.10h and shows mean cosine collinearity of selected supports. $g_{74,176}$ and $g_{0,8}$ are representative torsions from the true support, while the others are selected in any replicate. Pairs that are selected in any replicate are marked with a blue box.

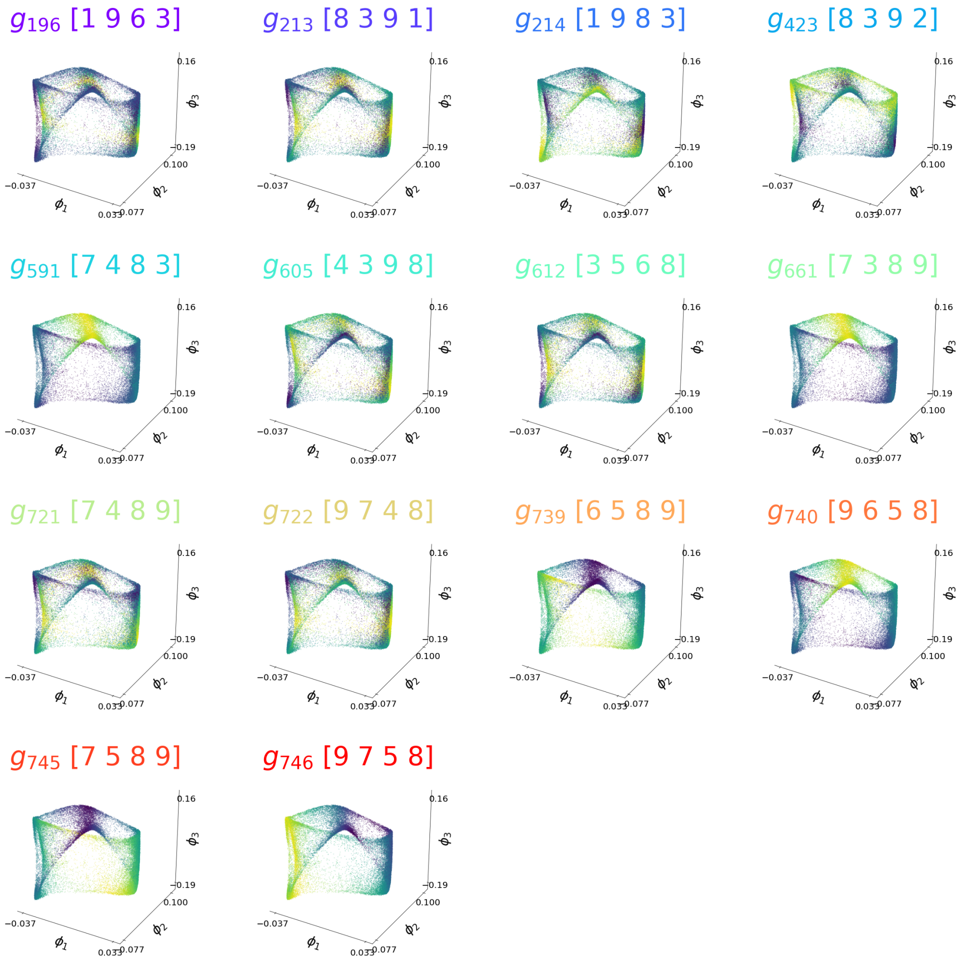


Figure 3.11: **Ethanol** support estimated using ManifoldLasso with full dictionary. Colors should be compared with Figure 3.10.

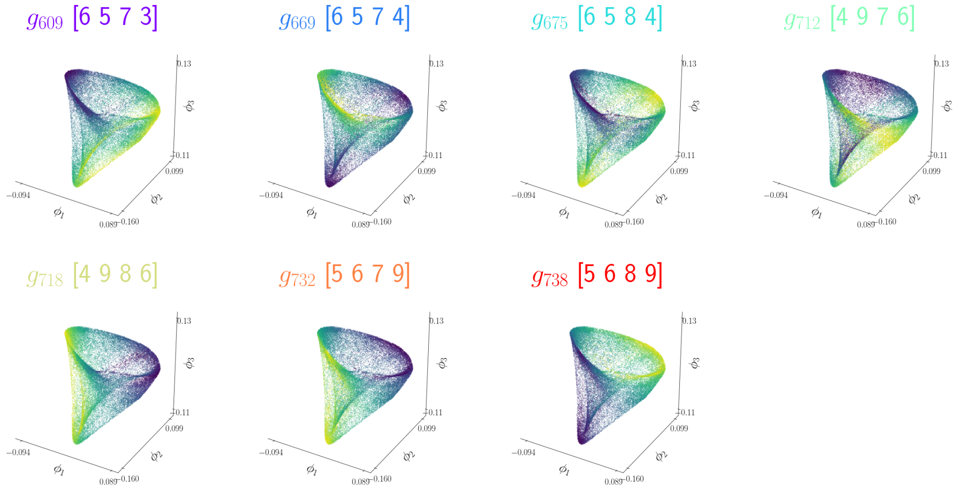


Figure 3.12: **Malonaldehyde** support using using ManifoldLasso with full dictionary. Colors should be compared with Figure 3.10.

3.9 Discussion

The ManifoldLasso algorithm presented here can be extended in several interesting ways, and indeed, Chapters 4, 5, and 6 all extend the methods here in one way or another. In Chapter 4, we directly explain the tangent subspace of $\mathcal{M}$, independently of any embedding In Chapter 5, we consider a related problem that clarifies the heuristic tendency of ManifoldLasso to select orthogonal coordinates, defines an empirical success criteria more specific than rank, and helps resolve violated support recovery conditions. In Chapter 6, we jointly study gradient and tangent space estimation. For these reasons, we defer most of our discussion of related and future work to later in the thesis.

There are however several differentiating features of this chapter. Gradient estimation on manifolds is typically derived from the perspective of local linear regression and tangent space estimation (140; 17). However, as in Luo et al. (126), we make explicit the logarithmic map by estimating and projecting upon the tangent space of $\phi(\mathcal{M})$, and our estimates of this tangent space are made using the pushforward metric of Perraul-Joncas and Meila (152).

The symbolic regression methods of Brunton et al. (35), Rudy et al. (162), and Champion et al. (39) for estimating governing laws of dynamical systems are especially similar to ManifoldLasso. These methods use sparse regression with respect to a dictionary to learn partial derivatives of the state of physical systems w.r.t. time. Their goal is to identify the functional equations of non-linear dynamical systems by regressing the time derivatives of the state variables on a subset of functions in the dictionary selected using a sparsity inducing penalty. In contrast, the response variable in our regression are partial derivatives of coordinate functions of the embedding space.

A similar group lasso approach to regressing vector fields was used in Haufe et al. (82). Our method is differentiated from this approach by several features. First, our gradients are w.r.t. a manifold. Second, we apply the vector field alignment to the problem of interpreting coordinates output by an unsupervised learning algorithm. Finally, our vectors are gradients of functions, and so our approach is actually solving the original functional support recovery problem.

Chapter 4
TANGENT SPACE LASSO

This chapter introduces a simpler alternative to the Manifold Lasso for interpretable manifold learning. This chapter shows how to adapt the Manifold Lasso approach for learning non-zero partial derivatives of one set of response functions jointly with respect to another set of covariate functions to the task of parameterizating a tangent space directly. This approach, which we call Tangent Space Lasso, shares a similar algorithmic structure and motivating use case in scientific data analysis, but rather than regressing individual embedding coordinates, it applies a multidimensional variant to explain the tangent space itself. This new approach does not make use of an embedding, and therefore should itself be thought of as an embedding algorithm. We review the conditions for the existence of such parameterizations in function space and for successful recovery from finite samples, and give results on small molecule MDS.

4.1 Introduction

The theoretical condition for successful support recovery of the Manifold Lasso algorithm introduced in Chapter 3 that selected elements from a dictionary of functions have gradient fields that are everywhere full rank does not refer to a learned embedding. This raises the question of whether we can use a convex regularized method to recover manifold functional support first embedding the data using a manifold learning algorithm. In this chapter, we show that this is indeed the case. We introduce a simplified version of the Manifold Lasso algorithm called Tangent Space Lasso that explains entire subspaces rather than individual gradients, and can therefore identify manifold parameterizations without the use of embedding coordinates.

This method shares many features with the Manifold Lasso algorithm. The gradients of a dictionary of domain related smooth functions are projected onto the estimated tangent spaces of the data manifold before being used as covariates in a regression. As before, this regression is highly overdetermined w.r.t. the manifold dimension, and so we apply a group lasso regularizer to simultaneously zero-out entire gradient fields of individual dictionary functions. However, the response variables in this regression are no longer the gradient field of an embedding coordinate function, but rather the basis vectors of the tangent spaces themselves. As long as the dictionary is constructed from functions that have meaning in the domain of the problem, this learned embedding is still interpretable by definition. The learned parameterization still has a functional form, and therefore can be used to compare embeddings from different sources, derive out-of-sample extensions, and to interrogate mechanistic properties of the analyzed system. This functional form constrasts with methods such as the Nystrom extension for Diffusion Maps that enable out-of-sample embedding (23).

Finding parameterizations without use of an embedding has several advantages. It streamlines the learning procedure and removes a mathematically unnecessary step. It may not always be easy to find a good embedding (43), and gradients of embedding coordinates output by algorithms like Diffusion Maps must be estimated, and so are a potential source of noise. Fourth, as we have seen in Chapter 3, embedding coordinates may privilege certain directions in the tangent space, and so by considering only tangent spaces, we remove some amount of redundancy. Despite this, the special relevance of non-parametric embedding algorithms will be further examined in Chapter 6.

Section 4.2 gives formuates the support recovery problem, and Section 4.3 presents the TSLasso algorithm. Section 4.4 reviews conditions for unique recovery and selection consistency. Section 4.5 shows experimental results on toy data and molecular dynamics datasets. Section 4.6 examines related work and Section 4.7 discusses interesting features and ideas for the future.

4.2 Problem and motivation

The Tangent Space Lasso directly learns a sparse parameterization of a d dimensional smooth manifold $\mathcal{M}$ from a dictionary of smooth analytically computable functions $\mathcal{G} = \{g^j, j \in [p]\}$ of an open set U containing $\mathcal{M}$ to $\mathbb{R}$. The mathematical context is mostly shared with Chapter 3.

Problem Statement Suppose data $\mathcal{D}$ are sampled from a d-dimensional smooth submanifold $\mathcal{M}$ embedded in the Euclidean space $\mathbb{R}^D$, where typically $D \gg d$. Assume that the intrinsic dimension d is known. In contrast to Chapter 3, in this Chapter by smooth we mean at least C^1.

Our goal is to identify a mapping $g^S : \{g^j\}_{j \in S}, S \subseteq [p], |S| = d$ such that at every point ξ, g^S is a diffeomorphism from an open neighborhood $U \subset \mathcal{M}$ of ξ to $g^S(U) \subset \mathbb{R}^{|S|}$.

Tangent space estimation At each data point $\xi_i \in \mathcal{M}$, TSLasso requires estimation of a $D \times d$ tangent basis $T_i^{\mathcal{M}}$ of a tangent space $\mathcal{T}_{\xi_i}\mathcal{M}$. We use the Weighted Local Principal Component Analysis wlpca algorithm given in Section 2.3.

4.3 TSLasso*Algorithm*

The idea of the TSLasso algorithm is to express the bases $T_i^{\mathcal{M}}$ of the data manifold tangent spaces $\mathcal{T}_{\xi_i}\mathcal{M}$ as sparse combinations of dictionary function gradient vector fields. This simplifies the non-linear problem of selecting a best functional approximation to $\mathcal{M}$ to the linear problem of selecting best local linear approximations. As in Chapter 3, our goal is to obtain S such that g^S is a local diffeomorphism, that is, $Dg^S(\xi_i)$ is rank d.

Let $X \in \mathbb{R}^{n \times d \times p}$ be the array formed by stacking of $\operatorname{grad}_{T_i^{\mathcal{M}}} g^j(\xi_i)$ as in Chapter 3, and let $X_{iS.}$ refer to the $R^{n \times d \times d}$ array with $j \in S \subseteq [p]$. If $Dg^S(\xi_i)$ is rank d, then there exists a $d \times d$ matrix $\beta_{iS.}$ such that

$$\beta_{iS.} = X_{iS.}^{-1}, \ \forall\, i \in [n]. \tag{4.1}$$

However, in the general case, $p > d$, and so given an array $\beta \in \mathbb{R}^{n \times p \times d}$, the decomposition

$$I_d = X_{i..}\beta_{i..}, \; \forall \; \xi_i. \tag{4.2}$$

holds for many different $\beta_{i..}$. By adding a joint sparsity constraint, we are able to identify a minimal subset of interest S with rank d.

4.3.1 Loss Function

We adapt the ManifoldLasso method to induce joint sparsity within matrices $\beta_{.j.}$. The objective function we propose is

$$J_{TS,\lambda}(X, \beta) := \frac{1}{2}\sum_{i=1}^{n} \|I_d - X_{i..}\beta_{i..}\|_2^2 + \frac{\lambda}{\sqrt{dn}}\sum_{j=1}^{p}\|\beta_{.j.}\|_F. \tag{4.3}$$

Compared with Objective 3.19 in the previous chapter, we have replaced gradients of embedding coordinates with orthonormal basis vectors of the tangent spaces. This captures many of the important properties of the Manifold Lasso objective, but in a simplified form. The group lasso penalty induces sparsity between dictionary functions and jointly across data points.

The main difference between Objectives 4.3 and 3.19 is that the former replaces the gradients of the embedding functions with the bases I_d of the tangent spaces $\mathcal{T}_{\xi_i}\mathcal{M}$. In Chapter 3 we showed that, given a point $\xi \in \mathcal{M}$, the choice of $T_\xi^{\mathcal{M}}$ that rotates gradients of both dictionary functions and embedding coordinates equivalently does not change the loss. Now, we show that the $J_{TS,\lambda}$ is invariant to the choice of basis of $\operatorname{grad}_{T_i^{\mathcal{M}}} g^j(\xi)$ only.

Lemma 39. *Given arrays $X \in \mathbb{R}^{n \times d \times p}$, $\beta \in \mathbb{R}^{n \times p \times d}$ and $X^i \in \mathbb{R}^{n \times d \times p}$ where $X^i_{i'} = X_{i'}$ if $i' \neq i, X^i_{i'} = UX_{i'}$ if $i' = i$ and U is a $d \times d$ unitary matrix, then*

$$J_{ts,\lambda}(X, \beta) = J_{ts,\lambda}(X^i, \beta^i) \tag{4.4}$$

where $\beta^i_{i'} = \beta_{i'}$ if $i' \neq i, \beta^i = \beta_{i'}U^T$ if $i' = i$.

Proof. Without loss of generality, let $i = 1$. Then, dropping constants,

$$J_{TS,\lambda}(X^1, \beta^1) = \sum_{i=1}^{n} \|I_d - X_i \beta^1_{i..}\|^2 + \lambda \sum_{j=1}^{p} \|\beta^1_{.j.}\|_F \tag{4.5}$$

$$= \|I_d - X^1_{1..}\beta^1_{1..}\|_F^2 + \sum_{i=2}^{n} \|I_d - X_{i..}\beta^1_{i..}\|_F^2 + \lambda \sum_{j=1}^{p} \|\beta^1_{.j.}\|_F \tag{4.6}$$

$$= \|I_d - UX_{1..}\beta_{1..}U^T\|_F^2 + \sum_{i=2}^{n} \|I_d - X_{i..}\beta^i_i\|_F^2 + \lambda \sum_{j=1}^{p} \|\beta^1_{.j.}\|_F \tag{4.7}$$

$$= \|U(I_d - X_{1..}\beta_{1..})U^T\|_F^2 + \sum_{i=2}^{n} \|I_d - X_{i..}\beta^i_i\|_F^2 + \lambda \sum_{j=1}^{p} \|\beta^1_{.j.}\|_F \tag{4.8}$$

$$= \|I_d - X_{1..}\beta_{1..}\|_F^2 + \sum_{i=2}^{n} \|I_d - X_{i..}\beta^i_i\|_F^2 + \lambda \sum_{j=1}^{p} \|\beta^1_{.j.}\|_F \tag{4.9}$$

$$= \sum_{i=1}^{n} \|I_d - X_{i..}\beta^i_{i..}\|_F^2 + \lambda \sum_{j=1}^{p} (\sum_{i=1}^{n} \|\beta^1_{ij.}\|_2^2) \tag{4.10}$$

$$= \sum_{i=1}^{n} \|I_d - X_{i..}\beta^i_{i..}\|_F^2 + \lambda \sum_{j=1}^{p} (\sum_{i=2}^{n} \|\beta_{ij.}\|_2^2 + \|\beta_{1j.}U^T\|_2^2)^{1/2} \tag{4.11}$$

$$= \sum_{i=1}^{n} \|I_d - X_{i..}\beta^i_{i..}\|_F^2 + \lambda \sum_{j=1}^{p} (\sum_{i=2}^{n} \|\beta_{ij.}\|_2^2 + \|\beta_{1j.}\|_2^2)^{1/2} \tag{4.12}$$

$$= \sum_{i=1}^{n} \|I_d - X_{i..}\beta^i_{i..}\|_F^2 + \lambda \sum_{j=1}^{p} \|\beta_{.j.}\|_F \tag{4.13}$$

$$= J_{TS,\lambda}(X, \beta). \tag{4.14}$$

□

In our context, this proposition applies to $X = [\text{grad}_{T_i^{\mathcal{M}}}\, g^j(\xi_i)\ s.t.\ i \in [n], j \in [p]]$ in tangent bases $T_i^{\mathcal{M}}$, and U represents an alternative choice of basis.

Proposition 40. *Suppose we are given arrays $X \in \mathbb{R}^{n \times d \times p}$ and $X^i \in \mathbb{R}^{n \times d \times p}$ where $X^i_{i'..} = X_{i'..}$ if $i' \neq i, X^i_{i'} = UX_{i'..}$ if $i' = i$. Let $\hat{\beta} = \arg\min_\beta J_{ts,\lambda}(X, \beta), \hat{\beta}^i = \arg\min_\beta J_{ts,\lambda}(X^i, \beta),$*

and suppose $\hat{\beta}^i$ is unique. Then

$$\hat{\beta}^i_{i'..} = \hat{\beta}_{i'..} \textit{ if } i' \neq i \tag{4.15}$$

$$\hat{\beta}^i_{i'..} = \hat{\beta}_{i'..} U^T \textit{ if } i' = i. \tag{4.16}$$

Proof. Suppose that there exists a β' such that $J_{TS,\lambda}(X^i, \beta') < J_{ts,\lambda}(X^i, \beta^i)$. Then, by Lemma 39, there exists a β^{-i} with $\beta_i^{-i} = \beta_i U^T$ such that $J_{ts,\lambda}(X, \beta^{-i}) < J_{TS,\lambda}(X, \hat{\beta})$, which contradicts our assumption that $\hat{\beta}$ minimizes $J_{TS,\lambda}(X, \hat{\beta})$. □

Finally, we examine the connection to the recovered support. Let $S(\beta) = j \in [p] \; s.t. \|\beta_{.j.}\|_2 > 0$.

Proposition 41. *Suppose we are given X, X^i and resulting β, β^i as in Proposition 40. Then $S(\hat{\beta}) = S(\hat{\beta}^i)$.*

Proof. Recall that the only difference between β and β_i is that $\hat{\beta}_{i..} = \hat{\beta}^i_{i..} U^T$. Thus, rows of $\beta_{i..}$ that are zero will remain so after rotation, and similarly for those that are non-zero. □

Thus, the selected support S is independent of the basis chosen for each tangent space.

4.3.2 TSLasso *Algorithm*

These propositions show that $J_{TS,\lambda}$ is suitable for manifold support recovery in a similar manner to the objective from Chapter 3. We can therefore transform the non-linear manifold support recovery problem into a collection of sparse linear problems in which we express coordinates of individual tangent spaces as linear combinations of gradients of functions from our dictionary. Tangent spaces at each point are estimated in step 8, enabling utilizing gradients of dictionary functions in $\mathcal{T}_{\xi_i}\mathcal{M}$ by projecting the gradient $\nabla_\xi g^j(\xi_i) \in \mathbb{R}^D$ on to estimated tangent spaces $T_i^{\mathcal{M}}$ as described in Chapter 3. We then input these gradients into Objective 4.3 and minimize to solve for the support.

Algorithm 5 TSLasso(Dataset $\mathcal{D}$, dictionary $\mathcal{G}$, intrinsic dimension d, kernel bandwidth ϵ, neighborhood cutoff size r, regularization parameter λ)

```
Construct $\mathcal{N}_i$ for $i = 1 : n$; $i' \in \mathcal{N}_i$ iff $\|\xi_{i'} - \xi_i\| \leq r$, and construct local data matrices $\Xi_{1:n}$
Construct kernel matrix $K \leftarrow \exp(\frac{{\|\Xi_i - 1_{k_i}\xi_i\|_2}^2}{\epsilon})$ for $\epsilon = \frac{r}{3}$
for $j = 1, 2, \ldots p$ do
    Compute $\nabla_\xi g^j(\xi_i)$ for $i = 1, \ldots n$
    Compute $\zeta_j^2$ by (3.21) and normalize $\nabla_\xi g^j(\xi_i) \leftarrow (1/\zeta_j)\nabla_\xi g^j(\xi_i)$ for $i = 1, \ldots n$
end for
for $i = 1, 2, \ldots n$ do
    Compute basis $T_i^{\mathcal{M}} \leftarrow$WLPCA$(\Xi_i, K_{i,\mathcal{N}_i}, d)$
    Project $X_i \leftarrow (T_i^{\mathcal{M}})^T \nabla_\xi g^{1:p} \in \mathbb{R}^{d \times p}$
end for
$\beta \leftarrow$ GroupLasso$(I_{d,1:n}, X_{1:n}, \lambda\sqrt{dn})$
Output $S = \operatorname{supp}\beta$
```

Normalization Normalization details are shared with Chapter 3.

Computation The major difference from ManifoldLasso, computationally, is that we no longer need to compute an embedding. Identifying local datasets Ξ_i is $O(Dnn')$, where $n' = |I|$, the number of assayed points. This is less expensive than the time need to construct the full neighborhood graph for an embedding. For each Group Lasso iteration, the compute time is $O({n'}^2 pd^3)$.

Tuning For TSLasso, the theoretical maximum λ value, is

$$\lambda_{\max} = \max_j (\sum_{i=1}^{n} \sum_{k=1}^{d} (\operatorname{grad}_{T_i^{\mathcal{M}}} g^j(\xi_i))^T e^k)^{1/2}. \tag{4.17}$$

Other details are shared with Section 3.4.4.

Connection to isometry As mentioned in Definition 22, a map ϕ between two d dimesional Riemannian manifolds $(M, \mathbf{g})$ and $(N, \mathbf{h})$ is an isometry if, for all points $\xi \in \mathcal{M}$

$$\langle u, v\rangle_{\mathbf{g}(\xi)} = \langle D\phi u, D\phi v\rangle_{\mathbf{h}(\phi(\xi))}.$$

When $\mathbf{g} = \mathbf{id}$ and $\mathbf{h} = \mathbf{id}$ are unitary, this is equivalent to $D\phi$ being unitary. If ϕ is a mapping into $\mathbb{R}^d$, then this means that $D\phi(\xi)$ consists of gradients $\operatorname{grad}_{T_i^{\mathcal{M}}} \phi^k(\xi)$ that are orthogonal and unit norm. Given such a ϕ, the Manifold Lasso objective 3.19 is the Tangent Space Lasso objective 4.3. However, note that such as ϕ will not exist for manifolds of non-trivial curvature (108).

4.4 Support Recovery Conditions

We give two types of support recovery coditions that mirror those in in Chapter 3. First, in order to ensure that the support is unique, it must be the only functionally independent set explaining the manifold. Second, the amount of colinearity present in the dictionary must be low enough to meet certain conditions that we will state in this section.

4.4.1 Uniqueness Condition

The support recovery uniqueness condition on the rank of Dg^S is identical to Proposition 36 from Chapter 3. That is, if the support S found by TSLasso is the only functionally independent subset with rank d, then it is the unique subset that parameterizes the manifold. As before, the proof invoke the implicit function theorem locally and then extrapolates to the manifold as a whole.

4.4.2 Support Recovery Consistency

Another challenge in identifying the true support stems from estimation of the tangent space. W assume that data are sampled from $\mathcal{M}$ without noise, and so incorporate the convergence rate result of (6). Our main result synthesizes this rate with the group lasso convergence rate given in Agarwal et al. (10). These results depend on the

$$\text{S-incoherence} \quad \mu_S = \max_{i=1:n, j\in S, j'\notin S} |X_{i.j}^T X_{i.j'}| \tag{4.18a}$$

$$\text{internal-colinearity} \quad \nu_S = \max_{i=1:n, \alpha\in\mathbb{R}^d:||\alpha||_2=1} \alpha^T (X_{i.S}^T X_{i.S})^{-1}\alpha. \tag{4.18b}$$

$$\tag{4.18c}$$

where X_i is the matrix constructed by stacking the gradient of dictionary functions. As in Chapter 3, the incoherence measures the correlation between functions in the true support and not in the true support, while the internal colinearity measures the amount of colinearity within the true support. One can then give conditions for support recovery consistency that depend on μ_S and ν_S.

Proposition 42. *Assume that*

1. $\mathcal{M}$ *is* $d-$*dimensional* C^k *compact manifold with positive reach.*

2. *Data* ξ *are sampled from some density* p *on* $\mathcal{M}$ *with* $p > 0$ *all over* $\mathcal{M}$.

3. $\xi \in \mathcal{M}^\circ$ *with probability 1 under* p.

Let S *be the 'true' support,* $S(\beta)$ *be the support selected by* TSLasso, μ_S *and* ν_S *be defined by* (4.18a) *and* (4.18c), *and further assume*

1. $|S| = d$.

2. Df_S *has rank* d *on* $\mathcal{M}^\circ$,

3. $\mu_S \nu_S d < 1$.

Then using the tangent space estimation algorithm in (6) with bandwidth choice $\epsilon = O(\log n/(n-1))^d$ *with* $n \geq ((1-\mu_S\nu_S d)/2\nu_S d)^{d/(k-1)}$

$$Pr(S(\widehat{\beta}) \subset S) \geq 1 - O\left((\frac{1}{n})^{\frac{k}{d}}\right) .$$

Several features differentiate this result from that in Chapter 3. First, since there is no embedding, noise stems only from tangent space estimation. Second, the convergence rate is related with the smoothness of the manifold k; a smoother manifold will have a faster converge rate. However, several features are shared. Functions with large collinearity need to be avoided in the dictionary.

4.5 Experiments

We illustrate the behavior of TSLasso on toy and real data examples. First, we establish that the algorithm is successful in simple toy examples, up to a noise threshold for tangent space estimation. Then, we apply it to small dictionaries from MDS data and show that we recover chemically meaningful results about the simulated system. Finally, we apply it to the dictionaries from Chapter 3 and show that the results are similar to those achieved using ManifoldLasso.

Dataset	n	N_a	D	d	ϵ_N	n'	p	ω
SwissRoll	100000	NA	49	2	.5	25	51	5
RigidEthanol	50000	9	50	2	.89	100	4	5
Toluene	50000	15	50	1	2.8	50	16	5
eMDA-H-H-Me	50000	12	50	1	2.3	50	8	5
Ethanol	50000	9	50	2	1.8	100	4	5
Malonaldehyde	50000	9	50	2	1.7	50	4	5
M-Xylene	50000	18	50	2	4.3	50	4	5
Dimethylfuran	50000	15	50	2	3.6	50	7	5
Ethanol	50000	9	50	2	3.5	100	12	25
Malonaldehyde	50000	9	50	2	1.	100	12	25
Toluene	50000	15	50	1	1.9	100	30	25
Ethanol	50000	9	50	2	3.5	100	756	25
Malonaldehyde	50000	9	50	2	1.	100	756	25
Ethanol	50000	9	50	2	3.5	500	756	25
Malonaldehyde	50000	9	50	2	1.	500	756	25

Table 4.1: Experimental parameters. **SwissRoll** and **RigidEthanol** are toy data, while the others are from MDS. n is the overall number of data points, N_a is the number of atoms in the molecular dynamics simulations, D is the dimension of the data set, d is the manifold dimension (assumed known), ϵ is the tangent space estimation kernel bandwidth, n' is the number of data points on which TSLasso was run, p is the size of the dictionary, and ω is the number of replicates.

Experimental setup For all of the experiments, the data consist of n data points in D dimensions. TSLasso is applied to a uniformly random subset of size $n' = |\mathcal{I}|$ using p dictionary functions, and this process is repeated ω number of times. Due to the small replicate size, we report via error bars. Note that the entire data set is used for tangent space estimation. In our experiments, the intrinsic dimension d is assumed known, but could be estimated by a method such as in Levina and Bickel (121). The local tangent space kernel bandwidth ϵ_N is estimated using the algorithm of Joncas et al. (98) for molecular dynamics data. The regularization parameter λ ranges from 0 to $\lambda_{\max}$. The last d surviving dictionary functions are chosen as the parameterization for the manifold. Parameters are summarized in Table 3.1.

Toy data We show that TSLasso recovers the correct parameterization in two noisy simulations. As in Chapter 3, we construct a Swiss Roll manifold randomly rotated into 49 dimensions, and a non-physical simulation of the ethanol molecule with only two rotational degrees of freedom, corresponding to bond torsions g^1 and g^2 (Figure 4.1). Preprocessing details are shared with Chapter 3. However, we now add Gaussian noise to the original atomic positions. Figure 4.1 demonstrates that TSLasso correctly recovers $\{g^1, g^2\}$ as the manifold parameterization, but that as high-dimensional noise is added, support recovery is impaired.

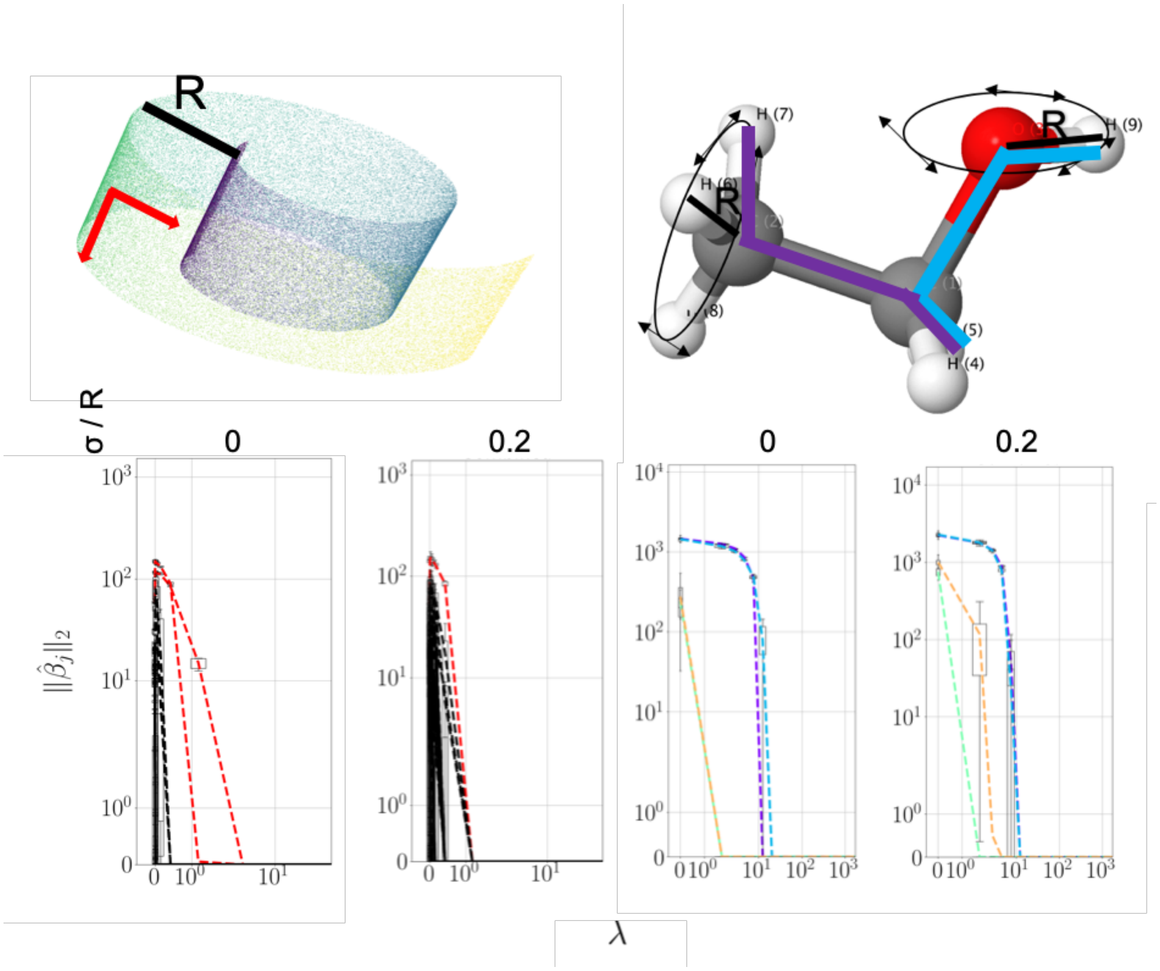
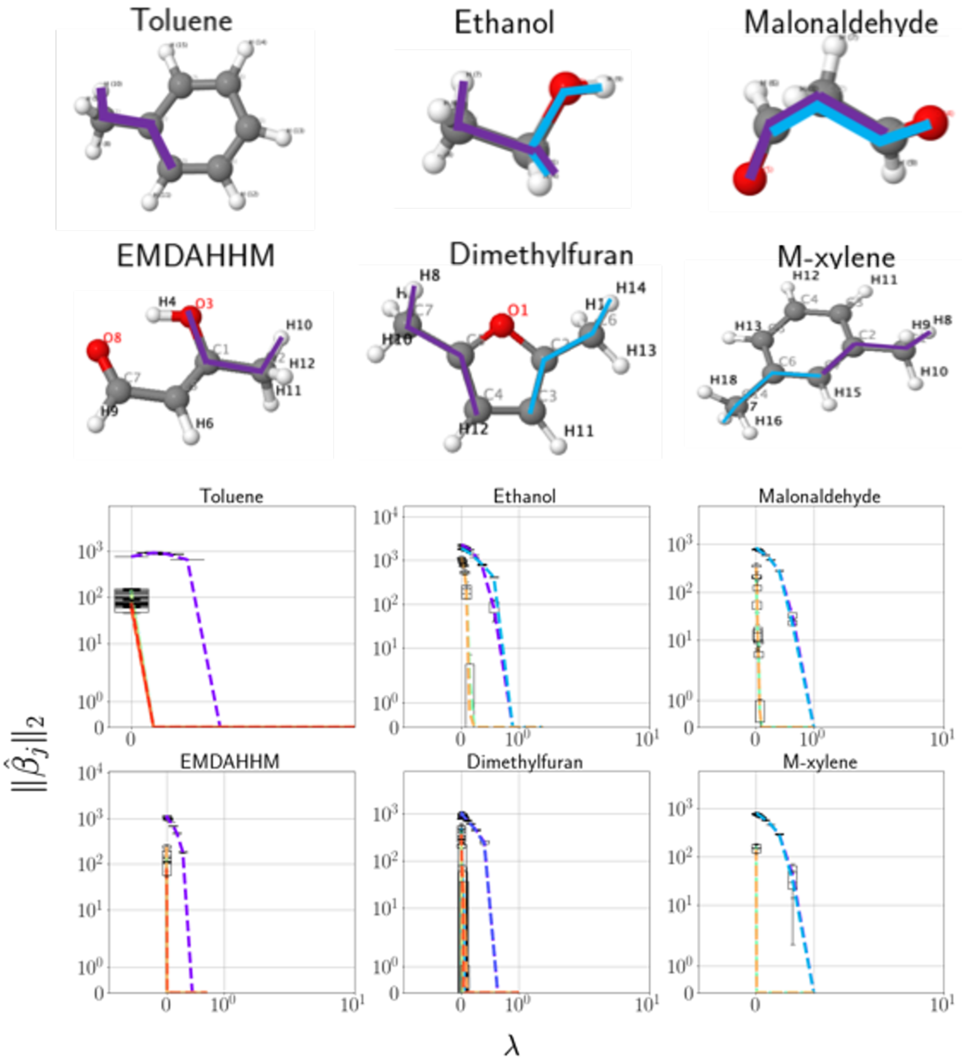


Figure 4.1: Top Left: The Swiss Roll. R refers to the reach of the manifold; σ refers to the noise level. Gradients of intrinsic coordinates are shown in red. Top Right: The ethanol molecule; the two circles correspond to the simulated degrees of freedom. These correspond to bond torsions, which are depicted by 3 contiguous same-color line segments linking 4 atoms. The torsion is the angle formed by the planes inscribing the first and last three atoms of the four. σ is the noise level, and R is the distance between the central and peripheral atoms of rotation. Bottom: Regularization paths for TSLasso for the Swiss Roll (left) and Rigid Ethanol (right) examples at different noise levels. Color of the true support matches labels in the top figures (Swiss Roll - red, Rigid Ethanol - purple and blue).

Functionally independent MDS dictionaries We apply TSLasso to small dictionaries hand chosen to contain a subset of functions satisfying our functional independence criteria Proposition 36. That is, there is a known "true" support within the dictionary. Figure 4.2 shows that TSLasso recovers the same parameterization over all replicates in a variety of MD simulations; that is, as λ is increased, the same d functions persist longest in the regularization diagram. These selected functions correspond to those torsions labelled in 4.2, and are of clear chemical significance. For comparison, we also include visual results confirming that these bond torsions are a parameterization of the data manifold. Figure 4.3 shows embeddings of the six assayed molecule datasets generated using the Diffusion Maps algorithm. The results obtained by TSLasso do not make use of these embeddings, but these verify that these bond torsions do in fact parameterize the data manifold. These results suggest that the local denoising property of the tangent space estimation, coupled with the global regularity imposed by the assumption that the manifold is parameterized by the same functions throughout, is sufficient to identify the slow modes of molecular motion. For these embeddings, hyperparameters ϵ are set as in Table 4.1. These embeddings are colored by the bond torsions selected by TSLasso.



(a)

Figure 4.2: The top figures show standard molecular bond models of the molecules generating our analyzed MDS data. The bottom figures show regularization paths over replications from TSLasso. The variable selection procedure is to increase λ until only d torsions from the original molecule are retained. The d selected torsions are plotted in the left figure with colors corresponding to the regularization paths on the right. For the sake of visualization, we have purposely reordered the dictionary so that the selected torsions always correspond to g^1 or g^2. g^1 is shown in purple, and g^2 in blue.

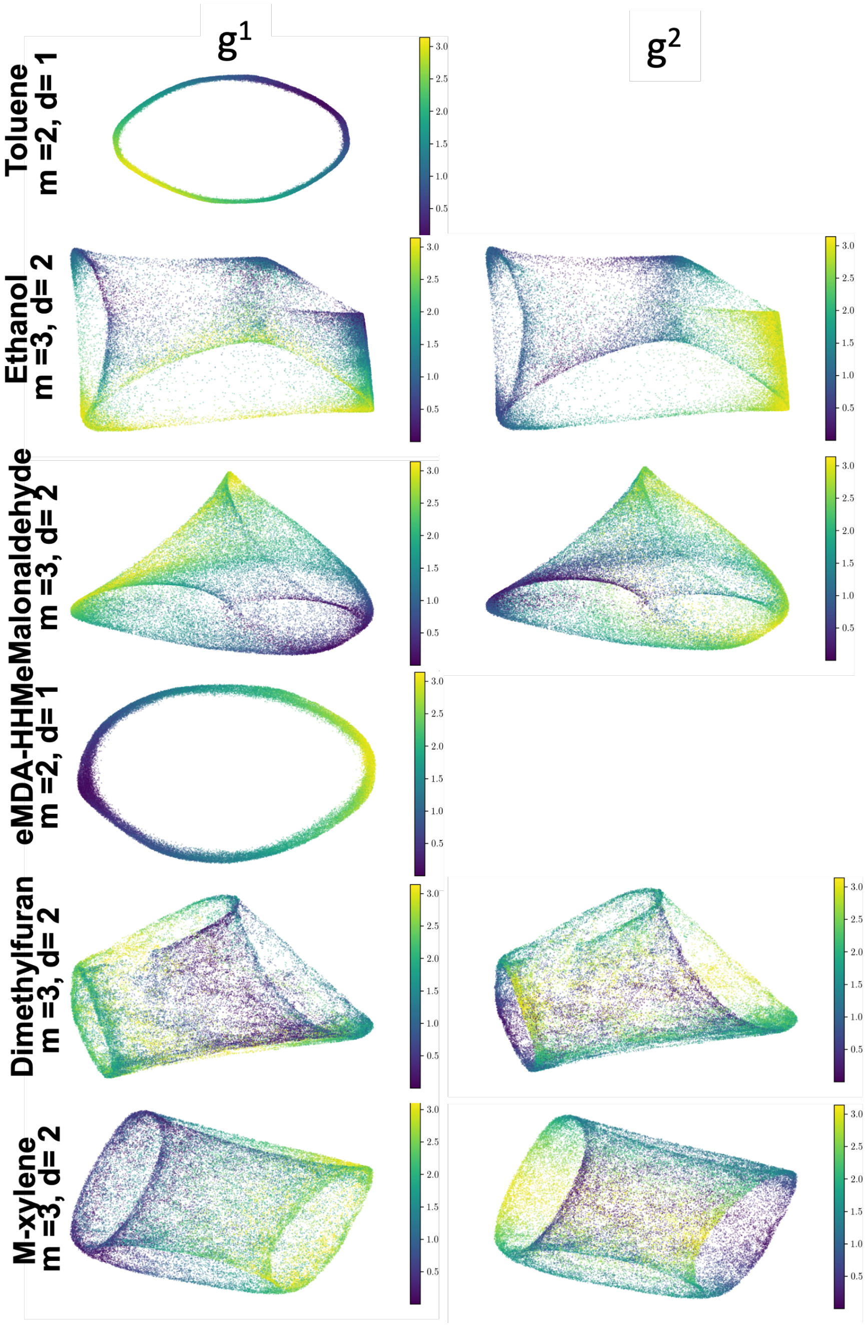
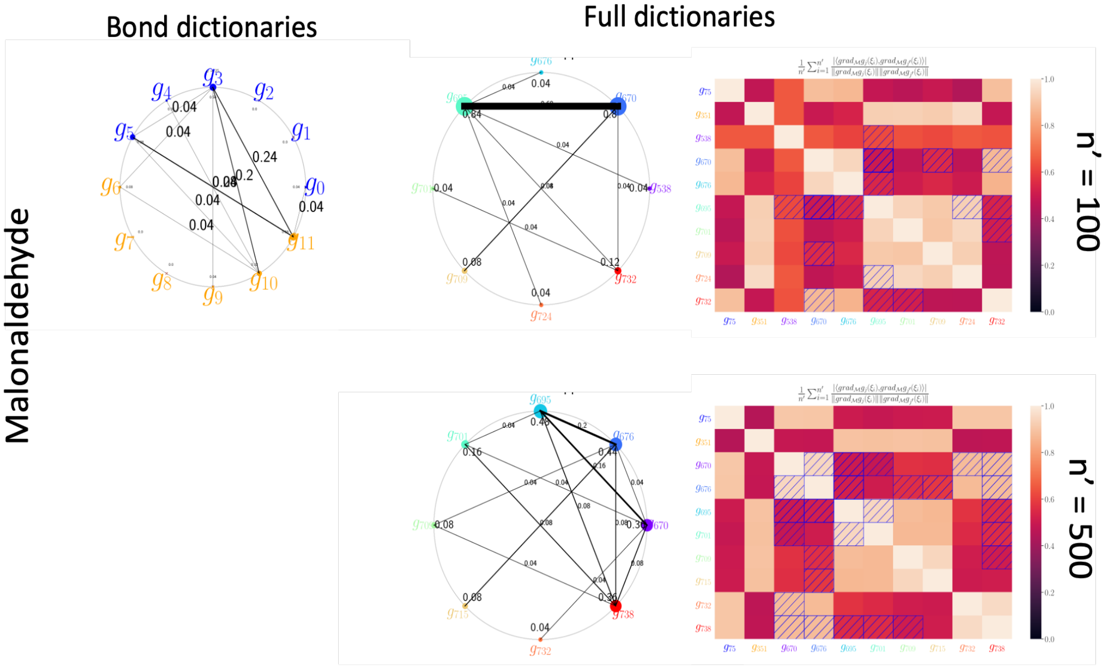


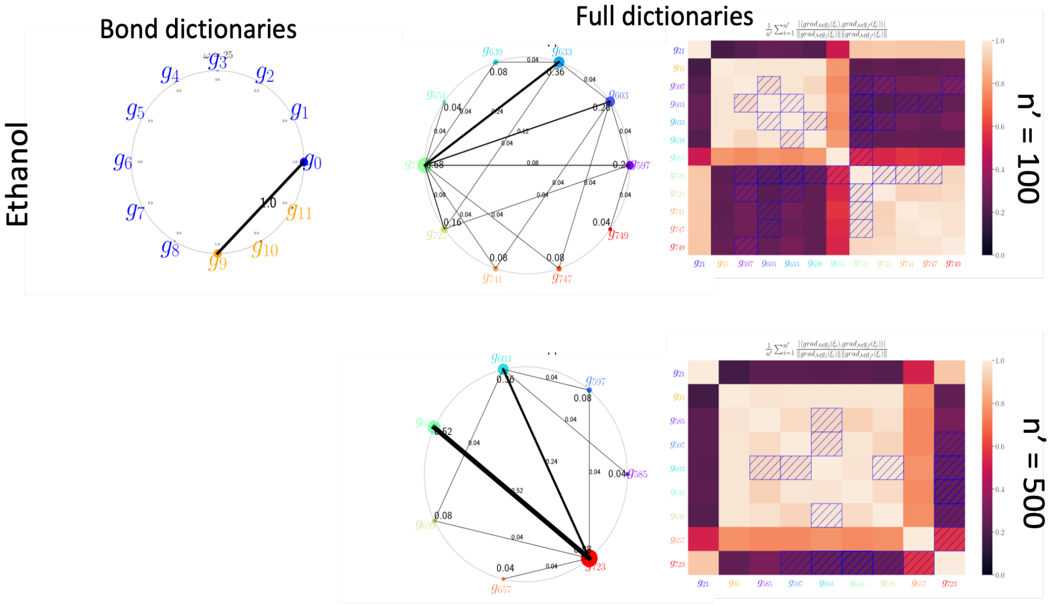
Figure 4.3: Diffusion maps embeddings of our datasets, colored by selected bond torsions from small hand-selected dictionaries.

Diagram and full dictionaries We also replicate the results from Chapter 3 using the simplified TSLasso algorithm. These show similar results to ManifoldLasso, and thereby show that the incorporation of the embedding coordinate gradients was unnecessary. For dictionaries consisting of torsions implicitly defined by bond diagrams, TSLasso recovers a "correct" support consisting of opposing bond torsion almost all of the time in both ethanol and malonaldehyde. On full dictionaries consisting off all possible torsions, TSLasso also behaves similarly to ManifoldLasso in that is generally successful, while sometimes selecting collinear supports. Increasing the number of assayed point to $n' = 500$ does not have an appreciable effect on stability. Note that the change in collinearities for **Malonaldehyde** compared with Chapter 3 are due to the different bandwidth parameter ϵ. Similarly, g^{21}, g^{35} and g^{75}, g^{351} and are now ground truth for **Ethanol** and **Malonaldehyde** , respectively.



(a)

Figure 4.4: Left: **Malonaldehyde** support estimated using TSLasso with diagram dictionary and $n' = 100$. Right: **Malonaldehyde** support estimated using TSLasso with full dictionary and $n' = 100$. Bottom: **Malonaldehyde** support estimated using TSLasso with full dictionary and $n' = 500$.



(a)

Figure 4.5: Left: **Ethanol** support estimated using TSLasso with diagram dictionary and $n' = 100$. Right: **Ethanol** support estimated using TSLasso with full dictionary and $n' = 100$. Bottom: **Ethanol** support estimated using TSLasso with full dictionary and $n' = 500$.

4.6 *Related work*

We distinguish between our approaches and purely non-parametric methods that attempt to learn a parameterization of $\mathcal{M}$. For example, Saul and Roweis (163) and Teh and Roweis (178) and references therein propose parametrizing $\mathcal{M}$ by finite mixtures of local linear models, aligned so as to provides global coordinates, in a way reminiscent of Local Tangent Space Alignment (203). Another idea is to use d eigenfunctions of the Laplace-Beltrami operator $\Delta_{\mathcal{M}}$ as a parametrization of $\mathcal{M}$. The Diffusion Maps coordinates could be considered such a parameterization (49; 50; 74). However, these are not in and of themselves interpretable, and it is not clear how many such coordinates are needed (43). Mohammed and Narayanan (138) showed that principal curves and surfaces can provide an approximate manifold parametrization. These methods can often be used as embedding algorithms in our approach, but make no attempts at synergizing with an interpretable dictionary. Roweis and Saul (161); Dsilva et al. (59); Kohli et al. (109); Chen and Meilă (43) tackle the related problem of choosing among the infinitely many Laplacian eigenfunctions a set which provide a d-dimensional parameterization of the manifold. However, these methods fail to provide physical meaning for the selected functions.

We note several distinctions between the TSLasso method and the ManifoldLasso method in Chapter 3. In view of recovering a support that satisfies Proposition 36 TSLasso is conceptually simpler, by directly providing a parametrization of $\mathcal{M}$, as well as computationally more efficient, since it doesn't require computation of an embedding. On the other hand, ManifoldLasso uses is able to explain individual embedding coordinate functions. In contrast, we have no consistent matching between dictionary functions and unit vectors in I_d, and so can only provide an overall regularization path, rather than one corresponding to individual tangent basis vectors. Although TSLasso is inspired by ManifoldLasso , the tangent bases are not themselves gradients of a known function, and, indeed it may not be the case that such a function even exists.

With respect to sparse regression, the seminal group lasso paper of Yuan and Lin (197) and support recovery analyses of Elyaderani et al. (62); Wainwright (187) are central to our approach. However, our use of replicates in experiments is reminiscent of the Stability Selection method of Meinshausen and Bühlmann (133). Such methods address instabilities of the variable selection, in particular, when restrictive theoretical conditions are violated (204; 90). The empirically-based two-stage OLS-hybrid approach we elucidate in Chapter 5 for resolving this issue is based on ideas in Efron et al. (61); Meinshausen (132); Hesterberg et al. (88). Some attractive alternate approaches to this problem that we do not pursue are the use of non-convex penalties such as SCAD (65; 33), weighted data points in the

Adaptive Lasso (206), and the elastic net, which induces correlated functions to reach zero simultaneously on the regularization path (207) We additionally note the method of Haufe et al. (82), which applies group lasso to analyze sparse decomposition of vectors fields, albeit in a different setting.

Our approach is related to methods like autoencoders (77) and factor models (195) that give functional forms for learned low-dimensional representations. However, the explanations we obtain are endowed with the meaning of the domain specific dictionaries. Less obviously, descriptors like principal curves or Laplacian eigenfunctions are generally still non-parametric (i.e exist in infinite dimensional function spaces), while the parameterizations by dictionaries we obtain (e.g. the torsions) are in finite dimensional spaces. This distinction is mirrored in comparison with the many so-called *dictionary learning* methods in which a low-dimensional transformation is learned simultaneously with its inverse. We note that our method is not dictionary learning per se, but rather sparse coding, in which the dictionary is given (177).

Our approach is particularly relevant to enhanced sampling methods in molecular dynamics (159; 160; 69; 70). In these methods, exploration of the molecular state space is accelerated through biasing of simulation towards directions of large scale variation identified through visual inspection of denoised embedding coordinates. More recently, reinforcement-learning type syntheses of these ideas have been applied (188; 148; 173). The eigenfunctions of the Laplacian have special relevance to quantum systems, for which the quantum correspondence principle states that classical dynamics should be observable in so-called stable eigenstates (86; 198; 113; 176).

Although in our application our dictionary consists of functions with physical meaning, our general principal of finding parametric geometrically-motivated approximations of learned representations is relevant to a range of machine learning contexts. Examining functions in embedding coordinates is quite typical in genomics (11), and much deep learning work also makes use of explicit traversal of a latent space (124; 172). Learned gradients provide interpretable (9) or otherwise statistically-useful information (192; 52; 196). However, our approach relies on WLPCA for tangent space estimation (98; 6). Improvement of this estimator in the presence of noise is an active area of research (155).

4.7 Discussion

The Tangent Space and Manifold Lasso methods introduced in this and the previous chapter apply a new tool - group lasso - to the problem of interpretable manifold parameterization. These methods are robust to non-linearity in both the algorithm and the covariates. They require functions that are smooth, as well as the assumption that the data lie on a smooth manifold. Both methods use differentials of functional covariates that are available analytically, and the ManifoldLasso method also uses differentials of the representation learned by a manifold embedding algorithm.

One possible modification to these methods is to analyze gradients in the coordinates of the ambient space $\mathbb{R}^D$ rather than in the manifold tangent space. That is, the regression problems would be solved using the same group lasso machinery in $\mathcal{T}\mathbb{R}^D$ rather than $\mathcal{T}\mathcal{M}$ In this variant, the greater penalty accrued by off-manifold gradients in TSLasso and ManifoldLasso due to their being normalized prior to projection is substituted by the greater penalty necessary to reconstruct the identity using a set of off-manifold gradients. As a related alternative, we could estimate gradients of interpretable dictionary functions in the embedding space, and use the pushforward Riemmanian metric as the response variable in a tangent space lasso or basis pursuit.

A question raised by this approach is the extent to which the mappings learned using Tangent Space Lasso are statistically preferable to a fully non-parametric approaches. The parametric form of g^S provides a more straightforward way to conduct out-of-sample extension of the learned embedding than classical methods such as Nystrom extension (111) or Nadaraya-Watson type estimators (189). Since the functions g^j defined in an neighborhood of $\mathcal{M}$ in $\mathbb{R}^D$, g^S can be extended to the ambient space around $\mathcal{M}$, and we can project points points ξ lying near $\mathcal{M}$ to $g^S(\mathcal{M})$. Although our method makes uses of a parameterized dictionary, we cannot currently prove that S may be estimated at parametric rates, even with a fixed p and the assumption of noiseless data. This is because tangent space estimation requires a number of samples exponential in d (6). Speculatively, an assumption on global regularity and a weak learnability-type argument based on the ability of disjoint neighborhoods to generate better-than-random guesses might justify a faster convergence rate.

A deficiency of these methods is that the support recovery conditions are restrictive. For example, any pair of feature coordinates would be a valid parameterization for the Swiss Roll. Any diffeomorphic transformation of g^S is also a parametrizing chart of $\mathcal{M}$, and therefore the dictionary $\mathcal{G}$ can contain any diffeomorphic transformation of the true support and still satisfy the rank condition Proposition 36. However, in our molecular dynamics examples, it is almost unavoidable to construct an overcomplete dictionary containing multiple possible solutions. A second challenge is collinearity. Similar to κ_S and γ_{max} in Chapter 3, μ_S is a maximum over all data points, and so a single outlying data point can violate our recovery conditions. Indeed, the failure of sparsity-inducing algorithms to achieve stability and consistency in a board variety of settings is well-known (91).

Chapter 5

MANIFOLD AND TANGENT SPACE BASIS PURSUIT

The regularized regression methods from Chapters 3 and 4 are challenged by our application to molecular dynamics data. Overcomplete dictionaries - dictionaries for which there are multiple functionally independent full rank solutions - are both unavoidable and problematic for our estimators. As a response to this challenge, this chapter introduces Manifold and Tangent Space Basis Pursuit, related algorithms that give a more specific characterization of support recovery. These algorithms use more robust optimization at the expense of computational tractability. We therefore also introduce a hybrid of these approaches with the Manifold and Tangent Space Lassos. Results are given on the overcomplete dictionaries from the previous chapters.

5.1 Introduction

Our theoretical conditions for successful support recovery using the methodology of the previous chapters are challenged by large dictionaries in several ways. First, the functional independence condition that the gradients span the tangent space can often be satisfied by ”uninformative” functions like the coordinate functions of the ambient space in the Swiss Roll example. In this case, there is no longer a well-defined notion of what it means to successfully recover the true functional support. Second, large dictionaries cause collinearity that violates our support recovery conditions. These problems are hard to avoid. For example, the 9 atom molecule ethanol has 756 potential bond torsions, but its slow mode is only a 2 dimensional manifold within a 20 dimensional shape space. Unless there is a disproportionate concentration off torsions in the directions normal to this manifold, there will be multiple possible parameterizations. A finer notion of support recovery and an algorithm for recovering this support are therefore desireable.

Overcompleteness describes the presence of multiple possible solutions within a dictionary. Methods for selecting between basis in overcomplete dictionaries are typically required in domains such as wavelet decomposition for which a dictionary may be constructed programmatically Molecular dynamics is such an area, since we can construct new dictionary elements at minimal computational cost, and there is no a priori differentiating feature of an particular function such as there would be for a system with measured observables. Inspired by techniques for selecting from overcomplete dictionaries in sparse coding, we introduce an alteration of the Manifold and Tangent Space Lassos that directly optimizes the penalty term. We call these methods Manifold and Tangent Space Basis Pursuit because of their explicit solving of basis pursuit versions of Manifold and Tangent Space Lasso. These algorithms are subtly different from the combinatorial duals of the Manifold and Tangent Space Lasso objective functions, and are sometimes better-suited for selecting parameterizing support functions from within elements of overcomplete dictionaries.

Section 5.2 gives background on sparse regression, the Manifold and Tangent Space Lassos, and manifold learning. Section 5.3 introduces the Tangent Space and Manifold Basis Pursuit objective functions and algorithms. Section 5.4 shows that the Tangent Space Basis Pursuit objective is related to the mathematical notion of isometry. Since these adaptations also make our algorithm less tractable, Section 5.5 introduces a two-stage method that combines our the Manifold and Tangent Space Lassos with their basis pursuit counterparts. Results on molecular experiments are given in Section 5.6.

5.2 Background

Our motivating challenge is the definition and selection of a "best" parameterizing subset of functions elements from within an dictionary consisting of multiple d sets of functionally independent functions, also known as an overcomplete dictionary. The Manifold and Tangent Space Basis Pursuit methods are variants of the Manifold and Tangent Space Lassos that are well-suited for this challenge. Although they are inspired by convex duals, they are subtly different due to the cardinality-constraint justified by our manifold support estimation task. In this section, we review relevant background in sparse regression, convex duality, and manifold learning.

5.2.1 Manifold learning

Manifold learning components such as embedding and tangent space estimation algorithm are shared with previous chapters previous chapters. As we focus on discriminating between parametrizations by their metric properties, we recall Proposition 23 which states that, in ambient metric, $D\phi(\xi)$ is unitary at all $\xi \in \mathcal{M}$ if and only if ϕ is an isometry. This is strictly stronger and more specific characterization of $D\phi$ than the full-rank condition required for functional independence. In Section 5.4, we explain a connection between isometry and basis pursuit in our manifold estimation setting.

5.2.2 Basis pursuit

The equivalent combinatorial optimization problem for Group Lasso is known as Group Sparse Basis Pursuit (GSBP) (157). The Manifold and Tangent Space Lasso objectives from Chapters 3 and 4 therefore find minimizers of their combinatorial GSBP equivalents. As in the previous chapters, we assume that we are given a dataset $\mathcal{D} \in \mathbb{R}^{n \times D}$ sampled from a d dimensional smooth manifold $\mathcal{M}$, a smooth embedding ϕ from $\mathbb{R}^D$ to $\mathbb{R}^m$, and a dictionary of smooth functions $\mathcal{G} : U \to \mathbb{R}^p$ defined on an open set U containing $\mathcal{M}$. Let

$$X = [\operatorname{grad}_{T_i\mathcal{M}}\, g^j(\xi_i)\ i \in [n], j \in [p]] \in \mathbb{R}^{n \times d \times p} \tag{5.1}$$

$$Y = [\operatorname{grad}_{T_i\mathcal{M}}\, \phi^k(\xi_i)\ i \in [n], k \in [m]] \in \mathbb{R}^{n \times d \times m}, \tag{5.2}$$

where these gradients have been normalized in the manner of Chapters 3 and 4. Also, given a array $\beta \in \mathbb{R}^{n \times p \times d}$, define the penalties

$$\|\beta\|_{2,0,2} := \sum_{j=1}^{p} 1_{\|\beta_{.j.}\|_F \neq 0}, \tag{5.3}$$

$$\|\beta\|_{2,1,2} := \sum_{j=1}^{p} \|\beta_{.j.}\|_F. \tag{5.4}$$

The GSBP equivalents of the Manifold and Tangent Space Lasso objective functions are then

$$J_{M,C=C}(\ X,\ Y,\beta) := \|\beta\|_{2,1,2}\ s.t.\ (\sum_{i=1}^{n} \|Y_{i..} -\ X_{i..}\beta_{i..}\|_F^2)^{1/2} \leq C \tag{5.5}$$

$$J_{TS,C=C}(\ X,\beta) := \|\beta\|_{2,1,2}\ s.t.\ (\sum_{i=1}^{n} \|I_d -\ X_{i..}\beta_{i..}\|_F^2)^{1/2} \leq C \tag{5.6}$$

These are the objectives which are minimized by values of λ approaching zero in the Manifold and Tangent Space Lasso objectives $J_{M,\lambda}$ and $J_{TS,\lambda}$ In particular, C and λ are in a monotonic relationship. A larger value of λ increases the weight on the penalty term, and corresponds to a larger constraint size C.

In Chapters 3 and 4, the theoretical goals of these algorithms were to select from $\mathcal{G}$ any d smooth functions whose gradients are everywhere full rank. Such a set is a minimizer of $\ell 0$ objectives

$$J_{M,\ell 0}(\ X,\ Y,\beta) = \|\beta\|_{2,0,2}\ s.t.\ \ Y_{i..} =\ X_{i..}\beta_{i..} \text{ for all } i = 1:n, \text{ and } k = 1:m, \tag{5.7}$$

$$J_{TS,\ell 0}(\ X,\beta) = \|\beta\|_{2,0,2}\ s.t.\ I_d =\ X_{i..}\beta_{i..} \text{ for all } i = 1:n, \text{ and } k = 1:d. \tag{5.8}$$

Therefore, the sufficient support recovery conditions in Chapter 4 for minimizers of objectives 5.5 and 5.6 to select minimizers of objectives 5.7 and 5.8 are related to those required for Program 2.15 to select a solution of program 2.14. When these conditions are violated, the solution path of the lasso regularized problem may not contain maximally sparse solution (179). On the other hand, while the maximally sparse solution is not unique for overcomplete dictionaries, objectives 5.5 and 5.6 can often differentiate between multiple sets of functions S, S' that are $\ell 0$-optimum in the sense that $|S| = |S'| = d$. Thus, we can see that the $\ell 0$ and group lasso programs are both in-and-of-themselves problematic - the former because it does not lead to a unique solution, and the latter because it may not find a parameterization that is maximally sparse.

Problem Statement We pose problems similarly to Sections 3.2 and 4.2, with an important distinction. We are given data $\mathcal{D} = \{\xi_i \in \mathbb{R}^D : i \in 1 \dots n\}$ sampled i.i.d. from a smooth manifold $\mathcal{M}$ of intrinsic dimension d embedded in a feature space $\mathbb{R}^D$ by the inclusion map, and a dictionary of smooth functions $\mathcal{G} = \{g^1, \dots g^p, \text{ s.t. } g^j : U \subseteq \mathbb{R}^D \to \mathbb{R}\}$, where U is an open set containing $\mathcal{M}$. In the previous Chapters, we sought to identify a set of d distinct functions $g^S : \{g^j\}_{j \in S}, S \in [p]_d$, the set of d element samples from $[p]$ without replacement, such that at every point ξ, g^S is a diffeomorphism on an open neighborhood $U \subset \mathcal{M}$ containing ξ to $g^S(U) \subset \mathbb{R}^{|S|}$. Here, we not only identify a suitable sparse local d function diffeomorphism, but also explicitly aim to minimize $J_{M,C=0}$ and $J_{TS,C=0}$.

5.3 Manifold and Tangent Group Sparse Basis Pursuit

We therefore introduce a cardinality-constrained version of the group sparse basis pursuit objectives 5.5 and 5.6 for manifold support recovery. Our Manifold and Tangent Basis Pursuit programs are, respectively,

$$\arg \min_{\beta \in \mathbb{R}^{n \times p \times m} : |S(\beta)| = d} J_{M,C=0}(X, Y, \beta) \tag{5.9}$$

$$\arg \min_{\beta \in \mathbb{R}^{n \times p \times d} : |S(\beta)| = d} J_{TS,C=0}(X, \beta). \tag{5.10}$$

The constraint $|S(\beta)| = d$ is equivalent to saying that the solution is a minimizer of Objectives 5.7 or 5.8, respectively. Despite their familiar form, these objectives are not equivalent to the Manifold and Tangent Space Lassos. The solution to a $\ell 1$ regularized problems may not always solve the $\ell 0$ equivalent, and so constraining the solution to satisfy both optima gives a different objective. To see this another way, note that many functions are non-zero early in the regularization paths in Chapters 3 and 4, and to select only d functions, λ must be quite close to its maximum.

We could naively optimize MBasisPursuit and TSBasisPursuit by exhaustive search. In practice, this means enumerating subsets S of $[p]$ of size d and independently fitting least squares $\textsc{ols}(X_{i.S}, Y_i)$ or $\textsc{ols}(X_{i.S}, I_d)$ before summing up the squared losses across i to get a loss associated with each subset S. To accelerate this process, we take an alternative approach in Section 5.5. Other algorithmic steps such as normalization of gradient fields, tangent space estimation, and using a subset $I \subset [n]$ such that $|I| = n'$ for computational improvement are performed as in Chapters 3 and 5.

Computation The principal drawback of this approach is computational. The algorithmic runtime to explore the set of possible supports is $O(\frac{p!}{d!(p-d)!}(n'(d^2p + p^3))$. The factor $\frac{p!}{d!(p-d)!}$ necessary to perform exhaustive search over d element subsets of a p element dictionary is avoided by the convex methods of the previous chapters. Other computational details are shared.

Algorithm 6 MBasisPursuit(Dataset $\mathcal{D}$, dictionary $\mathcal{G}$, embedding coordinates $\phi(\mathcal{D})$, intrinsic dimension d, kernel bandwidth ϵ, neighborhood cutoff size r)

```
Construct $\mathcal{N}_i$ for $i = 1 : n$; $i' \in \mathcal{N}_i$ iff $\|\xi_{i'} - \xi_i\|_2 \leq r_N$, and local data matrices $\Xi_{1:n}$
Construct kernel matrix $K \leftarrow \textsc{Laplacian}(\mathcal{N}_{1:n}, \Xi_{1:n}, \epsilon_N)$
for $j = 1, 2, \ldots p$ do
    Compute $\nabla_\xi g^j(\xi_i)$ for $i = 1, \ldots n$
    Compute $\zeta_j^2$ by (3.21) and normalize $\nabla_\xi g^j(\xi_i) \leftarrow (1/\zeta_j)\nabla_\xi g^j(\xi_i)$ for $i = 1, \ldots n$
end for
for $i = 1, 2, \ldots n$ do
    Compute basis $T_i^{\mathcal{M}} \leftarrow \textsc{wlpca}(\Xi_i, K_{i,\mathcal{N}_i}, d)$
    Project $X_{i..} \leftarrow (T_i^{\mathcal{M}})^T \nabla_\xi g^{1:p}$
    Compute $Y_{i..} \leftarrow \textsc{PullBackDPhi}(\Xi_i, \Phi_i, T_i^{\mathcal{M}}, L_{i,\mathcal{N}_i}, d)$
end for
Compute $\zeta_k^2 \leftarrow \frac{1}{n}\sum_{i=1}^n \|Y_{i.k}\|^2$ (i.e. (3.20)), for $k = 1, \ldots m$ and
normalize $Y_{i..} \leftarrow Y_{i..}\,\mathrm{diag}\{1/\zeta_{1:m}\}$, for $i = 1, \ldots n$.
$\beta \leftarrow \arg\min_{\beta \in \mathbb{R}^{n \times p \times m} : |S(\beta)| = d} J_{M,C=0}(X, Y, \beta)$
Output $S = \mathrm{supp}\,\beta$
```

Algorithm 7 TSBasisPursuit(Dataset $\mathcal{D}$, dictionary $\mathcal{G}$, intrinsic dimension d, kernel bandwidth ϵ, neighborhood cutoff size r)

```
Construct $\mathcal{N}_i$ for $i = 1:n$; $i' \in \mathcal{N}_i$ iff $\|\xi_{i'} - \xi_i\|_2 \leq r_N$, and local data matrices $\Xi_{1:n}$
Construct kernel matrix $K \leftarrow \exp(\frac{\|\Xi_i - 1_{k_i}\xi_i\|_2{}^2}{\epsilon})$ for $\epsilon = \frac{r}{3}$
for $j = 1, 2, \ldots p$ do
    Compute $\nabla_\xi g^j(\xi_i)$ for $i = 1, \ldots n$
    Compute $\zeta_j^2$ by (3.21) and normalize $\nabla_\xi g^j(\xi_i) \leftarrow (1/\zeta_j)\nabla_\xi g^j(\xi_i)$ for $i = 1, \ldots n$
end for
for $i = 1, 2, \ldots n$ do
    Compute basis $T_i^{\mathcal{M}} \leftarrow$WLPCA$(\Xi_i, K_{i,\mathcal{N}_i}, d)$
    Project $X_{i..} \leftarrow (T_i^{\mathcal{M}})^T \nabla_\xi g^{1:p}$
end for
$\beta \leftarrow \arg\min_{\beta \in \mathbb{R}^{n \times p \times m}: |S(\beta)| = d} J_{TS,C=0}(X, \beta)$
Output $S = \operatorname{supp} \beta$
```

5.4 Theory

Empirically, both the Manifold and Tangent Space Basis Pursuit tend to select gradient bundles that are orthogonal and slowly varying. In this section, we examine how the Tangent Space Basis Pursuit minimizer in particular can satisfy an interesting mathematical property. We show that, given perfect tangent space estimation, if there exists a mapping g^S that is an isometry, then it will be selected by TSBasisPursuit.

5.4.1 Preliminaries

We introduce some streamlined notation. Given a multiway array $\beta \in \mathbb{R}^{n \times p \times d}$ with $p > d$, define the TSBasisPursuit loss function

$$l(\beta) = \|\beta\|_{2,1,2} \tag{5.11}$$

Using this loss, assuming $p = d$ and β_i is full rank, define a dual loss l^* w.r.t. an array $X \in \mathbb{R}^{n \times d \times d}$

$$l^*(X) = l(\beta) : I_d = X_{i..}\beta_{i..}.$$

That is, $\beta_{i..}$ is the right inverse of $X_{i..}$. Note that here we have constrained $p = d$ since we seek a d function support.

Define the $n_{2,2} : R^{n \times d} \to R^{n \times d}$ normalization maps

$$\tilde{V} := n_{2,2}(V) = \frac{\sqrt{n}V}{\|V\|_F} \tag{5.12}$$

and $n_{2,.,2} : R^{n \times p \times d} \to R^{n \times p \times d}$

$$\tilde{X} := n_{2,2,.}(X) = [n_{2,2}(X_{..j})\ j \in 1 \ldots p] \tag{5.13}$$

Normalization by (5.12) creates an equivalence between vector fields that differ by a constant factor. This then allows us to define the invariant loss

$$\tilde{l}^*(X) := l^*(\tilde{X}). \tag{5.14}$$

We will show that $\tilde{l}^*(X)$ is lower for sets of vectors $X_{i.S}$ which are orthogonal, of constant length, and tangent to the manifold. That is, $\tilde{X}_{i.S}$ is unitary.

Before proceeding, we require the following piece of Lemma 39.

Lemma 43. *Consider two sets of vector fields X and X^i where $X^i_{i..} = UX_{i..}$, where U is unitary and $X_{i'..} = X^i_{i'..}$ for other values $i' \neq i$. Then $l^*(X) = l^*(X^i)$*

Proof. Without loss of generality, let $i = 1$. We can write

$$l^*(X^i) = l(\beta^i) = \sum_{j=1}^{p} (\sum_{i'=2}^{n} \|\beta_{i'j.}\|_2^2 + \|\beta^i_{1j.}\|_2^2)^{1/2} = \sum_{j=1}^{p} (\sum_{i'=1}^{n} \|\beta_{i'j.}U\|_2^2)^{1/2} = l^*(X) \tag{5.15}$$

where the second to last equality is because the norm $\|v\|_2^2$ is unitary invariant. □

5.4.2 Selection of orthonormal sets of vectors

We next show that $\tilde{l}^*(X_{..S})$ is minimized by orthonormal sets of vectors. That is, $\tilde{l}^*(X_{..S})$ is lower for $X_{..S}$ such that $\tilde{X}_{i.S}$ is unitary. This is summarized by the following proposition.

Proposition 44. *Suppose we have a set of vector fields $X \in \mathbb{R}^{n \times d \times p}$ that includes a subset $X_{..S^+} \in \mathbb{R}^{n \times d \times d}$ indexed by a d-subset $S^+ \in [p]_d$ such that, for all $i \in [n]$, $X_{i.S_j{}^+}$ are mutually orthogonal over j and constant length over i. That is, for all i, we have $X^T_{i.S^+_j} X_{i.S^+_{j'}} = 0$ for all $j \neq j' \in [d]$, and $\|X_{i.S^+_j}\| = c_j$. Then*

$$S^+ = \arg\min_{S \in [p]_d} \tilde{l}^*(X_{..S})$$

Proof. Our proof strategy is to first show that $\tilde{l}^*(X_{..S^+}) = d\sqrt{n}$. Then, we will show that this gives a lower bound on $\tilde{l}^*(X_{..S})$ for all d element partitions of $[p]$.

Lemma 45. *$\tilde{l}^*(X_{..S^+}) = d\sqrt{n}$.*

Proof. $\tilde{l}^*(X_{..S^+}) = l^*(\tilde{X}_{..S^+})$. The matrices $\tilde{X}_{..j}$ are normalized to have norm $\sqrt{n}$ for each j. Since the lengths of vectors $\tilde{X}_{i.j}$ are equal across i by construction, this mean that $\|\tilde{X}_{i.j}\| = 1$

for each i. By Proposition 43, without loss of generality, we can thus consider

$$\beta_{ijk} = \begin{cases} 1 & j = k \in \{1 \dots d\} \\ 0 & \text{otherwise} \end{cases}. \tag{5.16}$$

Then $\|\beta_{.j.}\|_2 = \sqrt{\sum_{i=1}^n 1} = \sqrt{n}$, and so $\tilde{l}^*(X_{..S^+}) = d\sqrt{n}$. □

Now we show that, if some other $d-$ subset S does not have orthonormal $\tilde{X}_{i.S}$, then $\tilde{l}^*(X) > d\sqrt{n}$. A full rank $\tilde{X}_{i.S}$ can fail to be orthonormal in two ways. Either at least two gradients are not orthogonal, or at least one vector has norm $\neq 1$. Thus, we show that orthogonality lowers l^* over non-orthogonality, and constant length lowers l^* over non-constant length. We first consider the former case.

Lemma 46. *Let $X_{..S} \in \mathbb{R}^{n \times d \times d}$ be defined as above and let $X'_{..S}$ be an array such that $\|X'_{i.S_j}\|_2 = \|X_{i.S_j}\|_2$ for all $i \in [n], j \in [d]$ and $X'_{i.S}$ is column-orthogonal $\forall i \in [n]$. Then $\tilde{l}^*(X_{..S}) > \tilde{l}^*(X'_{..S})$.*

Proof. For every data point i, by Lemma 43, without loss of generality

$$\beta_{ijk}^i = \begin{cases} \|\tilde{X}'_{i.S_j}\|_2^{-1} & j = k \in \{1 \dots d\} \\ 0 & \text{otherwise} \end{cases}. \tag{5.17}$$

Therefore,

$$\tilde{l}^*(X') = \sum_{j=1}^d \sqrt{\sum_{i=1}^n \|\tilde{X}'_{i.S_j}\|_2^{-2}}. \tag{5.18}$$

On the other hand, the invertible matrices $\tilde{X}_{i.S}$ admit QR decompositions $\tilde{X}_{i.S} = QR$ where Q and R are square unitary and upper-triangular matrices, respectively (13). Since l^* is invariant to unitary transformations, we can without loss of generality, consider $Q = I_d$.

Denoting I_d to be composed of basis vectors $[e^1 \dots e^d]$, the matrix R has form

$$R = \begin{bmatrix} \langle e^1, \tilde{X}_{i.S_1} \rangle & \langle e^1, \tilde{X}_{i.S_2} \rangle & \dots & \langle e^1, \tilde{X}_{i.S_d} \rangle \\ 0 & \langle e^2, \tilde{X}_{i.S_2} \rangle & \dots & \langle e^2, \tilde{X}_{i.S_d} \rangle \\ 0 & 0 & \dots & \dots \\ 0 & 0 & 0 & \langle e^d, \tilde{X}_{i.S_d} \rangle \end{bmatrix}. \tag{5.19}$$

The diagonal entries $R_{jj} = \langle q^j, \tilde{X}_{i.S_j} \rangle$ of this matrix have form $\|\tilde{X}_{i.S_j} - \sum_{j' \in \{1 \dots j-1\}} \langle \tilde{X}_{i.S_j}, e^{j'} \rangle e^{j'} \|$. Thus, $R_{jj} \in (0, \|\tilde{X}_{i.S_j}\|]$. On the other hand $\beta_{iS.} = R^{-1}$, which has diagonal elements $\beta_{jj} = R_{jj}^{-1}$, since R is upper triangular. Thus, $\beta_{ijj} \geq \|\tilde{X}_{i.S_j}\|^{-1}$, and therefore $\|\beta_{iS_j.}\| \geq \|\beta'_{iS_j.}\|$. Since $\|\beta_{iS_j.}\| \geq \|\beta'_{iS_j.}\|$ for all i, then $\|\beta_{.S_j.}\| \geq \|\beta'_{.S_j.}\|$. □

The above proposition formalizes our intuition that orthogonality of X lowers $l^*(X)$ over non-orthogonality. We now show a similar result for the somewhat less intuitive heuristic that dictionary functions whose gradient fields are constant length will be favored over those which are non-constant. Since the result on orthogonality holds regardless of length, we need only consider the case where the component vectors in our sets of vector fields are mutually orthogonal at each data point, but not necessarily of norm 1. Note that were they not orthogonal, making them so would also reduce l^*.

Lemma 47. *Let $X'_{..S}$ be a set of vector fields $X'_{..S_j}$ mutually orthogonal at every data point i. Let $X''_{..S}$ be a set of vector fields $X''_{..S_j}$ mutually orthogonal at every data point i, and $\|X''_{i.S_j}\| = c_j$ for all $i \in [n]$. Then $\tilde{l}^*(X'_{..S}) \geq \tilde{l}^*(X''_{..S})$.*

Proof. Given indices i and j, let $c_{ij} = \|\tilde{X}_{iS_j.}\|$. Then, by Proposition 43, we can rotate $\tilde{X}_{i.S_j}$ so that

$$\beta'_{iS_jk} = \begin{cases} c_{ij}^{-1} \text{ if } j = k \\ 0 \text{ otherwise.} \end{cases} \tag{5.20}$$

Thus,

$$\|\beta'_{.S_j.}\|_2 = \|[c_{1j}^{-1} \dots c_{nj}^{-1}]^T\|_2. \tag{5.21}$$

On the other had, for $X''_{..S}$, the normalization guarantees that $\|X''_{i.S_j}\| = 1$, so

$$\|\beta''_{.S_j.}\|_2 = \|[1\dots 1]^T\| = \sqrt{n}. \tag{5.22}$$

We therefore want to show that $\|\beta'_j\|_2 \geq \sqrt{n}$.

Let $c_{ij} = \|X'_{i.j}\|_2$. To do this, we apply an iterative procedure, whereby pairs of vectors $\tilde{X}'_{i.S_j}$ and $\tilde{X}'_{i'.S_j}$ are perturbed to become more constant in length. We proceed in a pairwise manner, taking pairs of vector, one of which has norm greater than 1 and one which has norm less than 1, and transferring length from the longer to the shorter. As we have assumed orthogonality of the vectors in $X'_{i.S}$ for each i, we can perform these perturbations solely w.r.t. the scalars c_{ij}.

Dropping the index j for convenience, without loss of generality, let $c_1 < 1$ and $c_2 > 1$. Let $X^*_{..S_j}$,$\beta^*_{.S_j.}$, c_1^* and c_2^* generically refer to $X_{..S_j}$,$\beta_{.S_j.}$, c_1 and c_2, respectively, after a single step of this algorithm. Note that $\|\beta^*_{.S_j.}\|_2 \leq \|\beta'_{.S_j}\|_2$ implies Lemma 47, since we can iteratively make vector more constant in length until they are all unit length. There are two cases of interest that lead to the same analysis.

- Case 1: $|1 - c_1| < |1 - c_2|$. In this case, let $c_1^* = 1$ and $c_2^* = c_1^2 + c_2^2 - 1$. Thus, $\|[c_1^*, \dots c_n^*]^T\| = \sqrt{n}$, while $\|\beta^*_{.S_j.}\| = \sqrt{1 + \frac{1}{c_1^2 - c_2^2 - 1} + \sum_{i=3}^n c_i^{-2}}$.

- Case 2: $|1 - c_1| \geq |1 - c_2|$. In this case, let $c_2^* = 1$ and $c_1^* = c_1^2 + c_2^2 - 1$. Thus, $\|[c_1^*, \dots c_n^*]^T\| = \sqrt{n}$, while $\|\beta^*_{.S_j.}\| = \sqrt{\frac{1}{c_1^2 - c_2^2 - 1} + 1 + \sum_{i=3}^n c_i^{-2}}$.

Since $\|\beta'_{.S_j}\|_2 = \sqrt{\frac{1}{c_1^2} + \frac{1}{c_2^2} + \sum_{i=3}^n c_i^{-2}}$, this amounts to the claims

$$\frac{1}{c_1^2 + c_2^2 - 1} \leq \frac{1}{c_1^2} + \frac{1}{c_2^2} - 1 \tag{5.23}$$

$$\frac{1}{c_1^2 + c_2^2 - 1} \leq -\frac{c_1^2 c_2^2 - c_1^2 - c_2^2}{c_1^2 c_2^2} \tag{5.24}$$

$$\frac{c_1^2 c_2^2}{c_1^2 c_2^2 (c_1^2 + c_2^2 - 1)} \leq -\frac{(c_1^2 + c_2^2 - 1)(c_1^2 c_2^2 - c_1^2 - c_2^2)}{c_1^2 c_2^2 (c_1^2 + c_2^2 - 1)} \tag{5.25}$$

$$\frac{c_1^2 c_2^2 + (c_1^2 + c_2^2 - 1)(c_1^2 c_2^2 - c_1^2 - c_2^2)}{c_1^2 c_2^2 (c_1^2 + c_2^2 - 1)} \leq 0. \tag{5.26}$$

At this point we can remove the denominator, which will always be positive by the assumptions that $c_1 > 0, c_2 > 1$. Thus we continue

$$c_1^2 c_2^2 + (c_1^2 + c_2^2 - 1)(c_1^2 c_2^2 - c_1^2 - c_2^2) \leq 0 \tag{5.27}$$

$$c_1^2 c_2^2 + c_1^4 c_2^2 - c_1^4 - c_1^1 c_2^2 + c_1^1 c_2^4 - c_1^2 c_2^2 - c_2^4 - c_1^2 c_2^2 + c_1^2 + c_2^2 \leq 0 \tag{5.28}$$

$$c_1^4 c_2^2 - c_1^4 - 2c_1^1 c_2^2 + c_1^1 c_2^4 - c_2^4 + c_1^2 + c_2^2 \leq 0 \tag{5.29}$$

$$c_1^4 c_2^2 - c_1^4 - c_1^2 c_2^2 + c_1^2 + c_1^2 c_2^4 - c_1^2 c_2^2 - c_2^4 + c_2^2 \leq 0 \tag{5.30}$$

$$(c_1^2 c_2^2 - c_1^2 - c_2^2 + 1)(c_1^2 + c_2^2) \leq 0 \tag{5.31}$$

$$(c_1^2 - 1)(c_2^2 - 1)(c_1^2 + c_2^2) \leq 0 \tag{5.32}$$

$$(c_1 - 1)(c_1 + 1)(c_2 - 1)(c_2 + 1)(c_1^2 + c_2^2) \leq 0. \tag{5.33}$$

This condition is satisfied by the assumption that $c_1 > 0, c_2 > 1$. □

Together, Lemmas 46 and 47 show Proposition 44. This is because we can take any $X_{.S.}$ and orthogonalize the vectors $X_{i.S}$ at each data point i to get $X'_{..S}$ such that $\tilde{l}^*(X_{..S}) \geq \tilde{l}^*(X'_{..S})$. Then, for each j, we can make the vectors $\|X'_{i.S_j}\|$ more constant in length w.r.t i to get $X''_{..S}$ such that $\tilde{l}^*(X_{..S}) \geq \tilde{l}^*(X'_{..S})$. We now attend to the relation of this proposition to the tangent spaces of a manifold. □

5.4.3 The preference for tangency

We introduce a variant of $\tilde{l}^*(X)$ that is with respect to the tangent space of a d dimensional manifold $\mathcal{M}$, and show that it is minimized by X that are not only orthogonal and evenly varying, but tangent to $\mathcal{M}$. Suppose we are given vector fields $X \in \mathbb{R}^{n \times D \times p}$ as well as tangent bases $T_{1:n}^{\mathcal{M}} \in \mathbb{R}^{n \times D \times d}$. Denote $X_M := [T_i^{\mathcal{M}^T} X_{ij.}^T \; j \in 1 \ldots p, i \in 1 \ldots n] \in \mathbb{R}^{n \times p \times d}$. Then define a manifold specific loss

$$l(T_{1:n}^{\mathcal{M}}, X) := l^*(\tilde{X}_M),$$

where normalization is applied prior to projection. The next proposition formalizes the intuitive statement that vectors which are more tangent to $\mathcal{M}$ will be have longer length after projection, and therefore smaller loss.

Lemma 48. *Suppose we are given data ξ_i sampled from a d dimensional smooth manifold $\mathcal{M}$, tangent spaces $T_{1:n}^{\mathcal{M}}$, and vector fields $X \in \mathbb{R}^{n \times D \times p}$ that includes a subset $X_{..S^+}$ indexed by a d-subset $S^+ \subset [p]$ such that, given a $i \in [n]$, for each j in S_j, $X_{i.S_j{}^+}$ are mutually orthogonal, constant length, and tangent to $\mathcal{M}$ at each point ξ_i. That is, for all i, we have $X_{i.S_j^+}^T X_{i.S_{j'}^+} = 0$ for all $j \neq j' \in [d]$, $\|X_{i.S_j^+}\| = c_j$, and $\|T_i^{\mathcal{M}^T} X_{i.j}\| = \|X_{i.j}\|$. Then*

$$S^+ = \arg\min_{S \in [p]_d} l(T_{1:n}^{\mathcal{M}}, X_{..S}) \tag{5.34}$$

Proof. $\|T_i^{\mathcal{M}^T} \tilde{X}_{i.j}\| \leq \|\tilde{X}_{i..}\|$ for all j, while $\|T_i^{\mathcal{M}^T} \tilde{X}_{i.j}^T\| = \|\tilde{X}_{i.j}^T\|$ for all $j \in S^+$. Thus, $l(T_{1:n}^{\mathcal{M}}, X_{..S^+}) = \tilde{l}^*(X_{..S^+})$, while $l(T_{1:n}^{\mathcal{M}}, X_{..S}) \geq \tilde{l}^*(X_{..S})$ for all other d-subsets S. Therefore, Proposition 44 implies the lemma. □

This proposition extends Proposition 44 and claims that projection after normalization favors vectors which are tangent to a manifold $\mathcal{M}$. This proposition characterizes the preprocessing and variable selection steps of TSBasisPursuit , albeit considering the input to TSBasisPursuit to be arbitrary vectors rather than gradients.

5.4.4 Selection of isometries

Lemma 48 has special relevance when these vector fields are gradient fields. This is because a d-coordinate isometric coordinate chart $[g^1 \ldots g^d]$ of $\mathcal{M}$ if and only if the gradient vectors of these coordinate functions form unitary matrices upon projection on $\mathcal{T}_\xi \mathcal{M}$. That is, as shown in Proposition 23, $D\tilde{g}^{S^+}(\xi)$ unitary implies that g^{S^+} is at least locally an isometry.

To extend Lemma 48 to this functional setting, we recall the finite-sample functional analogue of our vector-field normalization

$$\tilde{f} := \frac{f}{\sqrt{\frac{1}{n}\sum_{i=1}^n \|\nabla f(\xi_i)\|_2^2}} \tag{5.35}$$

i.e. Equation 3.20. Given a d-subset S, denote $Jg^S(\xi_{1:n}) \in \mathbb{R}^{n \times D \times d}$ to be the tensor of gradients $\nabla g^j(\xi_i)$ and $Dg^S(\xi_{1:n}) \in \mathbb{R}^{n \times d \times d}$ to be the tensor of gradients $\operatorname{grad}_{\mathcal{M}} g^j(\xi_i)$.

Proposition 49. *Suppose we are given a d dimensional smooth manifold $\mathcal{M}$ and a p function dictionary $\mathcal{G} : U \to \mathbb{R}^p$ containing d functions indexed by $S^+ \subset [p]$ for which $\tilde{g}^{S^+} = [\tilde{g}^{S_1^+}, \dots \tilde{g}^{S_d^+}]$ is an isometry of $\mathcal{M}$. Then*

$$S^+ = \arg\min_{S \in [p]_d} l(T_{1:n}^{\mathcal{M}}, Jg^S). \tag{5.36}$$

Proof. We show that $Jg^S(\xi_{1:n})$ satisfies the conditions of Lemma 48. By assumption, $\tilde{g}^{S^+}$ is an isometry of $\mathcal{M}$. Note that $J\tilde{g}^S(\xi_{1:n}) = \tilde{J}g^S(\xi_{1:n})$, where the latter is given by the vector field normalization Equation 5.12. We show that $Jg^{S_j^+}(\xi_i) = \nabla_\xi g^{S_j^+}(\xi_i)$ are mutually orthogonal across j, constant length across i, and always tangent to M.

We begin by showing that $\nabla g^{S_j^+}(\xi_i)$ are tangent to $\mathcal{M}$. The assumption that $\tilde{g}^{S^+}$ is an isometry of $\mathcal{M}$ implies that $D\tilde{g}^{S^+}(\xi_i)$ is unitary. Therefore, $\| \operatorname{grad}_{T_i^{\mathcal{M}}} \tilde{g}^{S_j^+}(\xi_i)\| = 1$, so, since $n_{2,2}(Jg^j(\xi_{1:n})) = \sqrt{n}$ and and $\| \operatorname{grad}_{T_i^M} g^j(\xi_i)\| \leq \|\nabla \tilde{g}^j(\xi_i)\|$ for all j , $\|\nabla \tilde{g}^{S_j^+}(\xi_i)\| = 1$ as well, and so $\nabla g^{S_j^+}(\xi_i)$ are necessarily tangent to $\mathcal{M}$.

Since $\nabla g^{S_j^+}(\xi_i)$ are tangent to $\mathcal{M}$, and $\nabla \tilde{g}^{S_j^+}(\xi_i)$ are constant length, $\nabla g^{S_j^+}(\xi_i)$ are constant length as well. This is because the normalization $\nabla \tilde{g}^{S_j^+}(\xi_i) = \frac{\nabla g^{S_j^+}(\xi_i)}{\sqrt{\frac{1}{n}\sum_{i=1}^n \|\nabla g^{S_j^+}(\xi_i)\|_2^2}}$ rescales all $\nabla g^{S_j^+}$ at each data point ξ_i by the same constant.

Finally, we show mutual orthogonality across j at each i. The assumption that $\tilde{g}^{S_j^+}$ is an isometry of $\mathcal{M}$ implies that $D\tilde{g}^{S_j^+}(\xi_i)$ is unitary. Therefore, $(\operatorname{grad}_{T_i^{\mathcal{M}}} g^{S_j^+}(\xi_i))^T \operatorname{grad}_{T_i^{\mathcal{M}}} \tilde{g}^{S_{j'}^+}(\xi_i) = 0$ for all $j \neq j'$. Since $\operatorname{grad}_{T_i^{\mathcal{M}}} g^{S_j^+} = T_i^{\mathcal{M}^T} \nabla g^{S_j^+}(\xi_i)$, this implies that $(\nabla g^{S_j^+}(\xi_i))^T \nabla g^{S_{j'}^+}(\xi_i) = 0$. □

Proposition 49 says that, if a dictionary contains functions which are component functions of an isometric coordinate chart modulo rescaling, then they will be selected by TSBasisPursuit when the tangent spaces $T_i^{\mathcal{M}}$ are estimated without noise. Note that this also assumes that data ξ_i are sampled from $\mathcal{M}$ without noise. Beyond this potentially unrealistic statistical assumption, we also note two other oversights. First, isometric coordinate charts are not necessarily global isometries. As mentioned in Chapter 3, we cannot find embeddings of topologically non-trivial d dimensional manifolds in $\mathbb{R}^d$, and so we can only hope to find isometries away from sets of measure 0. Second, only regions of manifolds witjh zero Gaussian curvature will admit a local isometry into d dimensions Kohli et al. (109). For example, we can find an almost everywhere isometric coordinate chart for the flat torus in $\mathbb{R}^4$, but cannot for the torus of revolution in $\mathbb{R}^3$. In cases like the latter, the loss $\tilde{l}^*$ may be compared with the isometric optimimum of $d\sqrt{n}$ to measure the degree of non-isometry.

5.5 Two-stage basis pursuit

As discussed in Chapters 3 and 4, problems with shrinkage including variable selection inconsistency at large λ, as well as the desirable properties of an intermediate value, are well-established in the support recovery and sparse coding literature (166; 132; 88; 33; 116; 81). On the other hand, the MBasisPursuit and TSBasisPursuit algorithms are computationally intractable for large dictionaries and high dimensional manifolds. In this Section, we heuristically adapt the convex method to respond to this problem.

This adaptation is motivated in particular by the ideas in Meinshausen (132); Meinshausen and Yu (134). The approach of Meinshausen (132) is to first run a Lasso at an intermediate tuning parameter value, and then run a second Lasso on the surviving coefficients. In our approach, we first minimize $J_{M,\lambda=\lambda_{\max/2}}(X,Y)$ or $J_{TS,\lambda=\lambda_{\max/2}}(X)$ to obtain a vastly reduced dictionary size, and then minimize $J_{M,C=0}(X,Y)$ or $J_{TS,C=0}(X)$ to select from supports selected at the intermediate λ value. The λ at which these functions are obtained is somewhat arbitrary, but theoretical arguments using random Gaussian noise provide arguments in favor of $\lambda > O(\log p)$ (166). We experimentally find relatively wide regions of relatively low cardinality, and substantial improvements in the combinatorial loss with minimal computational burden at $\lambda = \lambda_{\max}/2$.

Algorithm 8 TwoStageMBP(Dataset $\mathcal{D}$, dictionary $\mathcal{G}$, embedding coordinates $\phi(\mathcal{D})$, intrinsic dimension d, kernel bandwidth ϵ, neighborhood cutoff size r)

```
Get $\lambda_{\max}$ using Equation (3.22)
S' = ManifoldLasso($\mathcal{D}, \mathcal{G}, \phi(\mathcal{D}), d, \epsilon, r, \lambda_{max}/2$)
S = MBasisPursuit($\mathcal{D}, g^{S'}, \phi(\mathcal{D}), d, \epsilon, r$)
Output $S$
```

Algorithm 9 TwoStageTSBP(Dataset $\mathcal{D}$, dictionary $\mathcal{G}$, intrinsic dimension d, kernel bandwidth ϵ, neighborhood cutoff size r)

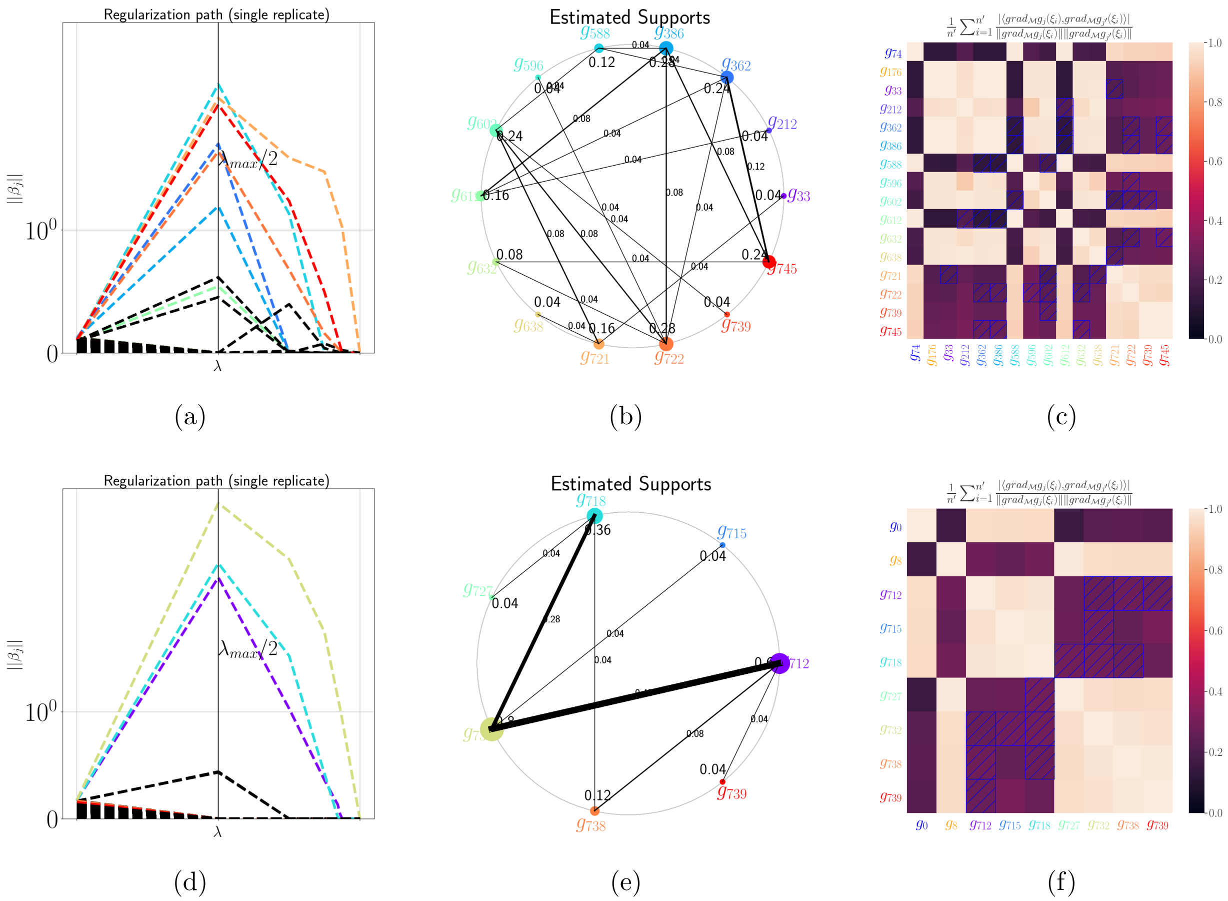
```
Get $\lambda_{\max}$ using Equation (4.17)
S' = TSLasso($\mathcal{D}, \mathcal{G}, d, \epsilon, r, \lambda_{max}/2$)
S = TSBasisPursuit($\mathcal{D}, g^{S'}, d, \epsilon, r$)
Output $S$
```

5.6 Two-stage Experiments

We provide experimental results that illustrate the performance of the two-stage methods TwoStageMBP and TwoStageTSBP.

Two-stage Manifold Basis Pursuit We first apply the TwoStageMBP to obtain highly orthogonal solutions at reduced computational cost in the experiments from Section 3.8. In the plotted replicate from Figure 3.10b, 10 out of 756 covariate functions are obtained, and the second stage of the two-stage solution is able to run rapidly. Figure 5.1 gives support recovery results for **Ethanol** and **Malonaldehyde** using the two-stage method for ManifoldLasso. These solutions are colinear with true support - $.97 \pm .03$ for **Ethanol** and $.96 \pm .01$ for **Malonaldehyde**.

Visual inspection of the colored embeddings for **Ethanol** in Figures 5.3 and 5.2 also confirms that these match our visual intuition of orthogonally varying torsions. Selected pairs of functions are in general more visually orthogonal than found using ManifoldLasso, particularly **Ethanol** . Examining the identities of these torsions with the molecules in Figures 3.1b and 3.1c, these tend to be hydrogen-hydrogen torsions abutting the true central bonds. Together, these experiments show that on noisy, large p problems, and with massive violations of the incoherence conditions, ManifoldLasso, while sometimes not successful on its own, can be made robust by off-the-shelf methods.



(a) (b) (c)

(d) (e) (f)

Figure 5.1: TwoStageMBP results for **Ethanol** and **Malonaldehyde**, respectively, with dictionaries given by all possible torsions. Figures 5.1a and 5.1d show individual replicates, with intermediate tuning parameter value $\lambda_{\max}/2$. Colors are plotted for functions selected by subsequent combinatorial analysis. Figure 5.1c and 3.10g shows support recoveries given by subset selection using group lasso at $\lambda_{\max}/2$ followed by TwoStageMBP over $\omega = 25$ different replications. Figure 3.10d and 3.10h and shows mean cosine collinearity of selected supports. $g^{74,176}$ and $g^{0,8}$ are representative torsions from the true support, while the others are selected in any replicate. Pairs that are selected in any replicate are marked with a blue box. $(n = 100, p = 756)$

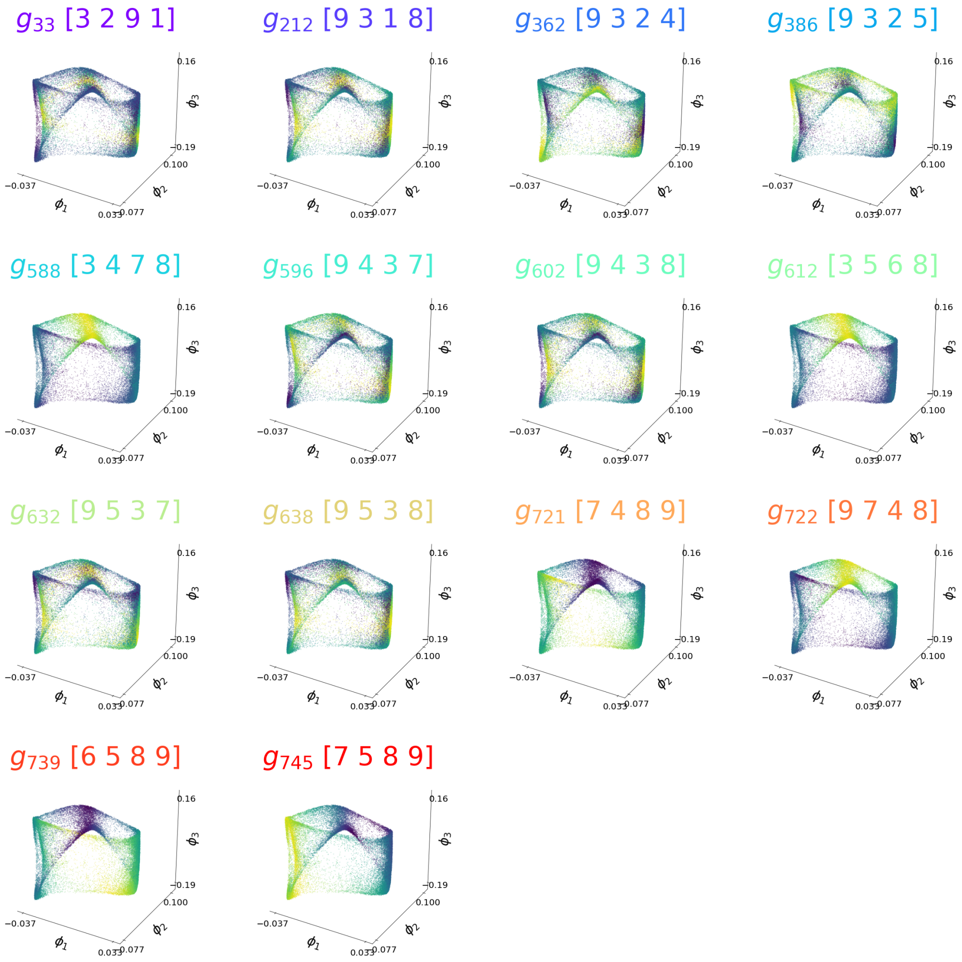


Figure 5.2: **Ethanol** support using basis pursuit on superset obtained using ManifoldLasso. Colors should be compared with Figure 5.1.

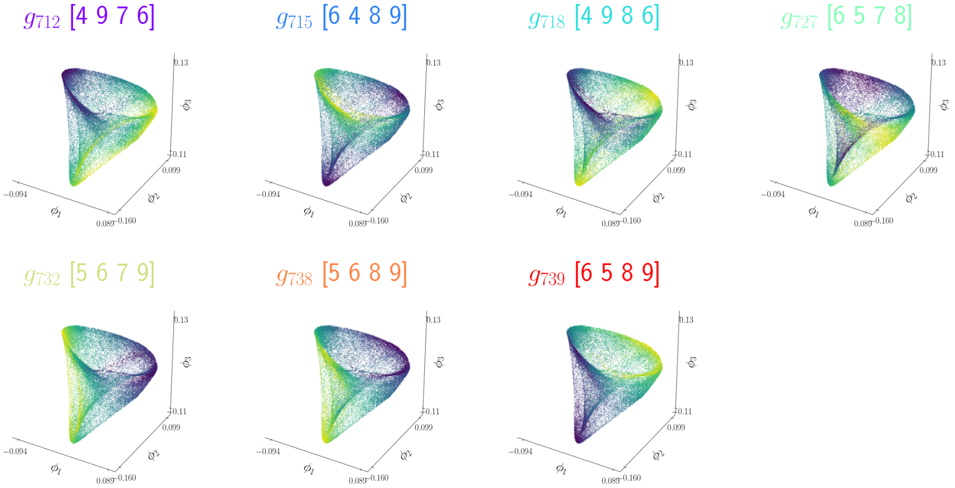
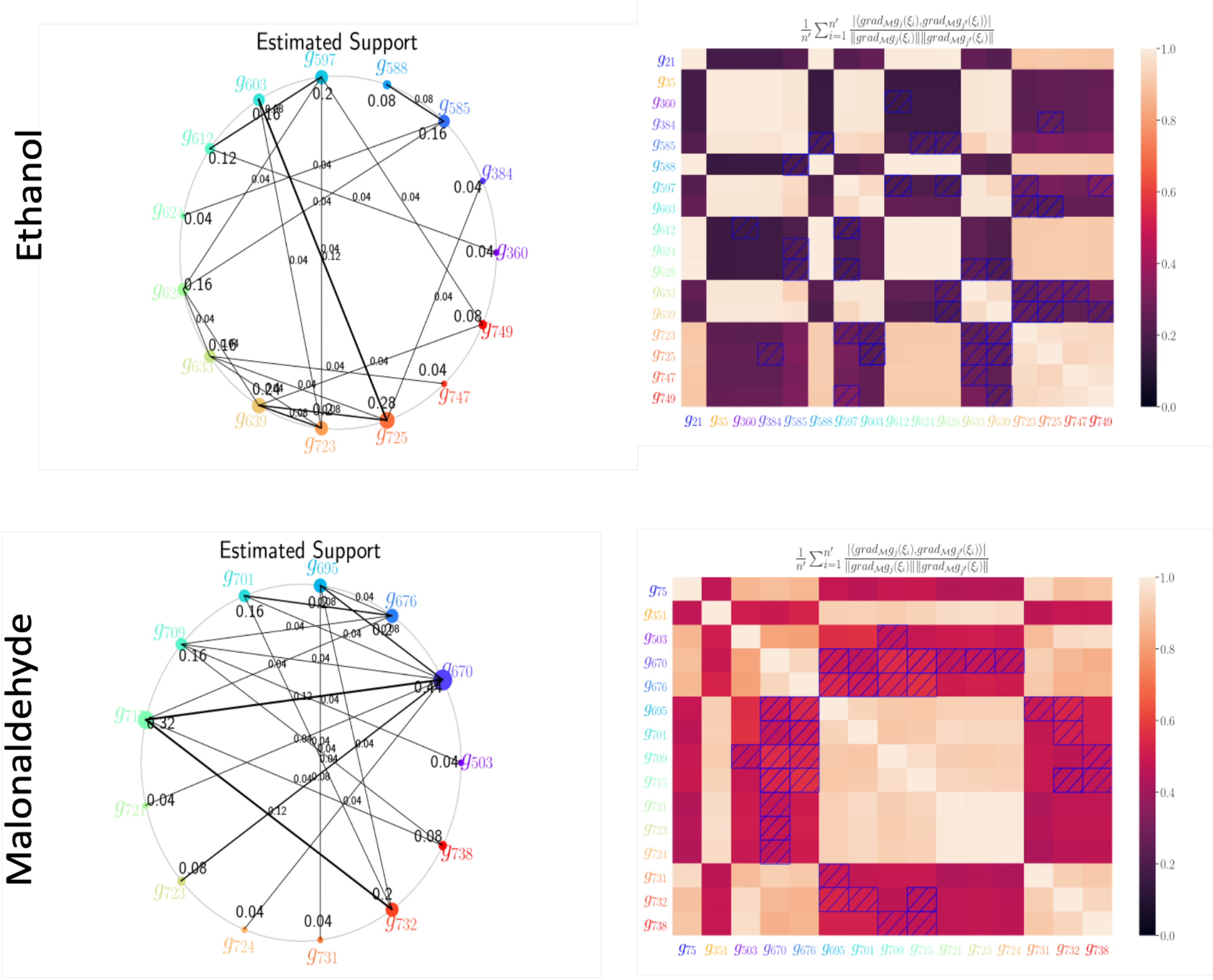


Figure 5.3: **Malonaldehyde** support using basis pursuit on superset obtained using ManifoldLasso. Colors should be compared with Figure 5.1.

Two-stage Tangent Space Basis Pursuit We repeat the same analyses using the two-stage combination of Tangent Space Lasso and Basis Pursuit. These results also show that the two-stage approach, is preferentially effective for identifying orthogonal supports. Note that the change in collinearities for **Malonaldehyde** compared with Chapter 3 are due to the different bandwidth parameter ϵ. Similarly, g^{21}, g^{35} and g^{75}, g^{351} and are now ground truth for **Ethanol** and **Malonaldehyde** , respectively.



(a)

Figure 5.4: TwoStageTSBP results using full dictionaries ($n = 100, p = 756$).

5.7 *Discussion*

The methods introduced in this chapter build on the ManifoldLasso and TSLasso algorithms to deal with challenges posed by real data. In particular, we introduce a more specific success criteria for support recovery, and combine it with the previous algorithms to create a two-stage approach. Several areas for potential improvement exist. First, the speed of the combinatorial problem could also be accelerated by computing matrix inverses in the $d = 2$ case. Second, as mentioned in Chapter 4, it seems plausible that our basis pursuit methods could be shown to achieve a polynomial convergence rate. Since the loss of every set S is an average of squares over data points, better-than-random estimates of the tangent space made with polynomial samples and assumptions on global regularity would enable a weak learnability argument based on the Hoeffding bound. Third, characterizing the reasons for retention of functions as the tuning parameter approaches 0 is also of interest. The number of retained functions in $\mathcal{G}$, while greater than d, remains low. Due to our normalization, it is possible that these functions simply large gradients at a single data point.

The Tangent Space Basis Pursuit problem is one of many methods for identifying isometries (109; 129), but it has the advantage of being a convex program. One important question is whether the global basis pursuit minimizer is retained within the pruned dictionary set. On the other hand, our method finds the most isometric global parameterization, and so is not suited for the local minimum-distortion parameterization problem (108), in which functions from a dictionary are selected from in order to find the most isometric parameterization at various points ξ_i. We suggest two novel alternate objectives for this problem. First,

$$\arg\min_{S \in [p]_d} \| \log \operatorname{spec}_d(\tilde{X}_{i.S}) \|_2^2, \tag{5.37}$$

where spec_d are the d eigenvalues of $X_{i.S}$. This loss will be minimized by an isometry, since all singular values are 1. Another alternative replaces $X_{i.j}$ with

$$\exp(-(\log \|X_{i.j}\|_2)^2) \frac{X_{i.j}}{\|X_{i.j}\|} \tag{5.38}$$

before running a group lasso on each $X_{i.j}$ independently. Without the need to induced groupwise sparsity on a regularization path, a method-of-frames objective in the combinatorial step may also be minimized by an isometry (166). Finding isometries from dictionaries with $s > d$ is another possible area for research, as are the situations such as fitting densities for which a non-isometric embedding may be desired.

Chapter 6

GRADIENT ESTIMATION USING LEARNED TANGENT SPACES

The previous chapters have introduced a set of algorithms for interpretable unsupervised learning based on the idea of differential composition for which estimates of the gradients of functions w.r.t. the data manifold $\mathcal{M}$ play a central role. The quality of estimates made through either local linear regression or projection of an analytic gradient onto the manifold directly depends on the quality of the estimates of the associated tangent spaces. Unfortunately, tangent space estimators are often challenged by even limited amounts of noise. In this chapter, we examine the relationship between the tangent space, local linear regression, and the learned embedding map in greater detail, and propose a new method for tangent space estimation.

Our method is to estimate the these tangent spaces with respect to a latent space determined by a manifold learning algorithm. In particular, we use the gradients of an embedding map to define this reduced dimensional space, and give a sufficient condition on the embedding map for this space to be the correct tangent space. This method can be used in conjunction with many existing gradient estimators. We demonstrate that this method improves accuracy of tangent space and gradient estimation in the presence of noise in simulations, and apply it to data from the Sloan Digital Sky Survey (26).

6.1 Introduction

Estimates of the gradient of a function f defined on a manifold $\mathcal{M} \subset \mathbb{R}^D$ are fundamental to a variety of statistical tasks, but determining the correct estimator for a particular problem remains a practical and theoretical challenge (139; 17; 45; 125). At a point $\xi \in \mathcal{M}$, the gradient with respect to the manifold $\mathrm{grad}_{\mathcal{M}} f(\xi)$ lies in the tangent space $\mathcal{T}_{\xi}\mathcal{M}$, and so tangent space estimation is often performed as an intermediate step of gradient estimation. Having a lower-dimensional projection enables performing local linear regression in rank-deficient tangent spaces, and tangent-based regularization for gradient estimation using local linear regression has been shown to have desirable statistical properties (17; 45). Even higher-order gradient estimators such as those based on fitting smooth functions like splines suffer when noise in observation of f is correlated with noise in observation of ξ (125). Therefore, achieving better tangent space estimation in the presence of noise is an active area of research.

Existing approaches to tangent space estimation vary in their applicability to noisy data. The manifold hypothesis describes data either distributed on (17; 45; 43; 122) or near (75; 6; 155; 67) a low-dimensional surface. This difference has an effect on tangent space estimation. The desirable finite sample and asymptotic properties of local PCA in the former case are shown in Aamari and Levrard (6), but their analyses rely on shrinking the localization window as sample size grows, and so depend asymptotically on the infinitesimal nearby structure of the data manifold. Since this requires distinguishing between directions of local and global variability, higher-dimensional noise can be especially problematic. Thus, the theoretical justifications of a shrinking window size are often subordinated to the empirical need for this window to be large enough to capture global structure rather than noise. This has motivated alternatives like smoothing of locally estimated tangent spaces (155).

An alternative line of reasoning appears elsewhere in the machine learning literature. In Chapter 3, we used manifold learning algorithms to identify slow modes of molecular dynamics data. We showed that the representations learned by Diffusion Maps corresponded to bond torsions with rotational degrees of freedom. However, these representations do not represent the entirety of the molecular dynamics - Diffusion Maps filters them out from faster modes of motion. This preservation of large-scale structure in the top eigenfunctions of the Laplacian is evident in a variety of settings (93).

This chapter examines the effect of this noise-reducing property on tangent space estimation. In our method, we use the an embedding algorithm to learn a denoised representation of our data, and use the estimated gradients of the learned coordinates to estimate the tangent space $\mathcal{T}_\xi\mathcal{M}$. We give background on our statistical set-up in Section 6.2. Section 6.3 then introduces our embedding-based tangent space estimator. A sufficient condition on the embedding map ϕ for its use in this estimator is given in Section 6.4. In Section 6.5, we estimate $\operatorname{grad}_{\mathcal{M}} f$ using these tangent spaces by local linear regression. Experimental results are given in Section 6.6.

6.2 Background

The mathematical framework introduced in this chapter explicitly models noise. We assume the existence of a smooth [1] compact d_M dimensional manifold $\mathcal{M}$ with **reach** R that we call the **data manifold** that is observed in the feature space $\mathbb{R}^D$. We also assume that $\mathcal{M}$ is a non-trivial submanifold of a d_E dimensional **noise manifold** $\mathcal{E}$, which is itself a compact smooth submanifold of the feature space $\mathbb{R}^D$.

The data and noise manifolds have the following relation. For every ξ, we cannot directly observe ξ, but rather observe a noisy realization $\xi_\epsilon = \xi + \epsilon_\xi$ where $\epsilon_\xi \in B_{\mathcal{M},\mathcal{E}}(\xi, R')$, a ball of radius $R' < R$ and dimension $d_E - d_M$ in $\mathcal{N}_\xi(\mathcal{M}, \mathcal{E})$ - the **normal space** of $\mathcal{M}$ in $\mathcal{E}$ at ξ. Thus, $\mathcal{E}$ is contained within a **tubular neighborhood** of $\mathcal{M}$, i.e. $\mathcal{E} \subseteq \cup_{\xi\in\mathcal{M}} B_{\mathcal{M},\mathcal{E}}(\xi, R')$. Furthermore, $\mathcal{M}$ contains a d_M ball of radius larger than $4R'$; in other words, $\mathcal{E}$ is thinner in the normal direction to $\mathcal{M}$. More information on fibrations is given in Section 2.2.

Problem statement Assume that we have access to covariate and response observations of n data points $\xi_{\epsilon,i} \in \mathbb{R}^D$ sampled from $\mathcal{E}$ and $f_{\epsilon,i} \in \mathbb{R}$, respectively, where $\xi_{\epsilon,i} = \xi_i + \epsilon_{\xi,i}$, $f_{\epsilon,i} = f_i(\xi_i) + \epsilon_i$ with $\mathbb{E}(\epsilon_\xi) = \vec{0}$, $\mathbb{E}(\epsilon) = 0$, and $\xi_i \in \mathcal{M}$. Let $\mathcal{D} = \xi_{\epsilon,1:n} \in \mathbb{R}^{n\times D}$. We assume that we are given the intrinsic dimensions d_M of $\mathcal{M}$ and d_E of $\mathcal{E}$, as well as an embedding dimension m of $\mathcal{B} := \phi(\mathcal{M})$. We assume that a manifold learning algorithm EMBED is available, which outputs an embedding $\phi(\mathcal{D}) := \phi(\xi_{\epsilon,1:n})$ of the data in $\mathbb{R}^{n\times m}$ satisfying certain conditions. We also assume that ϵ_ξ and ϵ are possibly dependent, and ϵ_ξ is allowed to

[1] Here we consider a smooth manifold to be of class $\mathcal{C}^3$ and a smooth function of class $\mathcal{C}^1$.

depend on ξ. Our goal is to estimate $\operatorname{grad}_{\mathcal{M}} f$, the gradient of f w.r.t. $\mathcal{M}$.

We tackle two intermediary problems. First, we provide an estimator for $\mathcal{T}_{\xi_i}\mathcal{M}$ based on a learned embedding map ϕ of $\mathcal{D}$, and give a sufficient condition on this map for our estimator to give the tangent space. Second, we use this estimate in a subsequent estimation of $\operatorname{grad}_{\mathcal{M}} f$ from $\mathcal{D}$ and $f_{\epsilon,i} = f_i + \epsilon_i$, where $\operatorname{Cov}(\epsilon, \epsilon_\xi) \neq 0$. We can extend this function to $\mathcal{E}$ as $f : \mathcal{E} \to \mathbb{R}$ given by $f(\xi_\epsilon) = f(\xi)$. Thus, $f_\epsilon(\xi_\epsilon) = f(\xi) + \epsilon$.

The rest of this section contains concepts from differential geometry and classical regression that introduce our combined estimator. For more details the reader should consult (118) for the former and (154) for the latter.

6.2.1 Manifold learning and fibered manifolds

In this Chapter we explain the use of manifold learning for estimating tangent spaces $\mathcal{T}_{\xi_i}\mathcal{M}$, and subsequent use of these tangent spaces in estimation of $\operatorname{grad}_M f(\xi_i)$. This application uses the definition of fibered manifold from Chapter 2 - a triple $(\mathcal{E}, \mathcal{B}, \phi)$, representing total space, base space, and projection, respectively, related by $\phi : \mathcal{E} \to \mathcal{B}$ where ϕ is a surjective submersion - to model the removal of noise by a manifold learning algorithm. For a pair of submanifolds $\mathcal{M}$ and $\mathcal{E}$ such that $\mathcal{M} \subset \mathcal{E} \subset \mathbb{R}^D$ as above, we define a **Manifold Learning** (ML) map to be a map $\phi : \mathcal{E} \to \mathbb{R}^m$ so that $(\mathcal{E}, \phi(\mathcal{M}), \phi)$ is a submersion. Thus, in our setting, $\mathcal{B} = \phi(\mathcal{M})$. Since our goal is to preserve the structure of the underlying manifold $\mathcal{M}$, it is natural to require that the restriction $\phi|_{\mathcal{M}} : \mathcal{M} \to \mathcal{B}$ is a diffeomorphism. Points in $\mathcal{E} \setminus \mathcal{M}$ are mapped by ϕ to $\mathcal{B}$; hence, regarding ϕ as a submersion rather than a diffeomorphism explicitly models the removal of noise.

The mapping ϕ induces a **fibered structure** on the tangent bundle of noise manifold $\mathcal{E}$, expressed as

$$\mathcal{T}_{\xi_\epsilon}\mathcal{E} = \mathcal{H}^{\phi}_{\xi_\epsilon}\mathcal{E} \times \mathcal{V}^{\phi}_{\xi_\epsilon}\mathcal{E}, \tag{6.1}$$

where $\mathcal{H}^{\phi}_{\xi_\epsilon}\mathcal{E}$ and $\mathcal{V}^{\phi}_{\xi_\epsilon}\mathcal{E}$ are known as the **horizontal** and **vertical spaces** in $\mathcal{T}_{\xi_\epsilon}\mathcal{E}$ induced by the mapping ϕ. The vertical space $\mathcal{V}^{\phi}_{\xi_\epsilon}\mathcal{E} := \ker(D\phi(\xi_\epsilon))$, while the horizontal space

$\mathcal{H}^{\phi}_{\xi_\epsilon}\mathcal{E} := \mathcal{T}_{\xi_\epsilon}\mathcal{E}/\mathcal{V}^{\phi}_{\xi_\epsilon}\mathcal{E}$. Note that in this chapter we write $D\phi$ - the differential of ϕ - as $D\phi^m_D := [\nabla_E \phi_k(\xi)\ k \in 1 \ldots m]^T \in \mathbb{R}^{m\times D}$ in order to indicate the number of coordinates of our feature and embedding spaces.

6.2.2 Estimation of gradients from samples

Our goal is to estimate $\operatorname{grad}_M f$. For convenience, we also define $\nabla_M f(\xi) := T^{\mathcal{M}}_{\xi} T^{\mathcal{M}^T}_{\xi} \nabla_{\xi} f$, which is $\operatorname{grad}_{\mathcal{M}} f$ expressed in coordinates of $\mathbb{R}^D$. This $\nabla_M f(\xi)$ is invariant to choice of $T^{\mathcal{M}}_{\xi}$.

A standard estimator for $\operatorname{grad}_M f(\xi_i)$ is implied by Equation 2.9 and Proposition 33.

$$f_{\epsilon,i'} - f_{\epsilon,i} \sim \langle \xi_{\epsilon,i'} - \xi_{\epsilon,i}, \operatorname{grad}_{\mathcal{M}} f(\xi)\rangle \tag{6.2}$$

where $i' \in \mathcal{N}_i$ is within a a sufficiently small locally linear neighborhood of ξ_i (140). Thus, given local covariate and response matrices

$$\delta^{\xi_\epsilon}_i := [\xi_{\epsilon,i'} - \xi_i : i' \in \mathcal{N}_i] \in \mathbb{R}^{|\mathcal{N}_i| \times D} \text{ and} \tag{6.3}$$

$$\delta^{f_\epsilon}_i := [f_{\epsilon,i'} - f_{\epsilon,i} : i' \in \mathcal{N}_i] \in \mathbb{R}^{|\mathcal{N}_i|} \tag{6.4}$$

Equation 6.2 then suggests an estimator

$$\widehat{\nabla_M f_\epsilon(\xi)} = \textsc{ols}(\delta^{\xi_\epsilon}_i, \delta^{f_\epsilon}_i). \tag{6.5}$$

In our manifold setting, the design matrix will typically be rank $d_E < D$. As an alternative approach to the implicit estimation of the tangent space occurring during pseudoinverse-based thresholding discussed in Section 2.4, we can apply the WLPCA algorithm from the previous chapters to estimate $T^{\mathcal{M}}_i$ prior to solving

$$\widehat{\operatorname{grad}_{T^{\mathcal{M}}_i}} f = \textsc{ols}(\delta^{\xi}_i T^{\mathcal{M}}_i, \delta^{f}_i). \tag{6.6}$$

Unfortunately, as shown by Proposition 2.13 in Section 2.4, the OLS estimator is not consistent when the covariates ξ are observed with error, even when the design matrix is full rank. Bias will result when $\operatorname{Cov}(\epsilon, \epsilon_\xi) \neq 0$, and even when ϵ_ξ, ξ, and ϵ are jointly independent, in which case it is called **attenuation bias**. The improvement of estimates made using local linear regression under correlated noise has motivated a multistage least squares estimators (15).

6.2.3 Estimation of $\mathcal{T}_{\xi_i}\mathcal{M}$

The tangent space $\mathcal{T}_{\xi_i}\mathcal{M}$ must be estimated so that the OLS design matrix is full rank. This step is essential for many gradient estimation algorithms (17). However, the local-PCA based approach TSLasso used throughout this thesis has problems with noise. For example, consistency requires noise tending to 0 as $n \to \infty$ (155; 6). In particular, the neighborhood radius parameter r must be shrunk to 0. Since this approach is not reasonable for i.i.d. noisy data, alternative approaches are required. One contrasting heuristic is to use a neighborhood radius larger than the magnitude of the observed noise (51).

Our approach takes inspiration from elsewhere in the linear regression literature. Slice inverse regression (15) and its local counterpart (192) propose methods whereby data are projected onto the span of regression coefficients. We will combine this idea with the globally-informed noise-reducing properties of manifold learning. In the next sections, we propose a suitable response variable and give a condition under which this procedure may be used to estimate $\mathcal{T}_{\xi_i}\mathcal{M}$.

6.3 Embedding-based tangent space estimation

In this section we show how an embedding map ϕ may be used to obtain estimates of $\mathcal{T}_{\xi_i}\mathcal{M}$ in the presence of noise. We estimate $H^{\phi}_{\xi_\epsilon}\mathcal{E}$, the horizontal space of the embedding map ϕ within the noise manifold $\mathcal{E}$. Section 6.4 then gives a condition under which $H^{\phi}_{\xi_\epsilon}\mathcal{E} = T_{\xi}\mathcal{M}$. This condition is that $\phi(\xi_\epsilon) = \phi(\text{cp}_{\mathcal{M}}(\xi_\epsilon)) = \phi(\xi)$ where $\text{cp}_{\mathcal{M}}$ is the **closest point** mapping from $\mathcal{E}$ to $\mathcal{M}$. This condition is proved in Section 6.4.

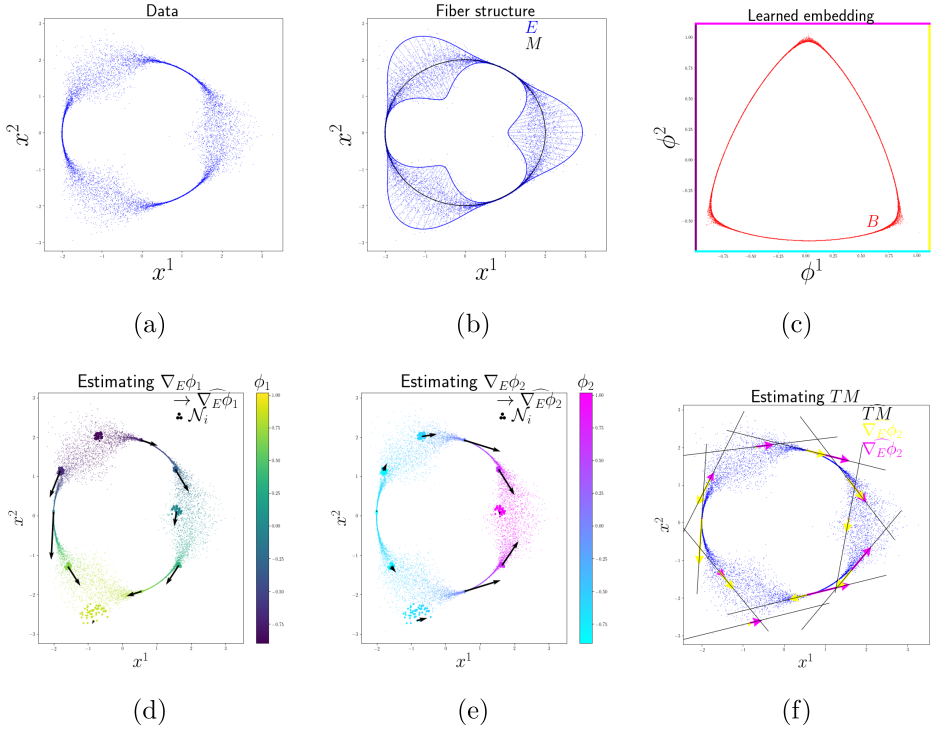
Given a data point $\xi_{\epsilon,i}$, the EmbedTS algorithm based on this idea takes as input local data matrices $\delta_i^{\xi_\epsilon}$, as well as local embedding matrices

$$\delta_i^{\phi} := [\phi(\xi_{\epsilon,i'}) - \phi(\xi_i) : i' \in \mathcal{N}_i] \in \mathbb{R}^{|\mathcal{N}_i| \times m}. \tag{6.7}$$

Step 1 computes localized geometric information necessary for estimating differential quantities. Step 2 then computes the tangent space of the noise manifold $\mathcal{E}$ necessary for a well-defined solution of OLS. Note that since this manifold contains the noise by definition, it is in a sense noiseless, and can be estimated at a minimax optimal rate by a local PCA algorithm (6). Step 3 then estimates the differential $D^m_D\phi$ of the embedding map ϕ in the original coordinates $\mathbb{R}^D$. Note that here we solve a multivariate OLS consisting of treating each response variable ϕ^k independently and concatenating the output coefficients. The resulting coefficients are an estimate of $\widehat{D\phi^m_D}$. Theoretically, $\widehat{D\phi^m_D}$ is rank d_M, and Step 5 extracts its d_M-dimensional principal subspace which represents the horizontal space of ϕ. The right singular vectors of $\widehat{D\phi^m_D}$ are then used to construct an estimate of the tangent space $T_i^{\mathcal{M}}$ in Step 4. Note that this final step assumes that our closest point condition Assumption 1 is satisfied.

```
Algorithm 10 EMBEDTS(Local data $\delta_i^{\xi_\epsilon}$, local embedding coordinates $\delta_i^{\phi}$, intrinsic dimensions $d_M$ and $d_E$, kernel bandwidth $\epsilon$)
1: Compute normalized Gaussian kernel row $K_{i,\mathcal{N}_i}$ using $\tilde{\delta}_i^{\xi_\epsilon}$ and $\epsilon$ (see Section 2.3)
2: Compute $\widehat{T_i^{\mathcal{E}}} \leftarrow \textsc{wlpca}(\delta_i^{\xi_\epsilon}, K_{i,\mathcal{N}_i}, d_E)$
3: Compute $\widehat{D\phi^m_D} \leftarrow \textsc{ols}(\delta_i^{\xi_\epsilon}\widehat{T_i^{\mathcal{E}}}, \delta_i^{\phi})\widehat{T_i^{\mathcal{E}}}^T \in \mathbb{R}^{D\times m}$
4: Compute $\widehat{T_i^{\mathcal{M}}}, \Sigma_i, V_i^T \leftarrow \textsc{svd}(\widehat{D\phi^m_D}, d_M)$
5: **Output** $\widehat{T_i^{\mathcal{M}}}$
```

$$\begin{tikzcd}
\mathbb{R}^p & \mathbb{R}^m \\
\cup & \cup \\
\mathcal{E} \arrow[r, "\phi"] & \mathcal{B} \\
\mathcal{M} \arrow[u, "+\epsilon_\xi"] \arrow[ur, "\phi_M"'] &
\end{tikzcd}$$



(a) (b) (c)

(d) (e) (f)

Figure 6.1: Our framework for tangent space estimation. Inclusions are shown in green, submersions in purple, and diffeomorphisms in yellow. Blue refers to our observables. Red refers to the output of a manifold-learning algorithm. $\mathbb{R}^D$ and $\mathbb{R}^m$ are the feature and embedding spaces, respectively. Our goal is the estimation of $\mathcal{T}_\xi \mathcal{M}$ at various data points ξ_ϵ drawn from $\mathcal{E}$. 6.1a - 6.1f Our tangent estimation strategy. 6.1a An example of data with multiscale non-i.i.d. noise. 6.1b Diagram of the manifold M and the noise manifold E. 6.1c The learned embedding. 6.1d, 6.1e Gradients of embedding coordinates estimated using local linear regression. 6.1f Our estimated tangent spaces.

To see the visual intuition of our method, we depict the steps in our estimator in Figures 6.1b - 6.1f. Figure 6.1a shows some data sampled from $\mathcal{E}$, while Figure 6.1b shows the true underlying tubular structure around the base space $\mathcal{M}$. In this depiction, we have relaxed the manifold structure of $\mathcal{E}$ to show multiscale noise. Figure 6.1c shows a learned embedding which is diffeomorphic to the circular base space. Figures 6.1d and 6.1e show estimated gradients of the coordinate functions of this embedding made w.r.t. $\mathcal{T}_{\xi_\epsilon}\mathcal{E}$ in the depicted neighborhoods. Note that these neighborhoods are smaller than the magnitude of the noise. Since ϕ is a diffeomorphism of the base space, at least one coordinate gradient must be non-zero at any given point. Figure 6.1f shows how these gradients are combined to show tangent spaces. Estimation of such tangent spaces given these small neighborhoods is a key advantage of our method.

6.4 Sufficient conditions

We provide a general property on the embedding map ϕ that implies that it is suitable for estimation of $\mathcal{T}_\xi\mathcal{M}$. Our main theoretical objective is to establish a correspondence between $\mathcal{H}^{\phi}_{\xi_\epsilon}\mathcal{E}$ and $\mathcal{T}_\xi\mathcal{M}$. To see why this is desirable, recall that our problem statement assumes that the space $\mathcal{E}$ is a tubular neighborhood (75; 6; 128). Within this space, we can define a Euclidean closest point function $\mathrm{cp}_{\mathcal{M}} : \mathcal{E} \to \mathcal{M}$ given by $\mathrm{cp}_{\mathcal{M}}(\xi_\epsilon) = \arg\min_{\xi\in\mathcal{M}} \|\xi_\epsilon - \xi\|_2$ (128). Then, $\mathrm{cp}_{\mathcal{M}}(\xi_\epsilon)$ is our best guess for ξ, since, by our assumption that ϵ_ξ is confined to the normal space, $\mathrm{cp}_{\mathcal{M}}(\xi_\epsilon) = \xi$. However, the map $\mathrm{cp}_{\mathcal{M}}$ is unknown, as is $D\mathrm{cp}_{\mathcal{M}}$, which projects into the tangent space $\mathcal{T}_{\mathrm{cp}_{\mathcal{M}}(\xi_\epsilon)}\mathcal{M}$, and from which we could therefore recover $\mathcal{T}_{\mathrm{cp}_{\mathcal{M}}(\xi_\epsilon)}\mathcal{M}$ exactly.

We thus propose that this tangent space can be recovered from $D\phi$ instead, given that ϕ also satisfies one more key condition.

Assumption 1. $\phi(\xi_\epsilon) = \phi(cp_{\mathcal{M}}(\xi_\epsilon))$.

Note that, by our previous assumption that ϵ_ξ lies in the normal space $\mathcal{N}_\xi(\mathcal{M}, \mathcal{E})$, we have that $\mathrm{cp}_{\mathcal{M}}(\xi_\epsilon) = \xi$, so we can simplify the above as $\phi(\xi_\epsilon) = \phi(\xi)$. Before giving our main result, we require one further lemma. Although we use the notation f for an arbitrary function, we will apply this result to embedding coordinates ϕ^k rather than the response variable f.

Lemma 50. *Given a function $f : \mathcal{E} \to \mathbb{R}$ satisfying $f(\xi_\epsilon) = f|_{\mathcal{M}}(cp_{\mathcal{M}}(\xi_\epsilon))$, and points $x \in \mathbb{R}, \xi \in \mathcal{M}, \xi_\epsilon \in \mathcal{E}$ s.t. $f|_{\mathcal{M}}(\xi) = f(\xi_\epsilon) = x$, the level sets of f at levels x satisfy*

$\mathcal{T}_{\xi_\epsilon} f^{-1}(x) \supseteq \mathcal{N}_\xi(\mathcal{M}, \mathcal{E})$ for all ξ_ϵ s.t. $cp_{\mathcal{M}}(\xi_\epsilon) = \xi$.

Proof. This proposition is adapted from (128) (see Definition 3.1 and Theorem 3.7). Given a value $x \in \mathbb{R}$, we define the level set of f w.r.t. x to be $f^{-1}(x) := \{\xi_\epsilon \in \mathcal{E} : f(\xi_\epsilon) = x\}$, and similarly for a map with manifold codomain. Note that, for a smooth map, this preimage locally has manifold structure, and so we can consider its tangent spaces. Theorem 3.7 of (128) gives that the level sets of $\mathrm{cp}_{\mathcal{M}}$ intersect $\mathcal{M}$ orthogonally, i.e.

$$T_\xi \mathrm{cp}_{\mathcal{M}}^{-1}(\xi) = \mathcal{N}_\xi(\mathcal{M}, \mathcal{E})$$

within a tubular neighborhood $\mathcal{E}$ of $\mathcal{M}$.

We make two adjustments to this proposition. First,

$$\mathcal{T}_{\xi_\epsilon} \mathrm{cp}_{\mathcal{M}}^{-1}(\xi) = \mathcal{N}_\xi(\mathcal{M}, \mathcal{E})$$

for all ξ_ϵ s.t. $\mathrm{cp}_{\mathcal{M}}(\xi_\epsilon) = \xi$. This is true in particular for the Euclidean closest point function. The shortest Euclidean path between two points is a straight line, and so $\mathrm{cp}_{\mathcal{M}}^{-1}(\xi)$ is a linear space, and thus, $\mathcal{T}_{\xi_\epsilon} \mathrm{cp}_{\mathcal{M}}^{-1}(\xi) = T_{\xi_\epsilon'} \mathrm{cp}_{\mathcal{M}}^{-1}(\xi)$ for all $\xi_\epsilon, \xi_\epsilon' \in \mathrm{cp}_{\mathcal{M}}^{-1}(\xi)$. We can then choose ξ for ξ_ϵ'. Second, in contrast to $\mathrm{cp}_{\mathcal{M}}$, which is a submersion onto a d_M dimensional manifold, and thus rank d_M, we make no restriction on the rank of f. For this reason,

$$T_{\xi_\epsilon} f^{-1}(x) \supseteq N_\xi(M, E)$$

for all ξ_ϵ s.t. $f(\xi_\epsilon) = x$ and ξ s.t. $\mathrm{cp}_{\mathcal{M}}(\xi_\epsilon) = \xi$. In particular, $f^{-1}(x) \supseteq \mathrm{cp}_{\mathcal{M}}^{-1}(\xi)$ where $f|_{\mathcal{M}}(\xi) = x$. Therefore, $T_{\xi_\epsilon} f^{-1}(x) \supseteq T_{\xi_\epsilon} \mathrm{cp}_{\mathcal{M}}^{-1}(\xi)$ where $f(\xi_\epsilon) = x$, $f|_{\mathcal{M}}(\xi) = x$. □

We have shown that, given the closest point condition on f, the level sets of f span the normal space of $\mathcal{M}$ in $\mathcal{E}$. Our main result applies this to coordinate functions ϕ^k of an embedding map ϕ and their gradients.

Proposition 51. *Suppose we are given data ξ_ϵ describe as above, and ϕ s.t. $\phi(\xi_\epsilon) = \phi(cp_{\mathcal{M}}(\xi_\epsilon))$ and $\phi|_{\mathcal{M}}$ is a diffeomorphism from $\mathcal{M}$ to $\mathcal{B}$, then $\mathcal{H}_{\xi_\epsilon}^{\phi} \mathcal{E} = \mathcal{T}_\xi \mathcal{M}$.*

Proof. By the definitions of horizontal and vertical space, $\mathcal{V}^{\phi}_{\xi_\epsilon} = \ker(D\phi)$ and $\mathcal{H}^{\phi}_{\xi_\epsilon} = \mathcal{T}_{\xi_\epsilon}\mathcal{E}/\mathcal{V}^{\phi}_{\xi_\epsilon}$, we have that $\mathcal{H}^{\phi}_{\xi_\epsilon} = \text{rowspan}([\text{grad}_{\mathcal{E}}\, \phi^k(\xi_\epsilon) : k \in 1 \dots m])$. We therefore want to show that $\text{rowspan}([\text{grad}_{\mathcal{E}}\, \phi^k(\xi_\epsilon) : k \in 1 \dots m]) = \mathcal{T}_\xi \mathcal{M}$.

First, since gradients are perpendicular to level sets, $\text{grad}_{\mathcal{E}}\, \phi^k(\xi_\epsilon) \in \mathcal{T}_{\xi_\epsilon}\mathcal{E}/\mathcal{T}_{\xi_\epsilon}\phi^{k^{-1}}(x)$ for x and ξ_ϵ s.t. $\phi^k(\xi_\epsilon) = x$. Thus, Lemma 50 implies that $\text{grad}_{\mathcal{E}}\, \phi^k(\xi_\epsilon) \in \mathcal{T}_{\xi_\epsilon}\mathcal{E}/\mathcal{N}_\xi(\mathcal{M}, \mathcal{E})$, where $\text{cp}_{\mathcal{M}}(\xi_\epsilon) = \xi$. Since ξ_ϵ is sampled from a ball that is perpendicular to $\mathcal{M}$ within a tubular neighborhood, $\text{cp}_{\mathcal{M}}(\xi_\epsilon) = \xi$, and so $\mathcal{T}_{\xi_\epsilon}\mathcal{E} = \mathcal{T}_\xi\mathcal{M} \times \mathbb{R}^{d_E - d_M} = \mathcal{T}_\xi \mathcal{E}$. Since the normal space $\mathcal{N}_\xi(\mathcal{M}, \mathcal{E})$ is defined as $\mathcal{N}_\xi(\mathcal{M}, \mathcal{E}) = \mathcal{T}_\xi\mathcal{E}/\mathcal{T}_\xi\mathcal{M}$, the above implies that $\text{rowspan}([\text{grad}_{\mathcal{E}}\, \phi^k(\xi_\epsilon) : k \in 1 \dots m]) \subseteq \mathcal{T}_\xi\mathcal{M}$.

On the other hand, since ϕ is a surjective submersion onto $\mathcal{B} = \phi(\mathcal{M})$, it is a rank d_M map, and so $[\text{grad}_{\mathcal{E}}\, \phi^k(\xi) : k \in 1 \dots m]$ is rank d_M. Thus, $\text{rowspan}([\text{grad}_{\mathcal{E}}\, \phi^k(\xi) : k \in 1 \dots m]) = \mathcal{T}_\xi\mathcal{M}$. □

Proposition 51 gives an intuitive geometric condition for a learned embedding ϕ to be usable in EmbedTS. In the next section, we will apply this method to estimation of $\text{grad}_{\mathcal{M}} f$ of an observed covariate f. This will result in a two-stage estimator.

6.5 *The* EmbedGrad *algorithm*

With $\mathcal{T}_{\xi_i}\mathcal{M}$ estimated, we now turn our focus to estimation of $\text{grad}_{\mathcal{M}} f$ from noisy observed data ξ_ϵ and f_ϵ. A mathematical schematic of our full problem setting is given in Figure 6.2. $\mathcal{M}$ and $\mathcal{E}$ are assumed to meet the conditions given in Section 6.2, and $\phi(\mathcal{M})$, our learned embedding, satisfies Assumption 1.

Given a data point $\xi_{\epsilon,i}$, the EmbedGrad algorithm takes as input local data matrices $\delta_i^{\xi_\epsilon}$ and δ_i^f, local embedding matrices δ_i^{ϕ}, intrinsic dimensions d_M and d_E, and smoothing bandwidth ϵ. The local position matrices are computed using neighborhoods $\mathcal{N}_i$ identified as in Section 2.3 using a neighborhood radius parameter such as $r = 3\epsilon$. We first call EmbedTS in Step 1 to estimate $\mathcal{T}_{\xi_i}\mathcal{M}$. Subsequently, Step 2 projects the local data $\delta_i^{\xi_\epsilon}$ into this tangent space prior to solving a local linear regression to estimate $\text{grad}_M f(\xi)$. Note that, assuming exact estimation of $\mathcal{T}_{\xi_i}\mathcal{M}$, the obtained gradient in this step is automatically in $\mathcal{T}_{\xi_i}\mathcal{M}$ because the regressors are already in this space. Therefore, we also optionally return $\widehat{\nabla}_{\mathcal{M}} f$ - the gradient in the original coordinates.

$$\begin{tikzcd}
\mathbb{R}^D & \mathbb{R}^m \\
\cup & \cup \\
\mathcal{E} \arrow[r, "\phi"] & \mathcal{B} \\
\mathcal{M} \arrow[u, "+\epsilon_\xi"] \arrow[ur, "\phi_{\mathcal{M}}"] \arrow[r, "f"] & f(\mathcal{M}) \arrow[d, "+\epsilon"] \\
 & f_\epsilon(\mathcal{M})
\end{tikzcd}$$

Figure 6.2: The functional setting of error-in-variables gradient estimation. Inclusions are shown in green, submersions in purple, and diffeomorphisms in yellow. Blue refers to our observables. Red refers to the output of a manifold-learning algorithm. $\mathbb{R}^D$ and $\mathbb{R}^m$ are the feature and embedding spaces, respectively. Our goal is the estimation of $\text{grad}_M f$ at various data points.

Algorithm 11 EmbedGrad(Local data matrices $\delta_i^{\xi_\epsilon}$ and δ_i^{f}, local embedding matrices δ_i^{ϕ}, intrinsic dimensions d_M and d_E, kernel bandwidth ϵ)

```
Compute $\widehat{T}_i^{\mathcal{M}} \leftarrow$ EmbedTS($\delta_i^{\xi_\epsilon}, \delta_i^{\phi}, d_M, d_E, \epsilon$)
Compute $\widehat{\text{grad}_{\mathcal{M}} f}(\xi_i) \leftarrow$ OLS($\delta_i^{\xi_\epsilon}\widehat{T_i^{\mathcal{M}}}, \delta_i^{f_\epsilon}$)$\widehat{T_i^{\mathcal{M}}}^T \in \mathbb{R}^D$ for $i = 1 : n$
Output $\widehat{\text{grad}_{\mathcal{M}} f}(\xi_i)$ (optional $\widehat{\nabla_{\mathcal{M}} f(\xi_i)} = \widehat{T_i^{\mathcal{M}}}\widehat{\text{grad}_{\mathcal{M}} f}(\xi_i)$ )
```

Hyperparameters We fix neighborhoods and weights over all our subalgorithms to illustrate the differentiating features of our approach. Fixing a radius for both gradient estimation and local PCA was used in (17). In general, choosing the neighborhood scale for manifold (16; 186; 117; 51) and tangent space (45; 6; 155) estimation in noise is an incompletely solved problem.

Computation The complexity of the SVD step is $O(|\mathcal{N}_i|)D \vee D^3$. Linear regression is $O(qd_E^2 + Dd_E^3)$ where $q = \sum_{i \in I} |\mathcal{N}_i|$. The neighborhood computation - also the expensive part of many embedding steps - is $D \log n$.

6.5.1 Variants and extensions

Gradient estimation The local linear regression estimator that we use exemplifies the role of the tangent space in gradient estimation, since comparison of Proposition 2.13 with respect to our gradient estimation problem shows that a non-zero value of $\mathrm{Cov}(\epsilon, \epsilon_\xi)$ will cause the estimator to be biased. However, many classes of both simpler (142) and higher-order estimators exist (125). We do not exhaustively examine these here, but note that projection onto $\mathcal{T}_\xi \mathcal{M}$ is necessary for computing $\mathrm{grad}_{\mathcal{M}} f$ regardless of estimator. Furthermore, estimators such as those introduced in (17) explicitly make use of the tangent space estimate as a regularizer, and claim this can have a substantial benefit on convergence, regardless of $\mathrm{Cov}(\epsilon, \epsilon_\xi)$. We therefore also examine the effect of estimation of $\mathcal{T}_\xi \mathcal{M}$ in the case of uncorrelated noise in Section 6.6.

Tangent space estimation Our EmbedTS estimator of $D\phi_D^m$ is related to performing a localized slice inverse regression using the Laplacian eigenfunctions to provide an effective dimension reduction space (193). However, other methods for establishing linear relationships could be used (30). In particular variety of adaptations of generalized eigenproblem-based decompositions including the first stage in two-stage least squares, in which $\delta_i^{\xi_\epsilon}$ is predicted using δ_i^{ϕ}, could also be applied to our overall mathematical set-up. Total least squares in an interesting limiting case of these methods, as it corresponds to local principal components with the inclusion of the embedding in the feature set (94).

Dimension Estimation Our approach connects to estimation of d_M, the intrinsic dimension of M. Two archetypal classes of intrinsic dimension estimators are those which rely on the spectra of local principal components, and those which are based on geometric quantities like the rate of increase of samples in balls of radius r (121; 63). We draw a connection to the former (45). The theoretical rank of $D\phi_D^m$ is d_M, so we can estimate d_M from the spectrum of $D\phi_D^m$. This can be understood in the context of our least squares estimator. Since $\mathcal{B}$ is d_M dimensional, the local difference matrix $\tilde{\delta}_i^{\phi}$ has rank d_M within a sufficiently small ball. Therefore, $\widehat{D\phi_D^m}$ estimated using the usual OLS regression formula necessarily has rank $\leq d_M$ as well, and spectral decomposition of our estimator gives an estimate of the tangent space that inherits dimensionality from the embedding. Thus, the spectrum of $\widehat{D\phi_D^m}$ can be used to estimate the dimensionality of $\mathcal{M}$. Note that it is the intrinsic dimension of $\mathcal{B}$, not the number of embedding coordinates m, that is important. However, if increasing m causes $\mathcal{B}$ to be diffeomorphic to a larger space than $\mathcal{M}$ (for example, $\mathcal{E}$ in the limit), then this estimator is no longer applicable. This property is examined experimentally in Section 6.6.

6.6 Experiments

We demonstrate our method on several real and simulated examples. The numerical study of estimation accuracy is challenging, since an estimator that is biased for estimation may still provide good predictions. Therefore, we begin by evaluating our approach in simulation.

Our main experimental emphases are the preferable performance of EmbedTS to WLPCA , and EmbedGrad to OLS with WLPCA-based estimation of $T_i^{\mathcal{M}}$ or $T_i^{\mathcal{E}}$. Recall that, at a minimum, $T_i^{\mathcal{E}}$ must be estimated prior to solving OLS$(\delta_i^{\xi_\epsilon}\widehat{T_i^{\mathcal{E}}}, \delta_i^f)$ in order for the solution to be well-defined. To control for variability between methods, we fix a radius, neighborhood graph, and bandwidth for use in both WLPCA and EmbedTS. Hyperparameters and other information are given in Table 6.1.

Information We list the parameters and dataset information for our two main experiments. Note that for the SDSS data, the data are mapped from the original 3750 dimensional space to a 50 dimensional PCA space prior to running EMBED and EMBEDGRAD. Empirically, for the Sloan Digital Sky Survery data discussed in Section 6.6.2, the multiscale data distribution necessitates use of k-nearest-neighborhoods rather than radius-based neighborhoods for generation of a smooth multidimensional embedding.

Experiment	n	p	d_E	d_M	γ	r	Num. Nebs.	m
Swerved cylinder	10000	3	2	1	.25	.25		6
SDSS	74628	50	20	2	35000		60	3

Table 6.1: Parameters and data information. WLPCA, EMBEDTS and EMBEDGRAD are performed with the same bandwidth γ, neighborhood radius r, or num. neighbors.

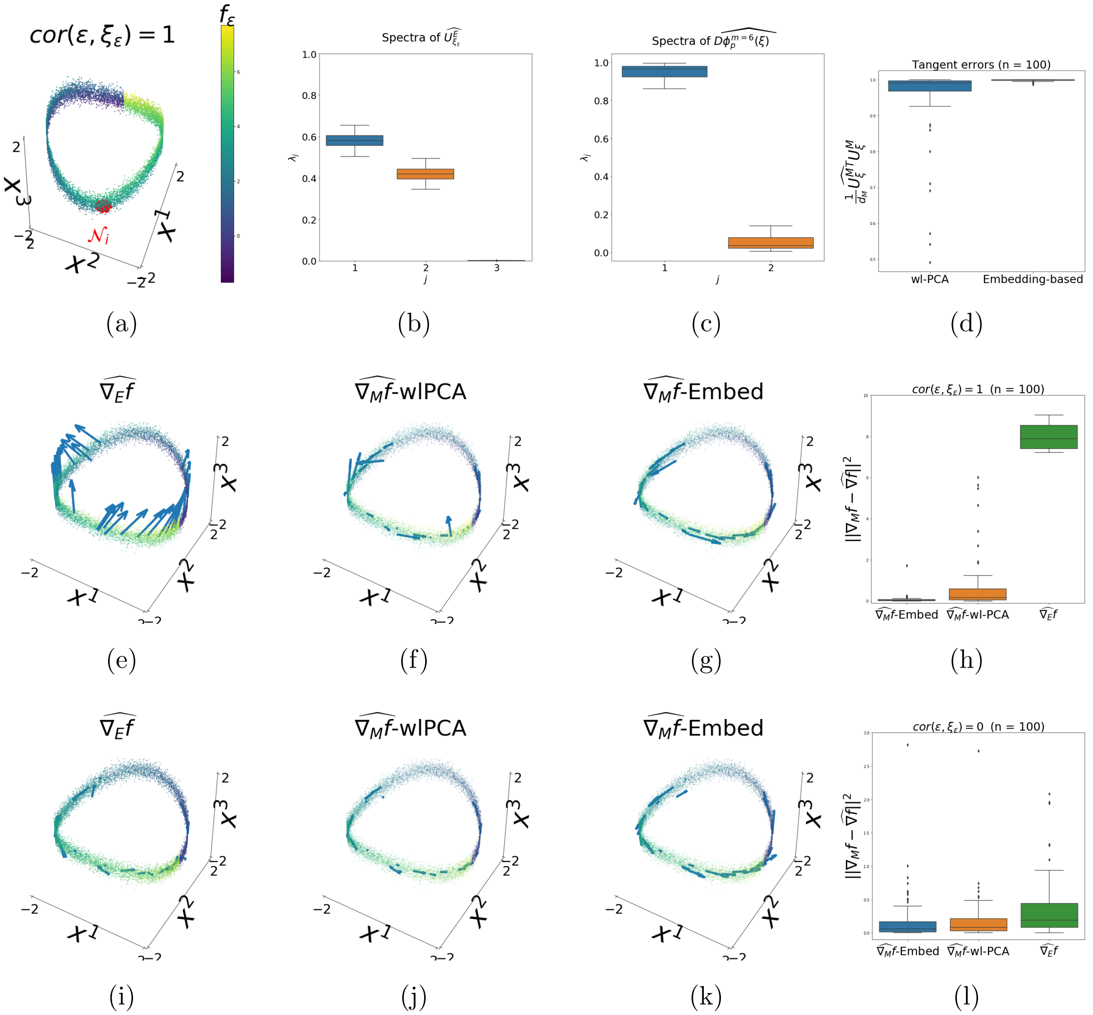
6.6.1 Toy data

We create two simulations of error-in-variable gradient estimation. For both, our manifold is parameterized in 3 dimensions as $[2\cos(\theta), 2\sin(\theta), \sin(\theta)^2]$ for $\theta \in [0, 2\pi)$. This is a challenging manifold to perform classical dimension reduction on due to its nonlinear nature. We add Gaussian noise $\xi_\epsilon \sim N(0, .2)$ in the "vertical" normal direction, i.e. the quotient of the normal space by the radial direction. This gives a manifold with $d_E = 2$ and $d_M = 1$. Our response function is $f_\epsilon = \theta + \epsilon$, where $\epsilon \sim N(0, .6)$. In our first simulation, $\text{cor}(\epsilon, \xi_\epsilon) = 1$. Figure 6.3a shows the manifold colored by the f_ϵ with such correlated noise. In the second, the two error terms are independent. However, there is still reason to believe that a projection onto $T_i^{\mathcal{M}}$ rather than $T_i^{\mathcal{E}}$ may be beneficial (17; 45). We evaluate our tangent space estimator and resultant gradient estimates with respect to 1 and 2 dimensional tangent space estimates made using WLPCA across 100 data points. Embedding coordinates are shown in Figure 6.4.

Our results show that, at these hyperparameters, EMBEDTS estimates the true $T_i^{\mathcal{M}}$ with lower error than WLPCA. We first illustrate that spectral gaps of both the differential $\widehat{D_D^m}$ used in EMBEDTS and the local weighted covariance from the penultimate step of WLPCA given in

Section 2.3.6. Comparing Figures 6.3b and 6.3c shows that EmbedGrad generates tangent spaces with larger spectral gap. This reflects the clean low-dimensional embedding in Figure 6.4, and contrasts with the noise in Figure 6.3a. Figure 6.3d then shows that EmbedTS performs preferably to WLPCA in maximizing the normalized similarity $\frac{1}{d_M}\|\widehat{T_i^{\mathcal{M}}}^T T_i^{\mathcal{M}}\|_2$ between the estimated and true tangent spaces.

These tangent spaces also yield lower error in subsequent gradient estimation. Figure 6.3e shows gradient estimates made with respect to the $d_E = 2$ dimensional tangent space, while Figures 6.3f and 6.3g show gradients estimated in $d_M = 1$ dimensional tangent spaces estimated by WLPCA and our approach, respectively. Figure 6.3h confirms what is visually evident that EmbedGrad method performs preferably w.r.t. ℓ_2 loss. Figures 6.3i- 6.3k show gradients estimated in uncorrelated noise, and 6.3l shows the preferable performance of EmbedGrad.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 6.3: Results for wavy cylinder simulation evaluated at 100 data points. 6.3a Data colored by f_ϵ with correlated errors shown on $\mathcal{E}$. A neighborhood is shown in red. 6.3b The spectrum of the local covariance used by WLPCA. 6.3c The spectrum of the $\widehat{D\phi^m_D}$. 6.3d Tangent space recovery error. 6.3e - 6.3h Results for correlated errors. 6.3e Estimates of $\nabla_{\mathcal{M}} f$ obtained w.r.t. two-dimensional $\mathcal{T}_{\xi_\epsilon}\mathcal{E}$ at 20 data points. 6.3f Estimates of $\nabla_{\mathcal{M}} f$ obtained w.r.t. the first component (i.e. estimate of $\mathcal{T}_\xi\mathcal{M}$) of WLPCA. 6.3g Estimates of $\nabla_{\mathcal{M}} f$ obtained using EMBEDGRAD. 6.3h Error of gradient estimates. 6.3i - 6.3l Results for uncorrelated errors. 6.3i Estimates of $\nabla_M f$ obtained w.r.t. a two-dimensional $\mathcal{T}_{\xi_\epsilon}\mathcal{E}$ at 20 data points. 6.3j Estimates of $\nabla_M f$ obtained w.r.t. the first component (i.e. estimate of $\mathcal{T}_\xi\mathcal{M}$) of WLPCA. 6.3k Estimates of $\nabla_M f$ obtained using EMBEDGRAD. 6.3l Error of gradient estimates.

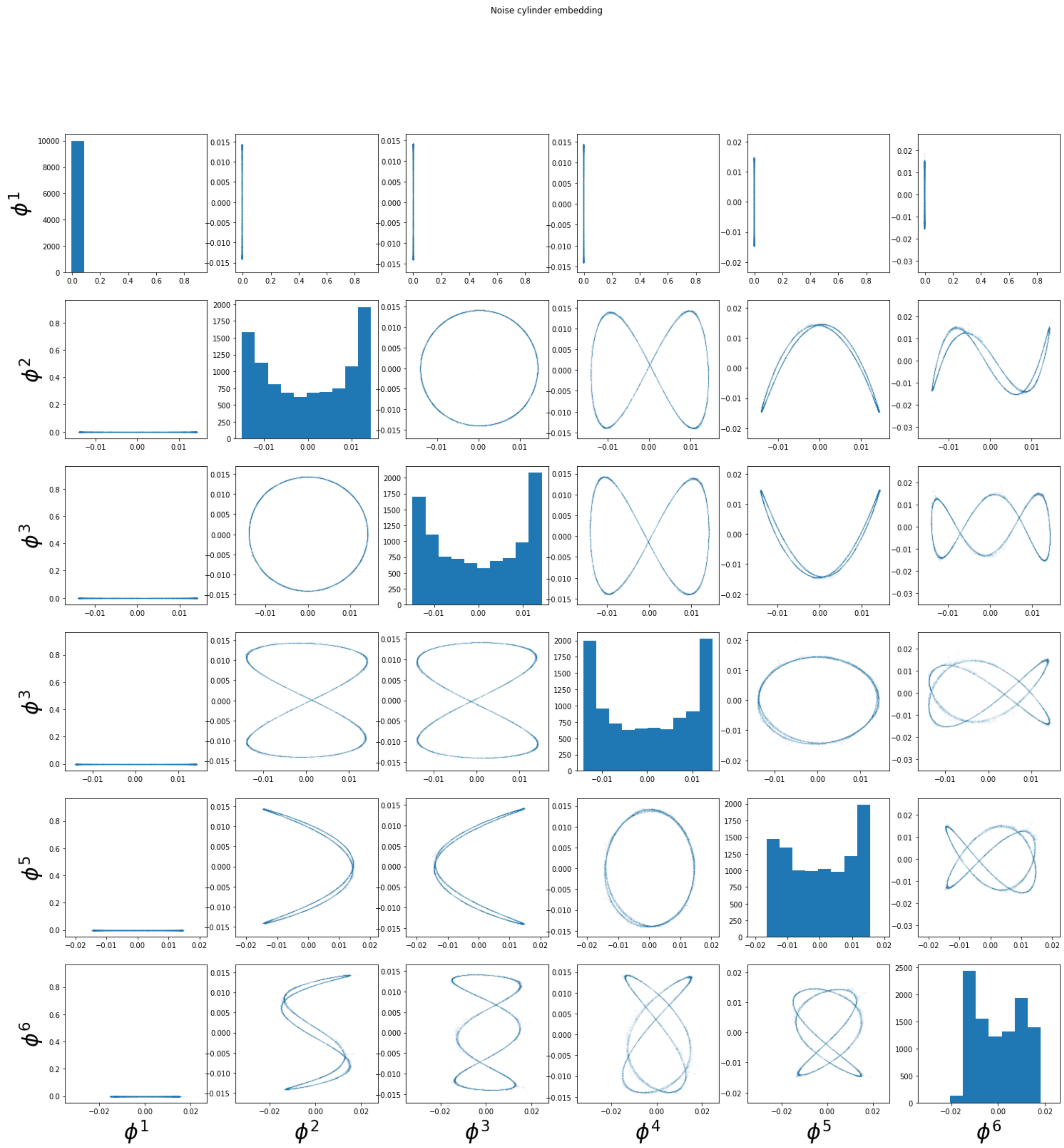


Figure 6.4: The 6 coordinates of the curved cylinder embedding used in EmbedGrad. These plots suggest that the embedding includes eigenfunctions that are harmonics of each other, but nevertheless that the learned manifold is a circle diffeomorphic to the base space.

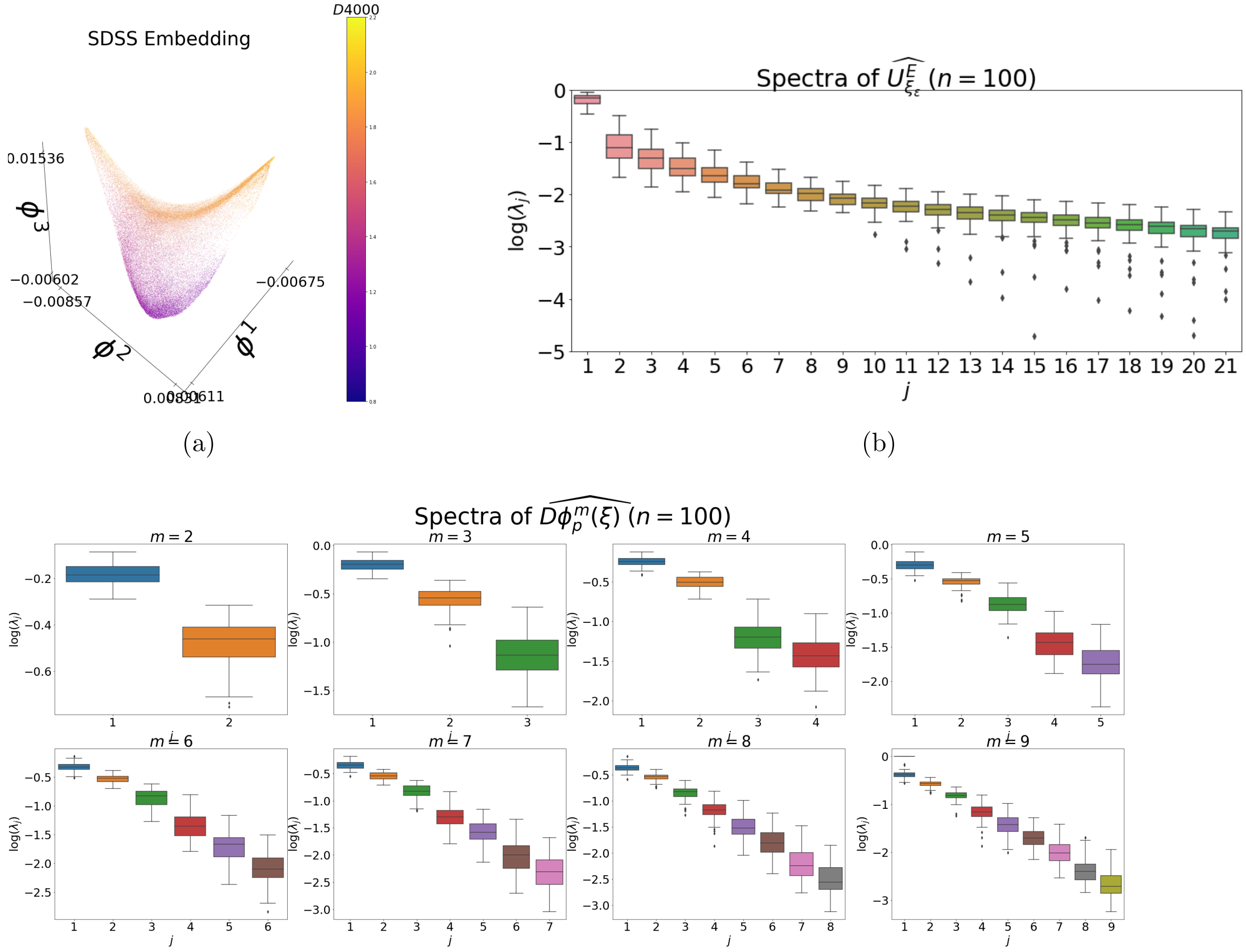
6.6.2 Gradient estimation in astronomy

Our motivating application comes from astronomy. The Sloan Digital Sky Survey dataset contains several modalities of data on galaxies (26). One of these describes the intensity of the observed signal corresponding to each galaxy at 3750 different wavelengths. This dataset is a classic use-case for manifold learning, since the observed data are concentrated around some low-dimensional manifold (43). Another data modality consists of other quantitative descriptors of the galaxies such as mass. Some of these descriptors such as band head index D4000 are calculated from the spectrographic measurements themselves, so we might expect them to be correlated with the observed intensities. Despite this, we do not have access to the analytic gradient, and so we therefore use $D4000$ as an exemplary black-box function whose gradient we wish to compute. Since we do not have access to the ground truth with which to quantitatively evaluate our estimates, we instead call attention to several key features of our results.

The D4000 covariate function is plotted in the top three embedding coordinates in Figure 6.5a. Our task is to estimate the gradient of this function with respect to the data manifold in the original, high-dimensional feature space. To slightly simplify the problem, and for the purposes of plotting, we first project the data from 3750 dimension down onto the top 50 principal components. The projections of each galaxy into this space are the $\xi_{\epsilon,i}$ on which we apply EmbedGrad.

Pairplots of these ξ_ϵ given in Figures 6.6, 6.8, and 6.7 show that that the observed data is seemingly much higher-dimensional than the learned two-dimensional embedding. In contrast, the spectra obtained using WLPCA at 100 data points shown in Figure 6.5b implies that the data is locally one-dimensional, with a steadily decreasing rate of eigenvalue decay up to $d_E = 20$. This is the typical of the difficult-to-interpret spectra generated by this algorithm on noisy data. It also contrasts with the clear spectral gap in $\widehat{D\phi_D^m}$ between two and three dimensions in Figure 6.5c at $m = 3$ and, particularly, $m = 4$. In order to examine the effect of embedding dimension on the spectra of the obtained tangent spaces, we include results up to $m = 9$. While these do not show a clear spectral gap, the overall rate of eigenvalue decay is more rapid than for WLPCA, indicating that the embedding coordinates are sharply identifying a lower-dimensional structure.

We plot results for $D4000$ gradients at 10 data points estimated using WLPCA tangent spaces of $d_M = 2$ and $d_E = 20$ dimensions, as well as EmbedGrad results, in the top 6 coordinates of the PCA feature space in Figures 6.6, 6.8, and 6.7. Figure 6.8 shows the gradients computed w.r.t. M by EmbedGrad. Despite being constrained to lie in only a two-dimensional space and estimated using only 60 neighbors, these gradients provide a visually reasonable parameterization of the visualized high-dimensional data.



(a)

(b)

(c)

Figure 6.5: Application to the SDSS dataset. 6.5a A learned manifold in the top three embedding coordinates colored by the $D4000$ covariate function. 6.5b The spectrum of the estimated tangent spaces using WLPCA. 6.5c The spectrum of the estimated tangent spaces obtained using EmbedTS at various embedding dimensions m.

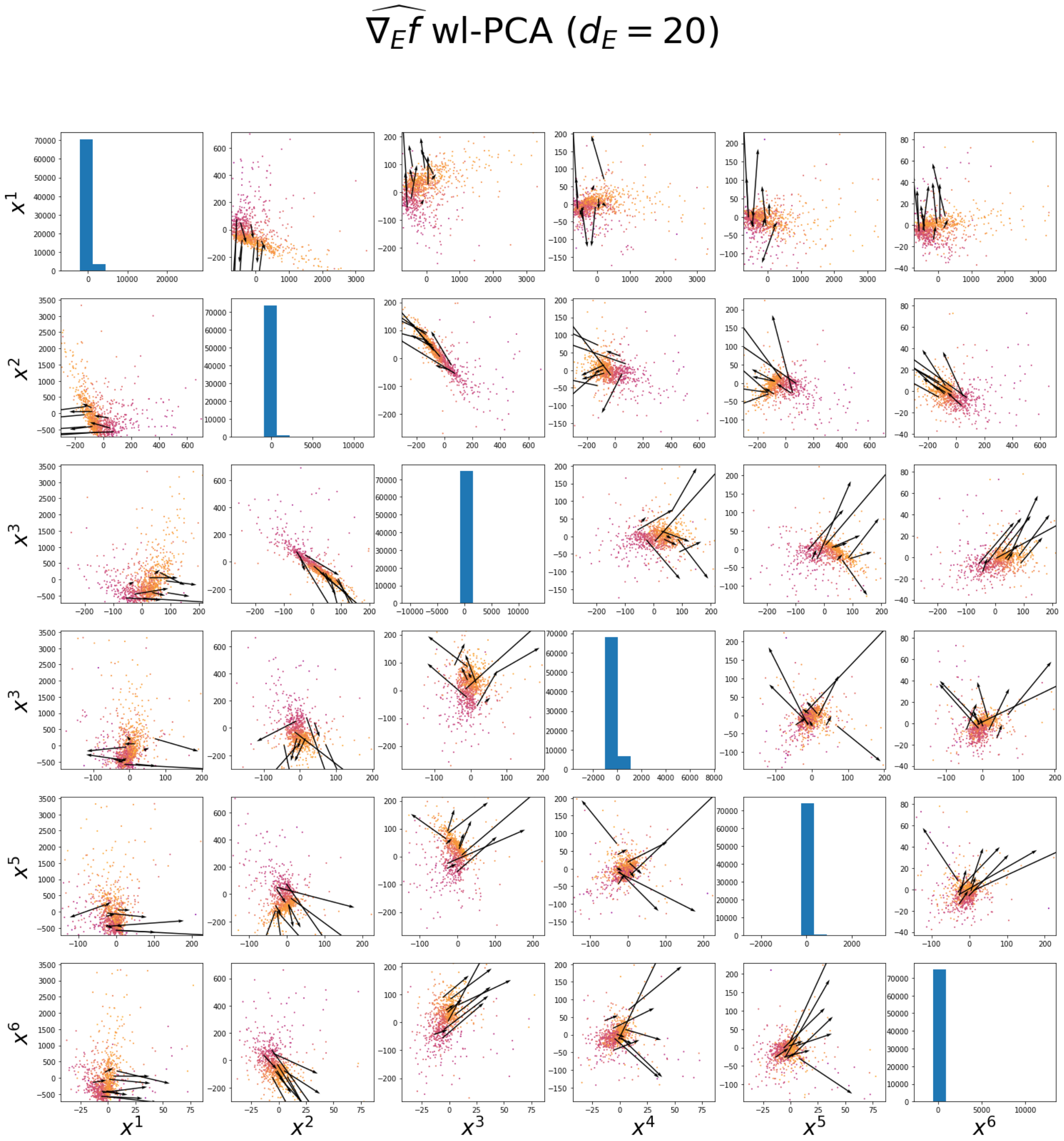


Figure 6.6: Gradients ∇_E of $D4000$ estimated using a 20 dimensional WLPCA tangent space. Data is plotted in the top 6 PCA coordinates, and colored by $D4000$.

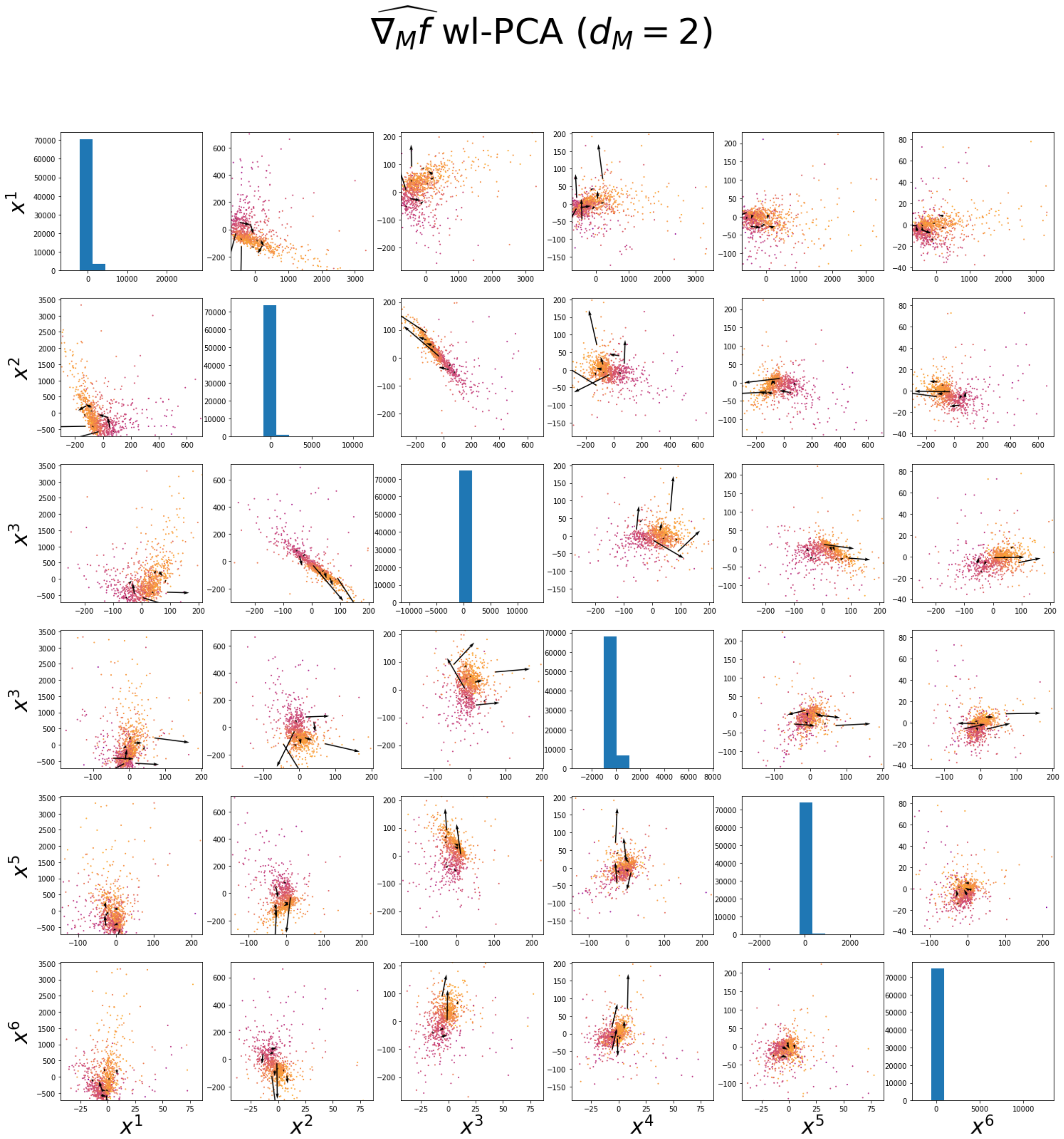


Figure 6.7: Gradients ∇_M of $D4000$ estimated using a 2 dimensional WLPCA tangent space. Data is plotted in the top 6 PCA coordinates, and colored by $D4000$.

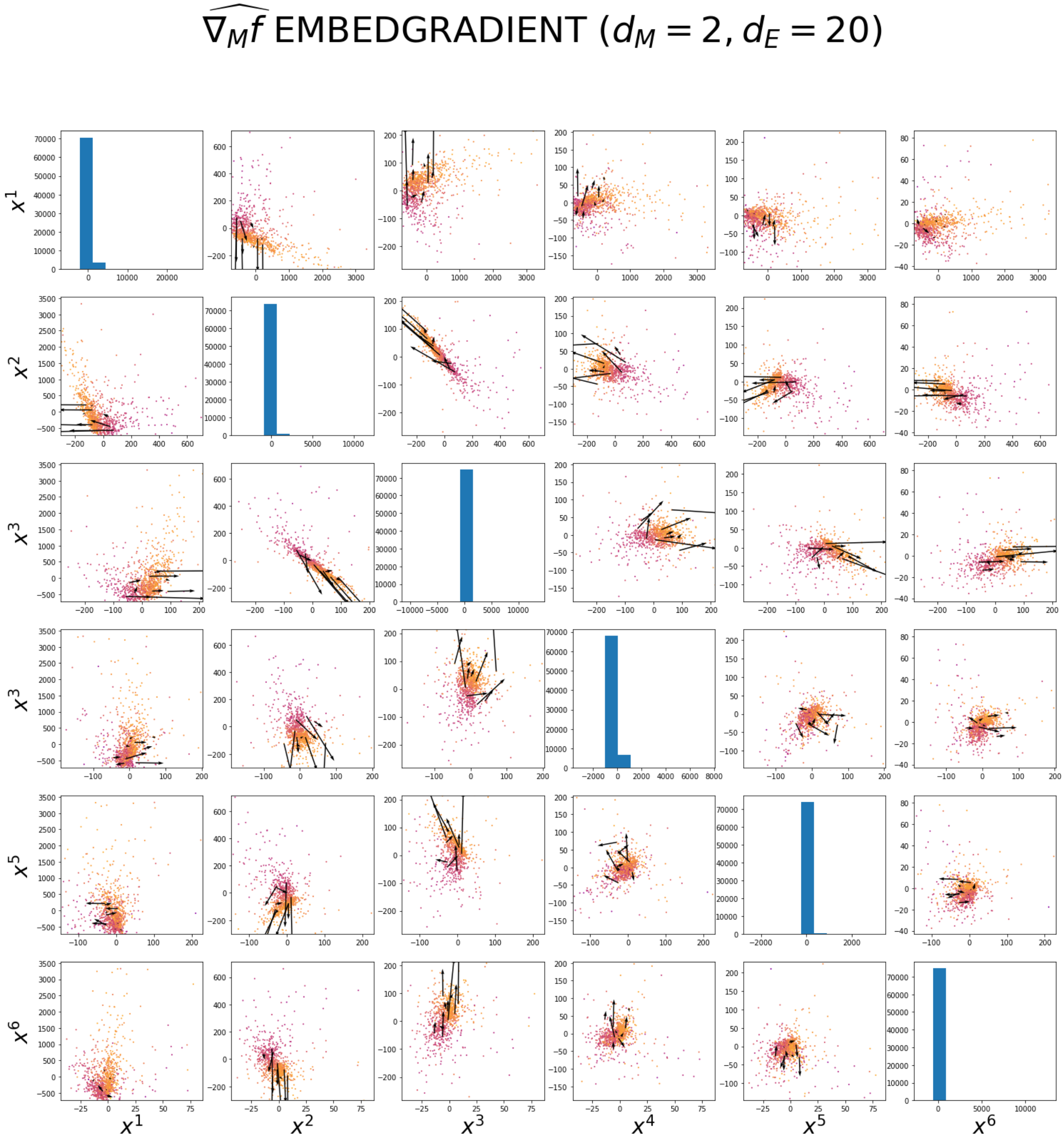


Figure 6.8: Gradients ∇_M of $D4000$ estimated EmbedGrad with $d_E = 20$ and $d_M = 2$. Data is plotted in the top 6 PCA coordinates, and colored by $D4000$.

6.7 Discussion

Our EmbedGrad algorithm is a two-stage algorithm that uses learned representations to estimate the data manifold tangent space before performing a local linear regression in the tangent space. By using a globally-informed embedding to estimate local structure, we see better performance than using a purely local estimator. This idea suggests several related lines of research.

From a theoretical perspective, showing that an embedding satisfies Assumption 1 asymptotically would be very useful. Consistent estimation of an embedding satisfying Assumption 1 seems to be a useful criteria for judging manifold learning algorithms. Both Principal Components Analysis and Principal Curves (79; 100) satisfy our main assumption in the infinite data limit. However, the former is only consistent for linear manifolds, and the later finds the curves in the original high-dimensional space, and so does not leverage the flexibility of Proposition 51. Other embedding algorithms that favor smoothness, such as Maximum Variance Unfolding (190), could also potentially be used. In our setting, given the similarity between the spectral decomposition in wlpca and Diffusion Maps, our choice of embed, it seems plausible that one could prove that the top eigenvectors of the sample Laplacian are more robust to noise than the top eigenvectors of wlpca. However, even the convergence of Laplacian Eigenmaps also requires asymptotically decreasing noise, although, promisingly, the rate of decrease is slower than for local PCA (171; 6) (see also Section 2.4). Compared with the recent method of (155), we could see our approach having an advantage in, for example, datasets consisting of disjoint manifolds. More generally, convergence results for multistage least squares algorithms would seem to have relevance for deep networks.

Given an embedding satisfying Assumption 1, there are several alternatives to our slice inverse-type regression method for estimating $\mathcal{T}_\xi \mathcal{M}$ given the fiber assumption with two-stage and partial least squares-type approaches (30). This includes the slice inverse regression-type method used here (193; 192), but also partial least squares and the first stage of two-stage-least-squares. These are all variants of the generalized eigenproblem (106; 14; 73), and it seems plausible that techniques for analyses of M-estimator could be used to assess which approach is asymptotically optimum (184). Note that the first stage of our estimator would not be necessary if obtained via differentiable programming rather than estimation.

The concept of a multidimensional or rank-deficient map is well-studied in the differential and algebraic geometry literature (18), but despite its relevance to practical datasets (79; 146), statistical accounting for multiscale and mixed-rank structure is limited (22; 168; 202). The effect of rank-deficiency of the embedding is is evident through comparison with methods for associating gradients in embeddings and feature spaces (127; 131). These approaches transform vectors between spaces of equal dimensionality, while we identify the counterpart of a lower-dimensional space within a higher-dimensional space. Relatedly, we note that there is no requirement that the embedding ϕ be an isometry w.r.t. the induced metrics from $\mathbb{R}^D, \mathbb{R}^m$, since the amount of local distortion is captured by Σ_i in EmbedTS. We note that estimation of gradients of embedding functions via local linear regression gives an alternative estimator of the pushforward Riemannian metric that does not rely on the asymptotic linearity of the estimator of the Laplacian.

This method and its presentation in this chapter have several limitations. The hyperparameters in embed, wlpca, and EmbedGrad have an effect on the results. However, wlpca hyperparameters that perform comparably to our method require large neighborhood sizes different from those suggested in theoretical analyses of wlpca, and do not generate as clear of a spectral gap as our approach. We also have not proved that an embedding algorithm satisfies our main assumptions, and did not examine estimability of $\nabla_E \phi^k$. Our use of consistent neighborhood sizes and smooth parameters between algorithms is critical fair comparison of tangent spaces learned using wlpca and our method, since we would like for the localized parts of each algorithm to 'see' the same data, so that the only difference is the computation of local covariance (in wlpca) and gradients of embedding coordinates (in EmbedTS). If the radius is too small, then both wlpca and EmbedTS perform poorly for estimating $\mathcal{T}_\xi \mathcal{M}$, but the latter fails because the embedding may become diffeomorphic to $\mathcal{E}$ rather than $\mathcal{M}$, while the former fails due to inability to observe global structure. This obstacle posed by small neighborhoods for finding a 'good' embedding may be particular to our choice of EMBED, and it is possible that a different approach would have more success. At large radius values, we found that WLPCA can be as successful as EMBEDTS, albeit with a small spectral gap. However, this good performance is different from that guaranteed in the typically-studied asymptotic regime. We emphasize that our results hold at intermediate neighborhood sizes where a successful embedding (in terms of generating a diffeomorphic manifold to $\mathcal{M}$) is generated, but WLPCA fails. Furthermore, the matrices resulting from application of WLPCA prior to our embedding-based tangent estimation must necessarily be well-conditioned for OLS to be used, but the role of the dimensionality of the noise manifold $\mathcal{E}$, the extent to which estimation deteriorates in the absence of favorable conditioning, and the benefit of our replacement of the Moore-Penrose inverse with WLPCA are still unexplored.

Chapter 7

CONCLUSION AND FUTURE DIRECTIONS

This thesis has proposed several algorithms in the area of interpretable unsupervised learning. Chapters 3 and 4 introduced a sparse convex regression approach for identifying local diffeomorphisms from a dictionary of interpretable functions. In Chapter 3, this algorithm made use of an embedding learned by a manifold learning algorithm, while in Chapter 4, this algorithm was applied without the use of a precomputed embedding. Chapter 5 then introduced a simpler set of alternative algorithms that avoided issues stemming from sparse regression, characterized the tangent space version of this algorithm as identifying isometries when available, and gave a two-stage algorithm combining this approach with the computational advantages of the algorithms in Chapters 3 and 4. Chapter 6 then introduced an alternate tangent space estimator based on a learned embedding, and used this as an initial estimator to tackle the related gradient estimation problem. Together, these approaches provide a toolbox of methods for computing and associating gradient information to learn descriptive parameterizations of data manifolds.

Interpretability is important for human understanding and decision making. However, as we observed in (36), it is also important for improving sample-efficiency of learning. Further research on this topic motivated by both of these considerations is ongoing (53). The gradient group lasso approach could be used both in this area, and also potentially in optimization of deep networks. The selection of which functions to include in feature set also deserves special attention. We have shown results identifying torsions as parameterizing manifolds observed in a planar angle feature space. This selection is to an extent arbitrary, but nevertheless it is possible that notions of intrinsic curvature, functional independence, and probability could resolve the ambiguous question of what makes a parameterization "best". Similarly, topological data analysis for mixed rank or other topologically complex structures is required for a more precise accounting of data geometry. Basic homological invariants insufficiently describe non-manifold shapes like the pinched torus. More speculatively, since atoms of the same type are interchangeable in the Schrodinger equation, the quantum shape space is quotiented into a mixed rank orbifold-type object. Although it would also be reasonable to duplicate the dataset using all possible permutations of alike atoms, more elegant solutions to this interchangability may also exist (27). Finally, fibration-based characterization of machine learning algorithms has attracted increasing amounts of attention (34). It is an exciting question whether this mathematical formalism will result in improved learning performance.

BIBLIOGRAPHY

[1] WOMPtalk-Manifolds.pdf. .

[2] 1902.06502.pdf. .

[3] GUDHI: Subsampling/example_choose_n_farthest_points.cpp. `https://gudhi.inria.fr/doc/latest/_subsampling_2example_choose_n_farthest_points_8cpp-example.html`, . Accessed: 2021-11-6.

[4] BasisPursuit-SIGEST.pdf. .

[5] 2.2. manifold learning. `https://scikit-learn.org/stable/modules/manifold.html`, . Accessed: 2021-10-20.

[6] Eddie Aamari and Clément Levrard. Nonasymptotic rates for manifold, tangent space and curvature estimation. *Ann. Stat.*, 47(1):177–204, February 2019.

[7] Andrew B Abel.

[8] Matthew A. Addicoat and Michael A. Collins. Potential energy surfaces: the forces of chemistry. In Mark Brouard and Claire Vallance, editors, *Tutorials in Molecular Reaction Dynamics*, chapter 2, pages 28–49. Royal Society of Chemistry Publishing, London, 2010.

[9] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. October 2018.

[10] Alekh Agarwal, Sahand Negahban, and Martin J Wainwright. Fast global convergence of gradient methods for high-dimensional statistical recovery. *aos*, 40(5):2452–2482, October 2012.

[11] El-Ad David Amir, Kara L Davis, Michelle D Tadmor, Erin F Simonds, Jacob H Levine, Sean C Bendall, Daniel K Shenfeld, Smita Krishnaswamy, Garry P Nolan, and Dana Pe'er. viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nat. Biotechnol.*, 31(6):545–552, June 2013.

[12] El-Ad David Amir, Kara L Davis, Michelle D Tadmor, Erin F Simonds, Jacob H Levine, Sean C Bendall, Daniel K Shenfeld, Smita Krishnaswamy, Garry P Nolan, and Dana Pe'er. viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nat. Biotechnol.*, 31(6):545–552, June 2013.

[13] E Anderson, Z Bai, and J Dongarra. Generalized qr factorization and its applications. *Linear Algebra Appl.*, 162-164:243–271, February 1992.

[14] T W Anderson. Asymptotic theory for canonical correlation analysis. *Journal of Multivariate Analysis*, 2002.

[15] Joshua D Angrist and Alan B Krueger. Instrumental variables and the search for identification: From supply and demand to natural experiments. *J. Econ. Perspect.*, 15 (4):69–85, December 2001.

[16] Ery Arias-Castro, Gilad Lerman, and Teng Zhang. Spectral clustering based on local PCA. *J. Mach. Learn. Res.*, 18(9):1–57, 2017.

[17] Anil Aswani, Peter Bickel, and Claire Tomlin. Regression on manifolds: Estimation of the exterior derivative. March 2011.

[18] David Ayala, John Francis, and Hiro Lee Tanaka. Local structures on stratified spaces. September 2014.

[19] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[20] Mikhail Belkin and Partha Niyogi. Convergence of laplacian eigenmaps. In B Schölkopf, J C Platt, and T Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 129–136. MIT Press, 2007.

[21] Mikhail Belkin and Partha Niyogi. Convergence of laplacian eigenmaps. In B Schölkopf, J C Platt, and T Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 129–136. MIT Press, 2007.

[22] Mikhail Belkin, Qichao Que, Yusu Wang, and Xueyuan Zhou. Graph laplacians on singular manifolds: Toward understanding complex spaces: graph laplacians on manifolds with singularities and boundaries. November 2012.

[23] N Benjamin Erichson, Lionel Mathelin, Steven L Brunton, and J Nathan Kutz. Diffusion maps meet nyström. February 2018.

[24] Tyrus Berry and Timothy Sauer. Consistent manifold representation for topological data analysis. June 2016.

[25] Richard L Bishop. Riemannian geometry. March 2013.

[26] Michael R Blanton and Matthew A Bershady. Sloan digital sky survey IV: Mapping the milky way, nearby galaxies, and the distant universe. *Astron. J.*, 154:28, July 2017.

[27] Mark Blumstein and Henry Kvinge. Multi-Dimensional scaling on groups. December 2018.

[28] Luigi Bonati, Yue-Yu Zhang, and Michele Parrinello. Neural networks-based variationally enhanced sampling. *Proc. Natl. Acad. Sci. U. S. A.*, 116(36):17641–17647, September 2019.

[29] Fred L Bookstein. Size and shape spaces for landmark data in two dimensions. *SSO Schweiz. Monatsschr. Zahnheilkd.*, 1(2):181–222, May 1986.

[30] Magnus Borga, Tomas Landelius, and Hans Knutsson. A unified approach to pca, pls, mlr and cca, 1992.

[31] K. J. Bowers, D. E. Chow, H. Xu, R. O. Dror, M. P. Eastwood, B. A. Gregersen, J. L. Klepeis, I. Kolossvary, M. A. Moraes, F. D. Sacerdoti, J. K. Salmon, Y. Shan, and D. E. Shaw. Scalable algorithms for molecular dynamics simulations on commodity clusters. In *SC '06: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, pages 43–43, 2006. doi: 10.1109/SC.2006.54.

[32] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, USA, 2004. ISBN 0521833787.

[33] Patrick Breheny and Jian Huang. COORDINATE DESCENT ALGORITHMS FOR NONCONVEX PENALIZED REGRESSION, WITH APPLICATIONS TO BIOLOGICAL FEATURE SELECTION. *Ann. Appl. Stat.*, 5(1):232–253, January 2011.

[34] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. April 2021.

[35] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. ISSN 0027-8424. doi: 10.1073/pnas.1517384113. URL `http://www.pnas.org/content/113/15/3932`.

[36] James Buenfil, Samson J Koelle, and Marina Meila. Tangent space least adaptive clustering. June 2021.

[37] Emmanuel Candes and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. *aos*, 35(6):2313–2351, December 2007.

[38] Jack Carr. *Applications of Centre Manifold Theory.* Springer, New York, NY, 1982.

[39] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proc. Natl. Acad. Sci. U. S. A.*, 116(45):22445–22451, November 2019.

[40] Tara Chari, Joeyta Banerjee, and Lior Pachter. The specious art of Single-Cell genomics.

[41] Guangliang Chen, Anna V. Little, and Mauro Maggioni. *Multi-Resolution Geometric Analysis for Data in High Dimensions*, pages 259–285. Birkhäuser Boston, Boston, 2013. ISBN 978-0-8176-8376-4. doi: 10.1007/978-0-8176-8376-4-13. URL https://doi.org/10.1007/978-0-8176-8376-4-13.

[42] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. June 2016.

[43] Yu-Chia Chen and Marina Meilă. Selecting the independent coordinates of manifolds with large aspect ratios. July 2019.

[44] Yu-Chia Chen, James McQueen, Samson J. Koelle, Marina Meila, Stefan Chmiela, and Alexandre Tkatchenko. Modern manifold learning methods for md data a step by step procedural overview. www.stat.washington.edu/mmp/Papers/mlcules-arxiv.pdf, July 2019.

[45] Ming-Yen Cheng and Hau-Tieng Wu. Local linear regression on manifolds and its geometric interpretation. *J. Am. Stat. Assoc.*, 108(504):1421–1434, 2013.

[46] Ming-Yen Cheng and Hau-Tieng Wu. Local linear regression on manifolds and its geometric interpretation. *J. Am. Stat. Assoc.*, 108(504):1421–1434, 2013.

[47] Stefan Chmiela, Alexandre Tkatchenko, Huziel E Sauceda, Igor Poltavsky, Kristof T Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Sci Adv*, 3(5):e1603015, May 2017.

[48] Stefan Chmiela, Huziel E Sauceda, Klaus-Robert Müller, and Alexandre Tkatchenko. Towards exact molecular dynamics simulations with machine-learned force fields. *Nat. Commun.*, 9(1):3887, September 2018.

[49] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 30(1):5–30, 2006.

[50] R. R. Coifman, S. Lafon, A. Lee, Maggioni, Warner, and Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. In *Proceedings of the National Academy of Sciences*, pages 7426–7431, 2005.

[51] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Appl. Comput. Harmon. Anal.*, 21(1):5–30, July 2006.

[52] P Constantine, E Dow, and Q Wang. Active subspace methods in theory and practice: Applications to kriging surfaces. *SIAM J. Sci. Comput.*, 36(4):A1500–A1524, January 2014.

[53] Miles Cranmer, Alvaro Sanchez-Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. June 2020.

[54] Nakul Verma Cse and U C San Diego. Towards an algorithmic realization of nash's embedding theorem.

[55] P. Das, M. Moll, H. Stamati, L.E. Kavraki, and C. Clementi. Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. *Proceedings of the National Academy of Sciences*, 103(26):9885–9890, 2006.

[56] P P N de Groen. An introduction to total least squares. May 1998.

[57] Siyuan L U Date: December. Isometric embedding of riemannian manifolds. `https://www.math.mcgill.ca/gantumur/math580f12/siyuan.lu.pdf`. Accessed: 2021-12-12.

[58] Carmeline J. Dsilva, Ronen Talmon, Neta Rabin, Ronald R. Coifman, and Ioannis G. Kevrekidis. Nonlinear intrinsic variables and state reconstruction in multiscale simulations. *The Journal of Chemical Physics*, 139(18):184109, 2013. doi: 10.1063/1.4828457. URL `https://doi.org/10.1063/1.4828457`.

[59] Carmeline J Dsilva, Ronen Talmon, Ronald R Coifman, and Ioannis G Kevrekidis. Parsimonious representation of nonlinear dynamical systems through manifold learning: A chemotaxis case study. *Appl. Comput. Harmon. Anal.*, 44(3):759–773, May 2018.

[60] Morris L Eaton. Group invariance applications in statistics. *Regional Conference Series in Probability and Statistics*, 1:i–133, 1989.

[61] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, and Others. Least angle regression. *Ann. Stat.*, 32(2):407–499, 2004.

[62] M.K. Elyaderani, S.Jain, J.M.Druce, S.Gonella, and J.D.Haupt. Improved support recovery guarantees for the group lasso with applications to structural health monitoring. *CoRR*, abs/1708.08826, 2017.

[63] Vittorio Erba, Marco Gherardi, and Pietro Rotondo. Intrinsic dimension estimation for locally undersampled data. *Sci. Rep.*, 9(1):17133, November 2019.

[64] Felix A Faber, Anders S Christensen, Bing Huang, and O Anatole von Lilienfeld. Alchemical and structural distribution based representation for universal quantum machine learning. *J. Chem. Phys.*, 148(24):241717, June 2018.

[65] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Stat. Assoc.*, 96(456):1348–1360, December 2001.

[66] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. October 2013.

[67] Charles Fefferman, Sergei Ivanov, Matti Lassas, and Hariharan Narayanan. Fitting a manifold of large reach to noisy data. October 2019.

[68] Andrew L Ferguson, Athanassios Z Panagiotopoulos, Pablo G Debenedetti, and Ioannis G Kevrekidis. Systematic determination of order parameters for chain dynamics using diffusion maps. *Proc. Natl. Acad. Sci. U. S. A.*, 107(31):13597–13602, August 2010.

[69] Giacomo Fiorin, Michael L Klein, and Jérôme Hénin. Using collective variables to drive molecular dynamics simulations. *Mol. Phys.*, 111(22-23):3345–3362, December 2013.

[70] Kelly L Fleming, Pratyush Tiwary, and Jim Pfaendtner. New approach for investigating reaction dynamics and rates with ab initio calculations. *J. Phys. Chem. A*, 120(2):299–305, January 2016.

[71] Richard A Friesner. Ab initio quantum chemistry: methodology and applications. *Proc. Natl. Acad. Sci. U. S. A.*, 102(19):6648–6653, May 2005.

[72] M Gastegger, L Schwiedrzik, M Bittermann, F Berzsenyi, and P Marquetand. wACSF-Weighted atom-centered symmetry functions as descriptors in machine learning potentials. *J. Chem. Phys.*, 148(24):241709, June 2018.

[73] Rong Ge, Chi Jin, Sham M Kakade, Praneeth Netrapalli, and Aaron Sidford. Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis. April 2016.

[74] C W Gear. Parameterization of non-linear manifolds. August 2012.

[75] Christopher R Genovese. Minimax manifold estimation. `https://jmlr.org/papers/volume13/genovese12a/genovese12a.pdf`, 2012. Accessed: 2021-5-20.

[76] Christopher R Genovese, Marco Perone-Pacifico, Isabella Verdinelli, and Larry Wasserman. Manifold estimation and singular deconvolution under hausdorff loss. September 2011.

[77] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[78] Charles R. Harris, K. Jarrod Millman, Stfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernndez del Ro, Mark Wiebe, Pearu Peterson, Pierre Grard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585:357362, 2020. doi: 10.1038/s41586-020-2649-2.

[79] Trevor Hastie and Werner Stuetzle. Principal curves. *J. Am. Stat. Assoc.*, 84(406): 502–516, June 1989.

[80] Trevor Hastie and Werner Stuetzle. Principal curves. *J. Am. Stat. Assoc.*, 84(406):502, June 1989.

[81] Trevor Hastie and Robert Tibshirani. *Statistical learning with sparsity : the lasso and generalizations*. Monographs on statistics and applied probability, no. 143. CRC Press, special indian ed. edition, 2015.

[82] Stefan Haufe, Vadim V Nikulin, Andreas Ziehe, Klaus-Robert Müller, and Guido Nolte. Estimating vector fields using sparse basis field expansions. In D Koller, D Schuurmans, Y Bengio, and L Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 617–624. Curran Associates, Inc., 2009.

[83] M Hein, J Y Audibert, and U Luxburg. Graph laplacians and their convergence on random neighborhood graphs. *J. Mach. Learn. Res.*, 2007.

[84] Matthias Hein, Jean-Yves Audibert, and Ulrike von Luxburg. From graphs to manifolds - weak and strong pointwise consistency of graph laplacians. In *Learning Theory, 18th Annual Conference on Learning Theory, COLT 2005, Bertinoro, Italy, June 27-30, 2005, Proceedings*, pages 470–485, 2005. doi: 10.1007/11503415_32. URL http://dx.doi.org/10.1007/11503415_32.

[85] Matthias Hein, Jean-Yves Audibert, and Ulrike von Luxburg. Graph laplacians and their convergence on random neighborhood graphs. *Journal of Machine Learning Research*, 8:1325–1368, 2007. URL http://dl.acm.org/citation.cfm?id=1314544.

[86] Eric J Heller. The correspondence principle and intramolecular dynamics. *Faraday Discuss. Chem. Soc.*, 75(0):141–153, January 1983.

[87] Charles A Herring, Amrita Banerjee, Eliot T McKinley, Alan J Simmons, Jie Ping, Joseph T Roland, Jeffrey L Franklin, Qi Liu, Michael J Gerdes, Robert J Coffey, and Ken S Lau. Unsupervised trajectory analysis of Single-Cell RNA-Seq and imaging data reveals alternative tuft cell origins in the gut. *Cell Syst*, 6(1):37–51.e9, January 2018.

[88] Tim Hesterberg, Nam Hee Choi, Lukas Meier, and Chris Fraley. Least angle and ℓ_1 penalized regression: A review. February 2008.

[89] Tim Hesterberg, Nam Hee Choi, Lukas Meier, and Chris Fraley. Least angle and ℓ_1 penalized regression: A review. February 2008.

[90] Huan Xu, C Caramanis, and S Mannor. Sparse algorithms are not stable: A No-Free-Lunch theorem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(1):187–193, January 2012.

[91] Huan Xu, C Caramanis, and S Mannor. Sparse algorithms are not stable: A No-Free-Lunch theorem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(1):187–193, January 2012.

[92] Bing Huang and O Anatole von Lilienfeld. Communication: Understanding molecular representations in machine learning: The role of uniqueness and target similarity. *J. Chem. Phys.*, 145(16):161102, October 2016.

[93] Bruno Levy Inria-Alice. Laplace-Beltrami eigenfunctions towards an algorithm that "understands" geometry. http://graphics.stanford.edu/courses/cs233-20-spring/ReferencedPapers/understand_geometry_01631196.pdf. Accessed: 2021-5-28.

[94] Sabine Van Huffel Ivan Markovsky. Overview of total least-squares methods. In *SIGNAL PROCESSING*, 2007.

[95] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. June 2018.

[96] Mainak Jas, Titipat Achakulvisut, Aid Idrizović, Daniel E Acuna, Matthew Antalek, Vinicius Marques, Tommy Odland, Ravi Prakash Garg, Mayank Agrawal, Yu Umegaki, Peter Foley, Hugo L Fernandes, Drew Harris, Beibin Li, Olivier Pieters, Scott Otterson, Giovanni De Toni, Chris Rodgers, Eva Dyer, Matti Hamalainen, Konrad Kording, and Pavan Ramkumar. Pyglmnet: Python implementation of elastic-net regularized generalized linear models, February 2020.

[97] Ian Jolliffee. A note on principal components in regression. *Applied Statistics*, 1982.

[98] Dominique Joncas, Marina Meila, and James McQueen. Improved graph laplacian via geometric Self-Consistency. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4457–4466. Curran Associates, Inc., 2017.

[99] R O Jones. Density functional theory: Its origins, rise to prominence, and future. *Rev. Mod. Phys.*, 87(3):897–923, August 2015.

[100] N Kambhatla and T K Leen. Dimension reduction by local principal component analysis. *Neural Comput.*, 1997.

[101] D G Kendall. A survey of the statistical theory of shape. *Stat. Sci.*, 1989.

[102] D G Kendall, D Barden, T K Carne, and H Le, editors. *Shape & Shape Theory*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ, USA, October 1999.

[103] David G Kendall. Shape manifolds, procrustean metrics, and complex projective spaces. *Bull. Lond. Math. Soc.*, 16(2):81–121, March 1984.

[104] Matthäus Kleindessner and U V Luxburg. Dimensionality estimation without distances. *AISTATS*, 2015.

[105] Matthäus Kleindessner and Ulrike von Luxburg. Dimensionality estimation without distances. In *AISTATS*, 2015.

[106] Magnus Borga Tomas Landelius Knutsson. A uni ed approach to PCA, PLS, MLR and CCA. http://www.diva-portal.org/smash/get/diva2:288565/FULLTEXT01.pdf. Accessed: 2021-11-17.

[107] Samson Koelle, Hanyu Zhang, Marina Meila, and Yu-Chia Chen. Manifold coordinates with physical meaning, 2021.

[108] Dhruv Kohli, Alexander Cloninger, and Gal Mishne. LDLE: Low distortion local eigenmaps. January 2021.

[109] Dhruv Kohli, Alexander Cloninger, and Gal Mishne. LDLE: Low distortion local eigenmaps. January 2021.

[110] Mario Krenn, Florian Häse, Akshatkumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn.: Sci. Technol.*, 1(4):045024, October 2020.

[111] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling Methods for the Nystrm Method. *Journal of Machine Learning Research*, pages 981–1006, Apr 2012.

[112] João Marcelo Lamim Ribeiro, Davide Provasi, and Marta Filizola. A combination of machine learning and infrequent metadynamics to efficiently predict kinetic rates, transition states, and molecular determinants of drug dissociation from G protein-coupled receptors. *J. Chem. Phys.*, 153(12):124105, September 2020.

[113] N P Landsman. Between classical and quantum? http://philsci-archive.pitt.edu/2328/1/handbook.pdf. Accessed: 2021-1-25.

[114] P. Langley. Crafting papers on machine learning. In Pat Langley, editor, *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pages 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

[115] Huiling Le and David G Kendall. The riemannian structure of euclidean shape spaces: A novel environment for statistics. *Ann. Stat.*, 21(3):1225–1271, 1993.

[116] Johannes Lederer and Christian Müller. Don't fall for tuning parameters: Tuning-free variable selection in high dimensions with the TREX. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligenc*, January 2015.

[117] Ann B Lee and Larry Wasserman. Spectral connectivity analysis. November 2008.

[118] John M. Lee. *Introduction to Smooth Manifolds.* Springer, 2003.

[119] John M Lee. *Introduction to Smooth Manifolds*. Springer, New York, NY, 2012.

[120] Elizaveta Levina and Peter J Bickel. Maximum likelihood estimation of intrinsic dimension. `https://www.stat.berkeley.edu/~bickel/mldim.pdf`, . Accessed: 2021-12-17.

[121] Elizaveta Levina and Peter J Bickel. Maximum likelihood estimation of intrinsic dimension. `https://www.stat.berkeley.edu/~bickel/mldim.pdf`, . Accessed: 2021-5-28.

[122] Siyuan Li, Haitao Lin, Zelin Zang, Lirong Wu, Jun Xia, and Stan Z Li. Invertible manifold learning for dimension reduction. October 2020.

[123] Siyuan Li, Haitao Lin, Zelin Zang, Lirong Wu, Jun Xia, and Stan Z Li. Invertible manifold learning for dimension reduction. In *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 713–728. Springer International Publishing, 2021.

[124] Zinan Lin, Kiran Koshy Thekumparampil, Giulia Fanti, and Sewoong Oh. InfoGAN-CR and ModelCentrality: Self-supervised model training and selection for disentangling GANs. June 2019.

[125] Yu Liu and Kris De Brabanter. Smoothed nonparametric derivative estimation using weighted difference quotients. *J. Mach. Learn. Res.*, 21(65):1–45, 2020.

[126] Chuanjiang Luo, Issam Safa, and Yusu Wang. Approximating gradients for meshes and point clouds via diffusion metric. *Comput. Graph. Forum*, 28(5):1497–1508, July 2009.

[127] Chuanjiang Luo, Issam Safa, and Yusu Wang. Approximating gradients for meshes and point clouds via diffusion metric. *Comput. Graph. Forum*, 28(5):1497–1508, July 2009.

[128] Thomas März and Colin B Macdonald. Calculus on surfaces with general closest point functions. *SIAM J. Numer. Anal.*, 50(6):3303–3328, January 2012.

[129] James Mc Queen, Marina Meila, and Dominique Perrault-Joncas. Nearly isometric embedding by relaxation.

[130] Marina Meila, Samson Koelle, and Hanyu Zhang. A regression approach for explaining manifold embedding coordinates. (1811.11891), 2018. URL `http://arxiv.org/abs/1811.11891`.

[131] Marina Meila, Samson Koelle, and Hanyu Zhang. A regression approach for explaining manifold embedding coordinates. November 2018.

[132] Nicolai Meinshausen. Relaxed lasso. *Comput. Stat. Data Anal.*, 52(1):374–393, September 2007.

[133] Nicolai Meinshausen and Peter Bühlmann. Stability selection: Stability selection. *J. R. Stat. Soc. Series B Stat. Methodol.*, 72(4):417–473, July 2010.

[134] Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. June 2008.

[135] Rachel Metz and CNN Business. How AI that reads emotions is changing the online classroom. *CNN*, February 2021.

[136] Igor Mezic. Spectrum of the koopman operator, spectral expansions in functional spaces, and state space geometry. February 2017.

[137] Kitty Mohammed and Hariharan Narayanan. Manifold learning using kernel density estimation and local principal components analysis. September 2017.

[138] Kitty Mohammed and Hariharan Narayanan. Manifold learning using kernel density estimation and local principal components analysis. *arxiv*, 1709.03615, 2017.

[139] Sayan Mukherjee and Qiang Wu. Estimation of gradients and coordinate covariation in classification. `https://jmlr.csail.mit.edu/papers/volume7/mukherjee06b/mukherjee06b.pdf`, 2006. Accessed: 2021-5-20.

[140] Sayan Mukherjee and Ding-Xuan Zhou. Learning coordinate covariances via gradients. *J. Mach. Learn. Res.*, 7(Mar):519–549, 2006.

[141] S B Myers and N E Steenrod. The group of isometries of a riemannian manifold. *Ann. Math.*, 40(2):400–416, 1939.

[142] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Found. Comut. Math.*, 17(2):527–566, April 2017.

[143] Frank Nielsen. An elementary introduction to information geometry. *Entropy*, 22(10), September 2020.

[144] Guillaume Obozinski, Martin J. Wainwright, and Michael I. Jordan. Support union recovery in high-dimensional multivariate regression. *The Annals of Statistics*, 39(1): 1–47, 2011. ISSN 00905364. URL `http://www.jstor.org/stable/29783630`.

[145] Shingo Oue. ON ASYMPTOTICS OF LOCAL PRINCIPAL COMPONENT ANALYSIS. *Hitotsubashi Journal of Commerce and Management*, 1996.

[146] Umut Ozertem. Locally defined principal curves and surfaces. *J. Mach. Learn. Res.*, 12:1249–1286, 2011.

[147] Gautam Pai, Ronen Talmon, Alex Bronstein, and Ron Kimmel. DIMAL: Deep isometric manifold learning using sparse geodesic sampling. November 2017.

[148] Shashank Pant, Zachary Smith, Yihang Wang, Emad Tajkhorshid, and Pratyush Tiwary. Confronting pitfalls of AI-augmented molecular dynamics using statistical physics. *J. Chem. Phys.*, 153(23):234118, December 2020.

[149] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, High-Performance deep learning library. December 2019.

[150] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[151] Dominique Perraul-Joncas and Marina Meila. Non-linear dimensionality reduction: Riemannian metric estimation and the problem of geometric discovery. May 2013.

[152] Dominique Perraul-Joncas and Marina Meila. Non-linear dimensionality reduction: Riemannian metric estimation and the problem of geometric discovery. May 2013.

[153] Patrick O Perry. Cross-Validation for unsupervised learning. September 2009.

[154] Steve Pischke. Lecture notes on measurement error.

[155] Nikita Puchkin and Vladimir Spokoiny. Structure-adaptive manifold estimation. June 2019.

[156] Haozhi Qi. ISONet: Deep isometric learning for visual recognition (ICML 2020).

[157] Lele Qu, Shimiao An, Tianhong Yang, and Yanpeng Sun. Group sparse basis pursuit denoising reconstruction algorithm for polarimetric Through-the-Wall radar imaging. *Int. J. Antennas Propag.*, 2018, August 2018.

[158] João Marcelo Lamim Ribeiro, Pablo Bravo, Yihang Wang, and Pratyush Tiwary. Reweighted autoencoded variational bayes for enhanced sampling (RAVE). *J. Chem. Phys.*, 149(7):072301, August 2018.

[159] M. A. Rohrdanz, W. Zheng, M. Maggioni, and C. Clementi. Determination of reaction coordinates via locally scaled diffusion map. *The Journal of chemical physics*, 134(12), 2011.

[160] Mary A. Rohrdanz, Wenwei Zheng, and Cecilia Clementi. Discovering mountain passes via torchlight: Methods for the definition of reaction coordinates and pathways in complex macromolecular reactions. *Annual Review of Physical Chemistry.64:295-316*, 64:295–316, 2013.

[161] Sam Roweis and Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.

[162] Samuel Rudy, Alessandro Alla, Steven L Brunton, and J Nathan Kutz. Data-Driven identification of parametric partial differential equations. *SIAM J. Appl. Dyn. Syst.*, 18 (2):643–660, January 2019.

[163] Lawrence K. Saul and Sam T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *J. Mach. Learn. Res.*, 4:119–155, December 2003. ISSN 1532-4435. doi: 10.1162/153244304322972667. URL `https://doi.org/10.1162/153244304322972667`.

[164] R Schoen and S Yau. Lectures on harmonic maps. *undefined*, 1997.

[165] S Schoenholz and J Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. *International*, 2018.

[166] Scott Shaobing Chen and David L. Donoho and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM REVIEW*, 43(1):129, February 2001.

[167] Zahra Shamsi, Kevin J Cheng, and Diwakar Shukla. Reinforcement learning based adaptive sampling: REAPing rewards by exploring protein conformational landscapes. *J. Phys. Chem. B*, 122(35):8386–8395, September 2018.

[168] Shan Shan. *Probabilistic Models on Fibre Bundles.* PhD thesis, Duke University, 2019.

[169] Weimin Sheng. Section 5. geodesics and the exponential map, December 2009.

[170] Zuoqiang Shi. Convergence of laplacian spectra from random samples. July 2015.

[171] Zuoqiang Shi. Convergence of laplacian spectra from random samples. July 2015.

[172] Ankita Shukla, Shagun Uppal, Sarthak Bhagat, Saket Anand, and Pavan Turaga. Geometry of deep generative models for disentangled representations. February 2019.

[173] Hythem Sidky, Wei Chen, and Andrew L Ferguson. Machine learning for collective variable discovery and enhanced sampling in biomolecular simulation. *Mol. Phys.*, 118 (5):e1737742, March 2020.

[174] A Singer. From graph to manifold laplacian: The convergence rate. *Science Direct*, May 2006.

[175] Justin S Smith, Benjamin T Nebgen, Roman Zubatyuk, Nicholas Lubbers, Christian Devereux, Kipton Barros, Sergei Tretiak, Olexandr Isayev, and Adrian E Roitberg. Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning. *Nat. Commun.*, 10(1):2903, July 2019.

[176] Christopher D Sogge. *Hangzhou Lectures on Eigenfunctions of the Laplacian*. Princeton University Press, 2014.

[177] Z. Szab, B. Pczos, and A. L?rincz. Online group-structured dictionary learning. In *CVPR 2011*, pages 2865–2872, 2011. doi: 10.1109/CVPR.2011.5995712.

[178] Yee Whye Teh and Sam T. Roweis. Automatic alignment of local representations. In *NIPS*, 2002.

[179] Trevor Hastie Robert Tibshirani. The lasso and generalizations. .

[180] Trevor Hastie Robert Tibshirani. The lasso and generalizations. `https://hastie.su.domains/StatLearnSparsity_files/SLS_corrected_1.4.16.pdf`, . Accessed: 2021-12-15.

[181] Daniel Ting, Ling Huang, and Michael I. Jordan. An analysis of the convergence of graph laplacians. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1079–1086, 2010. URL `http://www.icml2010.org/papers/554.pdf`.

[182] Daniel Ting, Ling Huang, and Michael Jordan. An analysis of the convergence of graph laplacians. January 2011.

[183] Z Trstanova, B Leimkuhler, and T Lelièvre. Local and global perspectives on diffusion maps in the analysis of molecular systems. *Proc. Math. Phys. Eng. Sci.*, 476(2233): 20190036, January 2020.

[184] A W van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 1998.

[185] Nakul Verma. Towards an algorithmic realization of nash's embedding theorem.

[186] Ulrike von Luxburg, Mikhail Belkin, and Olivier Bousquet. Consistency of spectral clustering. *aos*, 36(2):555–586, April 2008.

[187] Martin J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1 -constrained quadratic programming (lasso). *IEEE Transactions on Information Theory*, 55:2183–2202, 2009.

[188] Yihang Wang, João Marcelo Lamim Ribeiro, and Pratyush Tiwary. Past-future information bottleneck for sampling molecular reaction coordinate simultaneously with thermodynamics and kinetics. *Nat. Commun.*, 10(1):3573, August 2019.

[189] Larry Wasserman. *All of Nonparametric Statistics (Springer Texts in Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387251456.

[190] Kilian Q Weinberger and Lawrence K Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. `https://www.aaai.org/Papers/AAAI/2006/AAAI06-280.pdf`. Accessed: 2021-5-28.

[191] Caroline L Wormell and Sebastian Reich. Spectral convergence of diffusion maps: improved error bounds and an alternative normalisation. June 2020.

[192] Q Wu, J Guinney, M Maggioni, and S Mukherjee. Learning gradients: predictive models that infer geometry and statistical dependence. *J. Mach. Learn. Res.*, 2010.

[193] Qiang Wu, Sayan Mukherjee, and Feng Liang. Localized sliced inverse regression. *Adv. Neural Inf. Process. Syst.*, 21:1785–1792, 2008.

[194] Tian Xie, Arthur France-Lanord, Yanming Wang, Yang Shao-Horn, and Jeffrey C Grossman. Graph dynamical networks for unsupervised learning of atomic scale dynamics in materials. *Nat. Commun.*, 10(1):2667, June 2019.

[195] Ilker Yalcin and Yasuo Amemiya. Nonlinear factor analysis as a statistical method. *Statist. Sci.*, 16(3):275–294, 08 2001. doi: 10.1214/ss/1009213729. URL `https://doi.org/10.1214/ss/1009213729`.

[196] Greg Yang. Tensor programs ii: Neural tangent kernel for any architecture. June 2020.

[197] M Yuan and Y Lin. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Series B Stat. Methodol.*, 2006.

[198] S Zelditch. Quantum ergodicity and mixing of eigenfunctions. In *Encyclopedia of Mathematical Physics*, pages 183–196. Elsevier, 2006.

[199] Steve Zelditch. Eigenfunctions of the laplacian of riemannian manifolds.

[200] Igor Ying Zhang and Andreas Grüneis. Coupled cluster theory in materials science. *Frontiers in Materials*, 6:123, 2019.

[201] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and E, Weinan. Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.*, 120(14):143001, April 2018.

[202] Ruda Zhang and Roger Ghanem. Normal-bundle bootstrap. July 2020.

[203] Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J. Scientific Computing*, 26(1):313–338, 2004.

[204] Peng Zhao and Bin Yu. On model selection consistency of lasso. *J. Mach. Learn. Res.*, 7:2541–2563, 2006.

[205] Wenwei Zheng, Mary A Rohrdanz, and Cecilia Clementi. Rapid exploration of configuration space with diffusion-map-directed molecular dynamics. *J. Phys. Chem. B*, 117 (42):12769–12776, October 2013.

[206] Hui Zou. The adaptive lasso and its oracle properties. *J. Am. Stat. Assoc.*, 101(476): 1418–1429, December 2006.

[207] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Series B Stat. Methodol.*, 67(2):301–320, April 2005.