Thesis Proposal
High-Dimensional Analysis of Unsupervised Learning Problems

Martin Azizyan

May 2015

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Aarti Singh (Chair)
Larry Wasserman
Ryan Tibshirani
Robert Tibshirani (Stanford)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Copyright © 2015 Martin Azizyan
Abstract

High dimensional inference problems can present both statistical and computational challenges. In this thesis, we are concerned with developing computationally tractable methods for dimensionality reduction in unsupervised settings with optimal statistical properties.

First, we consider the problem of learning Gaussian Mixture Models (GMMs), under a sparsity assumption on the subspace separating the mixture components. Previous work has shown that, ignoring computational concerns, such sparsity conditions can reduce the sample complexity of mixture learning from polynomial to logarithmic in the dimension of the problem. Existing methods for sparse GMM learning either fail to guarantee such a reduction in sample complexity, or require the covariance of each mixture component to be spherical. We provide to our knowledge the first polynomial time algorithm which can resolve sparse Gaussian mixtures of non-spherical components, with statistical risk that is only logarithmic in ambient dimension. We validate the algorithm empirically on several synthetic datasets.

Second, we consider learning the principal subspace of a large set of vectors from an extremely small number of compressive measurements of each vector. We design and study an estimator based on back-projections of these compressive samples. We show that observing even a constant number of compressive measurement of each vector, we can consistently estimate the principal subspace to arbitrary precision, provided that the number of vectors is large. This result is achieved by a simple algorithm that computes the eigenvectors of an estimate of the covariance matrix. The main insight is to exploit an averaging effect that arises from applying a different random projection to each vector. We also consider the problem of detecting whether a point lies in a known subspace from compressive measurements, and design a method which succeeds with high probability given the number of compressive measurements is slightly larger than the dimension of the subspace.

Next, we create a minimax framework for analyzing the performance of semisupervised regression methods. We define a smoothness condition on the regression function in terms of a density-sensitive distance metric, which corresponds to assuming that the function is smoother in regions of the explanatory variables with higher density. We provide an estimator for this distance metric which utilizes only the unlabeled data, and show that kernel regression using this estimated metric has minimax optimal risk. Also, we characterize conditions under which any estimator which only uses the labeled data – i.e., a fully supervised method – is strictly outperformed by the semisupervised estimator. Intuitively, this occurs when the distribution of the explanatory variables lies on or near a low-dimensional manifold. Furthermore, we show that a cross-validation procedure can be used to adapt to an (unknown) strength parameter of the semisupervised smoothness assumption, so our method does not suffer when the assumption does not hold in practice.

Finally, we study the risk of density mode clustering – a nonparametric clustering method where the clusters are the basins of attraction of the modes of a density estimator. We show that the risk of mode clustering over the high-density regions of the clusters can be small even in high dimensions. We also state a low noise condition under which the overall clustering risk is small in high dimensions even beyond the high density regions.
# Contents

1 Introduction .......................................................... 1
  1.1 Completed Work ..................................................... 3
  1.2 Proposed Work ...................................................... 5
  1.3 Exploratory Work .................................................... 5
  1.4 Timeline .............................................................. 6

2 Sparse Gaussian Mixture Learning .................................... 7
  2.1 Introduction .......................................................... 7
  2.2 Problem Setup and Assumptions .................................... 10
  2.3 Proposed Method ..................................................... 11
      2.3.1 Algorithm HARDTPRICE .................................... 12
  2.4 Main Result .......................................................... 13
      2.4.1 Recovery of relevant features ................................. 14
      2.4.2 Relaxing model assumptions ................................ 14
  2.5 Simulation Results .................................................. 15
  2.6 Discussion ........................................................... 15

3 Compressive Subspace Estimation ..................................... 17
  3.1 Introduction .......................................................... 17
  3.2 Results ............................................................... 18
  3.3 Experiments .......................................................... 21
  3.4 Conclusion ........................................................... 22

4 Compressive Subspace Detection ....................................... 23
  4.1 Introduction .......................................................... 23
  4.2 Measurement Model and Test Statistic ............................. 24
  4.3 Main results .......................................................... 25
  4.4 Simulation results .................................................... 26
  4.5 Conclusion ........................................................... 27

5 Density-sensitive Semisupervised Regression ......................... 28
  5.1 Introduction .......................................................... 28
  5.2 Definitions ............................................................ 31
  5.3 Density-Sensitive Function Spaces ................................. 32
      5.3.1 The Exponential Metric ...................................... 32
      5.3.2 The Regression Function .................................... 32
      5.3.3 Properties of the Function Spaces ......................... 33
5.4 Semisupervised Kernel Estimator .................................................. 33
5.5 Minimax Bounds ........................................................................ 34
  5.5.1 The Class $\mathcal{P}_n$ ................................................................. 34
  5.5.2 Supervised Lower Bound ....................................................... 35
  5.5.3 Semisupervised Upper Bound ................................................. 35
  5.5.4 Comparison of Lower and Upper Bound .............................. 35
5.6 Adaptive Semisupervised Inference ............................................. 36
5.7 Simulation Results ....................................................................... 36
5.8 Discussion .................................................................................. 37

6 Risk Bounds for Mode Clustering ................................................. 39
  6.1 Introduction .............................................................................. 39
  6.2 Mode Clustering and Morse Theory .......................................... 40
    6.2.1 Morse Theory ..................................................................... 40
    6.2.2 Clusters ............................................................................ 42
  6.3 Estimated Clusters .................................................................... 42
  6.4 Bounding the Risk ................................................................... 43
    6.4.1 Stability ............................................................................. 43
    6.4.2 The Cluster Cores .............................................................. 44
    6.4.3 Bounding the Risk Over the Cores .................................... 44
    6.4.4 Beyond the Cores ............................................................... 45
    6.4.5 Gaussian Clusters .............................................................. 46
    6.4.6 Low Dimensional Analysis ............................................... 47
  6.5 Experiments ............................................................................. 48
  6.6 Conclusion ............................................................................... 49

7 Proposed and Exploratory Problems ............................................. 50
  7.1 Proposed Work ......................................................................... 50
    7.1.1 Sparse Gaussian Mixture Learning with $K > 2$ Components ................................................................................. 50
    7.1.2 Feature Selection for Mode Clustering .................................. 52
  7.2 Exploratory Work ..................................................................... 52
    7.2.1 Convex Sparse Clustering .................................................... 52
    7.2.2 Detecting Arbitrary Non-Gaussian Directions ..................... 54
    7.2.3 Approximation Algorithms for Sparse K-means .................. 55

Bibliography .................................................................................. 56
Chapter 1

Introduction

This thesis is concerned with inference and estimation problems which involve some form of unsupervised dimensionality reduction. Below, we introduce several problems we have considered, and summarize our contributions. Our focus for each of these problems is i) to identify an estimation algorithm which is practical, in the sense of computational complexity; and ii) to provide a statistical analysis of the performance of the estimators, ideally with guarantees of minimax optimality. The problems in question are sparse Gaussian mixture learning; compressive covariance estimation and subspace detection; semisupervised learning; and mode clustering.

The last few years have seen extensive research on developing computationally efficient and statistically sound methods that can leverage sparsity of the relevant feature set for supervised learning (classification, regression, etc.) of high-dimensional data, where learning and selection of relevant features is possible even when the number of training data $n$ is much less than the number of features $d$. Similar results for the unsupervised task of clustering are largely non-existent. While there have been recent attempts at clustering high-dimensional data and selecting relevant features, these either do not come with theoretical guarantees for practical (i.e., sub-exponential time) algorithms, or assume very strong conditions. In the latter category, previous work [11, 118] has considered the problem of clustering data drawn from a Gaussian mixture model (i.e., a weighted combination of Gaussian densities), and obtained guarantees allowing for high-dimensional inference under the assumption that the differences between component mean vectors are sparse. However, this work assumes that each component in the Gaussian mixture has the same spherical covariance, i.e., the mixture has the form $\sum_i p_i N(\mu_i, \sigma^2 I)$, where $p_i$ and $\mu_i$ are the mixture weights and means, and $\sigma^2$ is the variance. This assumption is quite strong, as it does not account for cases where the cluster means are insufficient to decide whether a feature is relevant for clustering, such as the example in Figure 1.1. Motivated by this example, we consider a simple non-spherical Gaussian mixture model for clustering high-dimensional data, and provide a computationally efficient algorithm for simultaneous feature selection and clustering, that comes with sample complexity guarantees that depend primarily on the number of relevant features (intrinsic dimension) and only logarithmically on the total number of features (ambient dimension), thus allowing for clustering in a high-dimensional setting.

We consider learning the principal subspace of a large set of vectors from an extremely small number of compressive measurements of each vector. Low rank approximation, popularly referred to as principal component analysis, is a fundamental preprocessing task in a wide variety of machine learning and signal processing applications. Given a set of vectors $x_1, \ldots, x_n \in \mathbb{R}^d$, low rank approximation aims to find a $k$-dimensional subspace of $\mathbb{R}^d$ (for given $k < d$) that captures the largest possible amount of energy present in the vectors $\{x_i\}_{i=1}^n$. Concatenating the vectors as columns of a $d \times n$ matrix $X$, this principal subspace is given by the span of the left singular vectors of $X$ corresponding to its $k$ largest singular values,
forming the best \( k \)-dimensional subspace to approximate \( X \) in many senses. For example, it minimizes the squared Euclidean reconstruction error, \( \|X - \Pi X\|_F^2 \), over all \( k \)-dimensional projections \( \Pi \). There are many situations where obtaining this best subspace is computationally challenging, motivating several different theoretical and empirical studies. A number of clever approaches, ranging from sparsification [2] and column sampling [24], to sketching and streaming techniques [49, 83] have been used to reduce the computational burden associated with computing the principal subspace when \( n \) and \( d \) are large. An equally pressing concern is the cost of data acquisition, motivating lines of work on low rank approximation from missing data and compressive measurements. The matrix completion literature focuses on finding approximations from randomly or adaptively sampled entries of the data matrix [56, 73, 96, 106].

With compressive measurements, only a few linear combinations of each column of \( X \) are acquired, which mathematically translates to observing \( Y = RX \) where \( R \) is an \( m \times d \) compression matrix with \( m \ll d \). Usually, the columns of \( Y \) are projected back into \( d \)-dimensions and the principal subspace of these vectors is used as an estimator [59, 84, 109]. In our contribution, we deviate from this strategy, and propose an algorithm that uses a different rank \( m \) random compression matrix per vector to approximate the principal subspace of the vectors \( \{x_t\}_{t=1}^n \). By using a different random projection per vector, we show that our algorithm only needs two compressive measurements per vector provided there are enough vectors, while existing approaches require \( O(k/\epsilon) \) measurements to achieve an \( \epsilon \)-approximation with target rank \( k \) [59].

Motivating applications for this measurement model include time-series analysis, and signal processing over a distributed sensor network with high measurement and communication cost.

Compressive subspace detection refers to the problem of deciding whether a high dimensional vector \( \in \mathbb{R}^d \) lies in an \( s \)-dimensional subspace \( S \), where \( s \ll d \), given few compressive measurements of the vector. This problem arises in several applications such as detecting anomalies [112], medical imaging [4], hyperspectral target detection [75], radar signal processing [104], interference estimation [93], etc. In these applications, the object of interest is described by a large number of features and the ability to detect them using only linear combination of the features (without the need to measure, store or compute the entire feature vector) is desirable. Related work has considered the cases where the subspace contains only the origin \( (S = \{0\}) \) [47, 62, 99], where instead of compressive measurements we observe the vector with entries missing at random [16], or when the subspace \( S \) takes on certain special forms [5]. We address the compressed measurement case with an arbitrary subspace of interest \( S \), when the vector is corrupted with Gaussian noise. We present a test statistic for this problem and demonstrate that, when the dimension of the subspace \( s \) is much smaller than the ambient dimension \( d \), the probability of error
of the proposed detector decreases exponentially in the number of compressive samples, provided that the energy of vectors off the subspace scales as \( d \). Using information-theoretic lower bounds, we demonstrate that no other detector can achieve the same probability of error for weaker signals.

Semisupervised methods are techniques for using labeled data \((X_1, Y_1), \ldots, (X_n, Y_n)\) together with unlabeled data \(X_{n+1}, \ldots, X_N\) to make predictions. This scenario is motivated by cases where obtaining the labels \(Y_i\) is significantly more expensive than obtaining unlabeled data, e.g., when \(X_i\) are images or webpages easily obtained from the Internet and \(Y_i\) are descriptions which must be labeled by humans. Semisupervised methods invoke some assumption that links the marginal distribution \(P_X\) of \(X\) to the regression function \(f(x) = \mathbb{E}(Y|X = x)\). It is common to assume that \(f\) is very smooth over high density regions of \(P_X\), e.g., that \(f\) is constant over clusters in \(P_X\) (known as the cluster assumption.), or that it is constant over low dimensional submanifolds (the manifold assumption.). Many of the methods are ad-hoc and have been shown to work in specific examples but are lacking a theoretical foundation.

Existing theoretical work [18, 19, 77, 95, 97, 107, 110, 111] has shown that the degree to which unlabeled data improves or degrades the performance is very sensitive to the cluster and manifold assumptions.

We consider a formalization of the semisupervised smoothness assumption, which links the regression function \(f\) to a family of density-based distance functions which depend on the marginal distribution \(P_X\). Based on this class of functions, we derive a framework for establishing conditions under which our proposed semisupervised estimator strictly outperforms all possible fully supervised estimators, i.e., estimators which only make use of the labeled data. We also show that our estimator is robust to the strength of the semisupervised smoothness assumption, so that if the assumption does not hold in practice, then the risk of our estimator is no worse than that of any fully supervised estimator.

Density mode clustering is a nonparametric method for using density estimation to find clusters [7, 30, 40, 42]. Clusters are defined as the basins of attraction of a mode, i.e., the set of points where the gradient ascent path ends at the mode. Mode clustering is usually implemented using the mean-shift algorithm [40, 42, 53], which is a popular method in computer vision [43]. The algorithm is analyzed in [7], where pointwise convergence results are obtained. Namely, [7] show that given a fixed point \(x_0\), the gradient ascent path starting at \(x_0\) of a density estimator converges to that of the true density. Our contribution is an analysis of the overall risk of density mode clustering, in terms of how pairs of points are clustered under the true density versus the estimated density. We show that the cluster risk over the cluster cores – the high density portion of the basins – is exponentially small, independently of dimension. We also show that if a certain low noise assumption holds then the cluster risk outside the cluster cores is small.

This document is organised as follows. In this chapter we provide a summary of completed and proposed work, with a timeline for the thesis. In Chapter 2 we present our results on sparse Gaussian mixture learning and elaborate on the directions for extending the work. We consider compressive covariance estimation and compressive subspace detection in Chapters 3 and 4. Chapter 5 describes our framework for analyzing semisupervised inference with density-sensitive distance metrics. In Chapter 6 we analyze the risk of mode clustering under conditions which characterize the separation of the clusters, and describe some potential directions for incorporating feature selection. Finally, in Chapter 7 we give some background for the problems stated as proposed and exploratory work.

1.1 Completed Work

- **Sparse Gaussian mixture learning.** We consider the problem of learning Gaussian mixtures under sparsity assumptions in high dimensions, i.e., when the number of data points may be much smaller than the number of dimensions. Our main focus is the development of computationally tractable al-
gorithms for sparse mixture learning, with provable statistical benefits over ordinary mixture learning methods. Specifically, we consider a Gaussian mixture model with two non-spherical Gaussian components, where the Fisher’s linear discriminant distinguishing the clusters is sparse. In contrast, existing sparse mixture learning methods require the mixture components to have spherical covariances, which is a significant restriction.

The method we propose is a combination of a recent approach for learning parameters of a Gaussian mixture model and sparse linear discriminant analysis (LDA). We show that the error rate of this method is bounded by $O(\text{poly}(s \log(d)/n))$, where $d$ is the ambient dimension and $s$ is the sparsity level, i.e., the number of relevant features for separating the mixture components. In contrast, the typical rate for ordinary mixture learning algorithms is $O(\text{poly}(d/n))$.

• **Compressive subspace estimation.** We consider the problem of estimating the principal subspace of a collection of vectors given highly compressed measurements of each vector. We state and study an estimator based on back-projections of these compressive samples. Specifically, we show that the covariance matrix, and hence the principal subspace, of a collection of $n$ vectors in $d$ dimensions can be estimated to arbitrary precision from the projections of each vector on independent random $m$-dimensional subspaces, for any $2 \leq m < d$, given sufficiently large $n$. Specifically, our bound says that the effective sample complexity of the problem is scaled by a factor of $m^2/d^2$, and we provide a number of simulations confirming our theoretical results.

• **Compressive subspace detection.** We consider the problem of detecting whether a high dimensional vector $\in \mathbb{R}^d$ lies in a $s$-dimensional subspace $S$, where $s \ll d$, given $m$ compressive measurements of the vector. We present a test statistic for subspace detection using compressive samples and demonstrate that the probability of error of the proposed detector decreases exponentially in the number of compressive samples $m$, provided that the energy off the subspace scales as $d$. Using information-theoretic lower bounds, we demonstrate that no other detector can achieve the same probability of error for weaker signals.

• **Density-sensitive semisupervised regression.** We consider the problem of estimating a regression function using a mixture of labeled and unlabeled data, under a semisupervised smoothness assumption on the function given in terms of the marginal distribution of the explanatory variables. This assumption includes a strength parameter $\alpha$, which interpolates between the typical Lipschitz-type assumptions used in fully supervised regression, and the very strong cluster assumptions used elsewhere in the semisupervised learning literature. Specifically, we assume that the regression function varies smoothly with respect to a family of density-sensitive distance metrics. We provide a minimax optimal estimator for this problem, which makes use of the unlabeled data to estimate the density-based metric, and subsequently performs kernel regression using the estimated metric. Also, we characterize conditions under which any estimator which only uses the labeled data – i.e., a fully supervised method – is strictly outperformed by the semisupervised estimator. Intuitively, this occurs when the distribution of the explanatory variables lies on or near a low-dimensional manifold. We also show that a cross-validation procedure can be used to adapt to the (unknown) strength of the semisupervised smoothness condition $\alpha$.

• **Statistical risk bounds for density mode clustering.** Mode clustering assigns points to clusters based on the basins of attraction of the modes of a density estimator. We study the risk of mode-based clustering under smoothness and separation assumptions on the true data generating density. We define cluster cores as regions around modes where the density is high, and we show that the clustering risk within the cores is exponentially small even in high dimensions. We also characterize a low noise condition on the density near the boundaries of the basins of attraction, under which the
overall clustering error – even beyond the cores – is small in high dimensions.

1.2 Proposed Work

- **More general sparse mixture learning.** The current state-of-the-art sparse Gaussian mixture learning methods allow for more than two spherical mixture components or at most two non-spherical mixture components, the latter of which is achieved by our method. There are significant obstacles to extending the sparse learning methods in each setting to handle multiple non-spherical components. Namely, the straightforward translation of our method to multiple components would have computational and statistical complexity which scales exponentially in the number of components (see Section 7.1.1). One approach is to make more restrictive sparsity assumptions than the condition on the Fisher discriminant we take, such as assuming that all irrelevant features must be (marginally) independent of all relevant features. Another approach is to sacrifice consistency in favor of approximating the mixture, resulting in a computational-statistical tradeoff.

- **Feature selection for mode clustering.** While feature selection methods have been proposed for mixture learning and K-means clustering, to our knowledge no sparse density clustering approaches exist. We propose considering conditions under which removing a feature does not affect cluster assignments for any points. For instance, a simple sufficient condition for irrelevance of a feature is that i) it is distributed independently of the other features, and ii) the marginal density of the feature is unimodal. We seek to characterize other and less restrictive conditions to guarantee feature irrelevance, and design methods which can identify such features.

1.3 Exploratory Work

- **Statistical limits of convex sparse clustering.** Recently, several authors have analyzed algorithmic and statistical properties of convex clustering, which can be seen as a convex relaxation of a K-means type objective. We can modify this approach to incorporate feature selection, resulting in a convex sparse clustering method. However, empirical results suggest that this approach does not have a strong statistical advantage over ordinary convex clustering, even when the true data generating distribution contains feature-sparse clusters. A possible direction of future research is to establish whether or not this is an insurmountable statistical limitation of the approach.

- **Finding non-Gaussian directions, with power.** There exist several methods, such as projection pursuit and non-Gaussian component analysis, which, given a multivariate dataset, aim to find directions in it along which the data distribution is clearly non-Gaussian. These methods rely on various measures of Gaussianity, none of which appear to be consistent in separating non-Gaussian distributions. We propose using the Maximum Mean Discrepancy (MMD) approach to two sample testing, which can be shown to be consistent against all alternatives, to search for non-Gaussian directions in data.

- **Approximation algorithms for sparse K-means.** The only algorithms available for solving sparsi-fied K-means objectives, such as Sparcl, consist of iterative block coordinate descent type methods. These require initialization with some pre-computed cluster assignment of the data, and our empirical results suggest that this initialization strongly affects the quality of the resulting approximation of the true optimum of the objective. Exploring other approximation algorithms for sparsi-fied K-means objectives is another potential direction of research.
1.4 Timeline

The timeline for the proposed work is as follows:

1. **Spring 2015**: thesis proposal;
2. **Summer 2015**: extensions to sparse Gaussian mixture learning;
3. **Summer-Fall 2015**: sparse mode clustering;
4. **Fall 2015**: projection pursuit with MMD;
5. **Fall 2015-Spring 2016**: analysis of convex sparse clustering; sparse K-means algorithms;
6. **Spring 2016**: thesis writing and defense.
Chapter 2

Sparse Gaussian Mixture Learning

We consider the problem of clustering data points in high dimensions, i.e., when the number of data points may be much smaller than the number of dimensions. Specifically, we consider a Gaussian mixture model (GMM) with two non-spherical Gaussian components, where the clusters are distinguished by only a few relevant dimensions. The method we propose is a combination of a recent approach for learning parameters of a Gaussian mixture model and sparse linear discriminant analysis (LDA). In addition to cluster assignments, the method returns an estimate of the set of features relevant for clustering. Our results indicate that the sample complexity of clustering depends on the sparsity of the relevant feature set, while only scaling logarithmically with the ambient dimension. Further, we require much milder assumptions than existing work on clustering in high dimensions. In particular, we do not require spherical clusters nor necessitate mean separation along relevant dimensions. (For technical details including proofs, see [14].)

2.1 Introduction

The last few years have seen extensive research on developing computationally efficient and statistically sound methods that can leverage sparsity of the relevant feature set for supervised learning (classification, regression, etc.) of high-dimensional data. These methods show that learning and selection of relevant features is possible even when the number of training data $n$ is much less than the number of features $d$, which is typically the case for high-dimensional data. However, similar results for the unsupervised task of clustering are largely non-existent. The task of clustering high-dimensional data and extracting relevant features arises routinely in many applications, e.g., clustering of patients based on gene expression profiles and identifying the relevant genotypes, grouping web content and identifying relevant characteristics, clustering proteins with similar drug expression profiles, etc.

While there have been recent attempts at clustering high-dimensional data and selecting relevant features, these either do not come with theoretical guarantees or assume very strong conditions that suggest that even employing marginal feature selection, using projections of the data onto individual coordinates, as a pre-processing step before clustering might suffice. Thus, while supervised learning in high dimensions requires single-step methods that can perform the learning task and select relevant features simultaneously, it is not clear whether a sophisticated single-step approach is necessary for clustering in high dimensions.

A simple example which demonstrates that pre-processing the data using a marginal (coordinate-wise) feature selection step does not suffice for clustering, is provided by a mixture of two non-spherical Gaussian components (see Figure 2.1). It is clear that $x_1$ is relevant to define the clusters, however the marginal distribution of the data when projected onto $x_1$ is a single unimodal Gaussian. Hence, marginal
Figure 2.1: An example of two clusters where both features $x_1$ and $x_2$ are relevant to define the clusters, however the marginal distribution of data points along $x_1$ is unimodal, and hence no marginal feature selection method can work.

feature selection cannot be used to identify the relevant features.

Motivated by this example, we consider a simple non-spherical Gaussian mixture model (defined formally in the next section) for clustering high-dimensional data, and aim to provide a computationally efficient algorithm for simultaneous feature selection and clustering, that comes with sample complexity guarantees that depend primarily on the number of relevant features (intrinsic dimension) and only logarithmically on the total number of features (ambient dimension).

**Related work.** Before we describe our approach and results, we discuss related work in some more detail. Sparse clustering methods that perform feature selection for high-dimensional data have received attention recently.

K–means based approaches begin with the typical K–means objective and introduce some sparsity-inducing penalties [3, 34, 58, 113, 117]. While the penalization introduced in these papers is convex (akin to supervised learning approaches), the K–means objective itself is non-convex and in fact NP-hard. Thus, in general, solving any of these objectives is NP-hard and the papers propose iterative approaches akin to Llyod algorithm for solving the K–means objective. Moreover, these papers do not provide any statistical guarantees, with the exception of [34, 113]. The latter two papers do provide some consistency results, however these are for the true objective optimizers only which are NP-hard. Moreover, the notion of relevant features considered in all these papers is that the means are separated along each relevant feature, which may not necessarily be the case as demonstrated in Figure 2.1.

Another non-parametric approach to feature selection for clustering that is consistent in high dimensions is presented in [33], however it relies on pre-screening features which appear marginally unimodal, again failing for the example in Figure 2.1.

Learning Gaussian mixture models (GMMs) has a long history, particularly in computer science theory community, where the emphasis has been on relaxing the assumptions under which GMMs can be learnt under various metrics such as estimating the distribution, parameters or clustering [1, 36, 68]. However, these papers primarily focus on computational tractability and mostly have high sample complexity, particularly in high dimensions. For example, the most relevant to our work is the paper on learning non-spherical GMMs where the components are separated by a hyperplane [25], however it has sample complexity that depends as $d^4$ on the ambient dimension. The proposed estimator relies on first making the data isotropic (zero mean and overall identity covariance). This is achieved by pre-whitening the data by multiplying it with the inverse sample covariance matrix. However, in high dimensions when the number of samples drawn from the mixture $n \ll d$ (the number of features), the sample covariance matrix is
not invertible and hence the method cannot succeed. Moreover, no work in this line, to the best of our knowledge, addresses feature selection. There is a very recent work [60] where the question of optimal sample complexity for GMM parameter estimation in $\ell_\infty$ norm is addressed and we build on this paper to provide clustering and feature selection guarantees.

Apart from the work in the computer science theory community, multiple statistical approaches have also been proposed to learning Gaussian mixture models in high dimensions and feature selection [50, 54, 72, 76, 78, 80, 88, 90, 91, 92, 98, 103]. These employ various sparsity assumptions, e.g. that the components are spherical and have sparse mean vectors, or that the covariance matrices (or their inverses) are also sparse, etc. However, as the K–means based methods, these approaches either a) require approximating maximum likelihood parameters without providing efficient algorithms; or b) do not come with precise finite sample statistical properties of the estimators.

Assuming mixtures of equal weight spherical components with sparse mean separation, [11] provide some minimax bounds for the problem with sample complexity that scales with the number of relevant features and only logarithmically with the total number of features. Similar statistical guarantees are obtained by [118] for learning mixtures of more than two spherical components with sparse mean vectors. However, the assumption of spherical components necessitates that relevant features are characterized by mean separation, and hence the results do not apply for cases like the one described in Figure 2.1.

Under less restrictive assumptions on the components, [6] analyze detection of high-dimensional Gaussian mixtures (vs. a single Gaussian as null) and selection of sparse set of features along which mean separation occurs, from a minimax perspective. Their minimax optimal estimators involve combinatorial search, and while the authors also investigate some tractable procedures, they are either based on marginal feature selection or assume that the component covariance matrices are known and diagonal.

Finally, we mention that if the cluster assignments are known, the problem of clustering reduces to binary classification. And specifically, clustering using a mixture of two identical covariance Gaussians reduces to linear discriminant analysis, if the cluster assignments are known. Feature selection using a sparse linear discriminant analysis has been analyzed in [26]. We will leverage this approach, in combination with the results for $\ell_\infty$ parameter estimation for GMMs [60] to demonstrate a method and results for sparse clustering and feature selection under high-dimensional non-spherical GMMs.

**Contributions.** Our contributions can be summarized as follows:

- We present a computationally efficient method for clustering in high dimensions that comes with finite sample guarantees on the misclustering rate. Our results show that sample complexity scales quadratically with the number of relevant features (inherent dimension), but only logarithmically with total number of features (ambient dimension). As a result, the proposed method enables learning of non-spherical Gaussian mixtures of two components in high dimensions (when the number of data points may be much smaller than the number of features), without assuming sparsity of the covariance or inverse covariance matrix.

- We provide guarantees for feature selection under a very generalized notion of relevant features that does not require that clusters necessarily have mean separation along the relevant features. This allows us to handle cases like that shown in Figure 2.1.

The rest of the chapter is organized as follows. In section 2.2, we formalize our setup. The proposed method combining ideas from [60] and [26] is presented in section 2.3. Section 2.4 states our results on misclustering rate, sample complexity and feature selection in high dimensions. Experimental results described in section 2.5 on some simulated datasets demonstrate the viability of our proposed method. We discuss some open directions in section 2.6.
2.2 Problem Setup and Assumptions

Inspired by Figure 2.1, we consider the following simple model.

A1) Data generating model: The data points $X_1, \ldots, X_n$ are generated i.i.d. from a mixture of two Gaussians of the form $\frac{1}{2}N(\mu_1, \Sigma) + \frac{1}{2}N(\mu_2, \Sigma)$ in $\mathbb{R}^d$.

The assumption that the components have equal weight and equal covariance is made largely for exponential simplicity. For the reasons discussed in section 2.4.2, we believe that extending our results to allow for arbitrary mixture weights and differing component covariances is possible without introducing any major technical issues, and should involve no more than some additional bookkeeping. In fact, in section 2.4) we demonstrate a successful application of our proposed approach to a mixture with unequal weights and covariances. On the other hand, extending to more than two mixture components is a significant challenge, and addressing it is out of the scope of this work.

The error of a clustering $\psi : \mathbb{R}^d \rightarrow \{1, 2\}$ is defined as follows. Let $X$ be a random draw from the true mixture, and let $Y \in \{1, 2\}$ be the (latent) label of the mixture component from which $X$ was drawn, i.e., $Y - 1 \sim \text{Bern}(\frac{1}{2})$ and $X|Y \sim N(\mu_Y, \Sigma)$. We define the overlap of the clustering $\psi$ as $\Upsilon(\psi) := \min_\pi \mathbb{P}(\psi(X) \neq \pi(Y))$ where the minimum is over permutations $\pi : \{1, 2\} \rightarrow \{1, 2\}$, and the error of $\psi$ is defined as $L(\psi) := \Upsilon(\psi) - \min_{\psi'} \Upsilon(\psi')$.

We define the optimal clustering $\psi^* := \arg\min_\psi \Upsilon(\psi)$, which coincides with the Bayes optimal classifier in the supervised problem of predicting $Y$ from $X$:

$$\psi^*(x) = \begin{cases} 1 & \text{if } (\mu_0 - x)^T \beta < 0, \\ 2 & \text{o.w.} \end{cases}$$

(2.1)

where $\beta = \Sigma^{-1} \Delta_\mu$, $\mu_0 = \frac{\mu_1 + \mu_2}{2}$ and $\Delta_\mu = \frac{\mu_1 - \mu_2}{2}$. Notice that the Bayes optimal decision boundary is linear and hence the problem corresponds to linear discriminant analysis (LDA).

If the labels are known, one can simply plug-in sample estimates of class conditional means $\hat{\mu}_Y$ and covariance matrix $\hat{\Sigma}$ to obtain an empirical classification rule. In clustering, the labels are latent. However, if we can learn the parameters $\mu_1, \mu_2, \Sigma$ of the Gaussian mixture model, we can plug these in and obtain a similar empirical clustering.

In the high-dimensional setting ($n \ll d$), estimates of the covariance matrix are typically not invertible, necessitating some additional assumptions to make the problem well-posed. In high-dimensional clustering, it is natural to expect that not all features are relevant for clustering. For example, in clustering proteins based on their drug expression profiles, not all drugs are responsible for differentiation of the proteins. This assumption can be captured as follows (using the notation $[d] = \{1, \ldots, d\}$).

A2) Sparsity of relevant features: The set of relevant features $S \subseteq [d]$, which are given by the non-zero coordinates of $\beta$, satisfy $|S| \leq s$, where $s \leq d$ is the sparsity level.

This notion of feature relevance is motivated by the fact that the optimal clustering $\psi^*$ in Eq. 2.1 depends on a given feature only when the corresponding coordinate of $\beta$ is non-zero.

We will demonstrate that the sample complexity of clustering in high dimensions depends on the number of non-zero coordinates $|\beta|_0 = |S| \leq s$, and only logarithmically on the total number of features $d$.

In comparison, existing work on high-dimensional clustering typically assumes $\Delta_\mu$ is sparse, and the relevant features are given by its non-zero coordinates, i.e., the coordinates along which mean separation occurs. So, they cannot identify relevant features such as $x_1$ in Figure 2.1. Also, some existing work on high-dimensional GMM learning assumes sparsity of the covariance $\Sigma$ or its inverse $\Sigma^{-1}$. These assumptions used in previous work are more restrictive than (and can be considered special cases of) our
notion of relevant features (nonzero coordinates of $\beta \equiv \Sigma^{-1} \Delta \mu$) which is precisely what the optimal clustering function depends on.

We make the following additional assumption which guarantees success of our computationally feasible method that uses the $\ell_1$ penalty.

A3) Restricted eigenvalue property: The covariance matrix $\Sigma$ satisfies

$$\min_{S \subseteq [d]: |S| \leq s, v \neq 0} \left\{ \frac{\| \Sigma v \|_2^2}{\| v \|_2^2} : \| v_S \|_1 \leq \| v \|_1 \right\} \geq \eta > 0$$

where $v_S$ is the projection of $v$ onto the coordinates in $S$, and $S^c = [d] \setminus S$ is the complement of $S$. A similar condition is required for feature selection in supervised learning using $\ell_1$ penalties (c.f. [21]). This condition ensures that there cannot exist two different values of the sparse vector $\beta$ which correspond to similar values for $\Sigma \beta = \Delta \mu$, and hence that a small error in estimating the parameters (either $\Delta \mu$ or $\Sigma$) must imply small clustering and feature selection error.

While the above assumptions suffice to evaluate the clustering performance in high dimensions, we also seek to correctly identify the set of relevant features. For this, we need to assume that each relevant feature is “relevant enough” to be detectable using a finite sample. Formally,

A4) Signal strength along each relevant feature: For each $i \in [d]$ such that $\beta(i) \neq 0$, let $|\beta(i)| \geq \beta_{\min}$, where $\beta_{\min} > 0$.

### 2.3 Proposed Method

Given samples $X_1, \ldots, X_n$ from the unknown mixture $\frac{1}{2} \mathcal{N}(\mu_1, \Sigma) + \frac{1}{2} \mathcal{N}(\mu_2, \Sigma)$, we propose a procedure composed of three stages. First we acquire initial estimates of mixture parameters using the algorithm of Hardt and Price [60]. Next we estimate the discriminating direction $\beta := \Sigma^{-1} \Delta \mu$ by means of solving a convex program analogous to the proposal of Cai and Liu [26] for sparse supervised linear classification. Finally we threshold the elements of the estimate of $\beta$ to recover the relevant features.

Precisely, the steps are as follows.

1. Obtain estimates $\hat{\mu}_1, \hat{\mu}_2, \hat{\Sigma}$ by invoking Algorithm HARDTPRICE defined in Section 2.3.1 with $\epsilon, \delta$ satisfying $\epsilon = C(\log(dn/\delta)/n)^{1/6}$ for some constant $C$.

2. For some $\lambda > 0$, set

$$\tilde{\beta}_\lambda = \arg\min_{z \in \mathbb{R}^d} \| z \|_1$$

subject to $\| \hat{\Sigma}z - \hat{\Delta}_\mu \|_\infty \leq \lambda$

where $\hat{\Delta}_\mu = \frac{\hat{\mu}_1 - \hat{\mu}_2}{2}$, $\| \cdot \|_\infty$ is the elementwise absolute maximum, and $\lambda$ is a tuning parameter the choice of which is discussed below. Let $\hat{\mu}_0 = \frac{\hat{\mu}_1 + \hat{\mu}_2}{2}$. The estimated clustering is defined as

$$\hat{\psi}_\lambda(x) = \begin{cases} 1 & \text{if} \quad (\hat{\mu}_0 - x)^T \tilde{\beta}_\lambda < 0, \\ 2 & \text{otherwise}. \end{cases}$$

Proposition 2.1 in Section 2.4 ties the error in the estimates of $\hat{\Sigma}$ and $\hat{\Delta}_\mu$ to the error of $\hat{\psi}_\lambda$. In this result, the bound on the clustering error is minimized when $\lambda$ takes on the smallest value such that the true $\beta$ is a feasible point of the constraints in (2.2). A specific value for $\lambda$ is given in Corollary 2.3, based on a few additional technical assumptions.
3. Estimate the relevant features $S$ by thresholding $\hat{\beta}_\lambda$:

$$\hat{S} = \{ i : \hat{\beta}_\lambda(i) > c \cdot \lambda \sqrt{s} \}$$

where $c > 0$ is a constant.

Our results for support recovery hold when $c > 2/\eta$.

2.3.1 Algorithm HARDTPRICE

We describe below the algorithm HARDTPRICE proposed by Hardt and Price [60] (section 3, algorithm B), simplified in accordance to the assumption (A1). Specifically, the version of the algorithm we state here skips the steps necessary to learn different component variances and the mixture weights.

The HARDTPRICE algorithm assumes the availability of another algorithm, GMFitLOWDIM, for mixture learning in a low-dimensional setting. The latter is any algorithm that, like HARDTPRICE, takes as input a set of samples as described in (A1) together with parameters $\epsilon, \delta > 0$, and has the same properties as those of HARDTPRICE stated in Theorem 2.2, but only for up to 2 dimensional mixtures.

Hardt and Price give a candidate for GMFitLOWDIM, which combines a type of moment method approach and a grid search over parameters. The algorithm involves a large number of steps, and we do not restate it here due to the space constraint. Since much of the computational and statistical difficulty of general Gaussian mixture learning is not present when only considering such small dimensional cases, we believe this does not take away from the exposition. In fact, in our simulation experiments (section 2.5) we successfully use an “off-the-shelf” EM based maximum likelihood mixture learning algorithm as GMFitLOWDIM.

**Input:** Samples $X_1, \ldots, X_n \in \mathbb{R}^d$, $\epsilon, \delta > 0$.

1. Set $\hat{V} = \max_{i \in [d]} \sum_{j=1}^n \frac{X_j(i)^2}{n} - \left( \sum_{j=1}^n \frac{X_j(i)}{n} \right)^2$, $\epsilon^* = \frac{\epsilon}{20} \cdot \delta^* = \frac{\delta}{100\pi^2}$. Algorithm GMFitLOWDIM will always be invoked with parameters $\epsilon^*$ and $\delta^*$.

**Estimate $\hat{\mu}_1$ and $\hat{\mu}_2$:**

2. For each $i \in [d]$, use GMFitLOWDIM on the univariate data $X_1(i), \ldots, X_n(i)$ obtaining estimates of the means $\xi_1(i)$ and $\xi_2(i)$.

3. If $|\xi_1(i) - \xi_2(i)| \leq \epsilon \hat{V}/4$ for all $i \in [d]$, put $\hat{\mu}_1 = \hat{\mu}_2 = \xi_1$ (and skip step 4).

4. Otherwise, let $i$ be the smallest index such that $|\xi_1(i) - \xi_2(i)| > \epsilon \hat{V}/4$ and, for each $j \in [d] \setminus \{i\}$ do:
   a) Apply GMFitLOWDIM to the bivariate data $[X_1(i), X_1(j)], \ldots, [X_n(i), X_n(j)]$ to obtain mean estimates $(\nu_k(i), \nu_k(j))$ for $k = 1, 2$.
   b) Let $k \in \{1, 2\}$ such that $|\xi_1(i) - \nu_k(i)| \leq \epsilon \hat{V}/10$. If such $k$ does not exist, the algorithm terminates with failure.
   c) Set $\hat{\mu}_1(j) = \nu_k(j)$ and $\hat{\mu}_2(j) = \nu_{3-k}(j)$.

**Estimate $\hat{\Sigma}$:**

5. For each $i \in [d]$, invoke GMFitLOWDIM on the univariate data $X_1(i), \ldots, X_n(i)$ and obtain an estimate of the diagonal element $\hat{\Sigma}(i; i)$.

6. For each $i < j$, invoke GMFitLOWDIM on the bivariate data $[X_1(i), X_1(j)], \ldots, [X_n(i), X_n(j)]$ and obtain an estimate of $\hat{\Sigma}(i, j) = \hat{\Sigma}(j, i)$.

7. Return $\hat{\mu}_1, \hat{\mu}_2, \hat{\Sigma}$.

Intuitively, the algorithm works as follows. Note that the marginal of a Gaussian mixture is the same as a mixture of the marginals of the mixture components. So, given sufficient data, the univariate mixture
mean estimates $\xi_1(i)$ and $\xi_2(i)$ learned in step 2 of the algorithm will be close to $\mu_1(i)$ and $\mu_2(i)$ for all $i \in [d]$, up to ordering. I.e., having $\xi_1(i)$ and $\xi_2(i)$, it remains only to decide whether $\xi_1(i)$ corresponds to the same mixture component as $\xi_1(j)$, or to $\xi_2(j)$ instead, for each other $j \in [d]$. To do this, in step 4 the algorithm looks at bivariate marginals of feature pairs $i,j$, and matches $\xi_1(i)$ to whichever one of $\xi_1(j)$ and $\xi_2(j)$ it co-occurs with in the bivariate marginal.

Similarly, if the component covariances are identical as we assume, then we only need to look at bivariate marginals to get $\Sigma(i,i)$, $\Sigma(i,j)$, and $\Sigma(j,j)$. Hardt and Price allow for different component covariances as well, where, similarly to learning the means, it is necessary to decide if $\Sigma_1(i,j)$ belongs to the same component as $\Sigma_1(k,l)$ or $\Sigma_2(k,l)$. To do this, Hardt and Price make use of up to 4-dimensional marginals.

In the first stage, the algorithm fits mixtures independently to the univariate projections of the data (step 2). However, it is important to note that the subsequent steps, which use bivariate projections of the data, recover information that cannot be obtained by purely marginal methods. For instance, the marginal method of [11] would fail to identify feature $x_1$ in the example in Figure 2.1 as relevant to the mixture, even with infinite data. The Hardt and Price algorithm, on the other hand, succeeds in this case, as demonstrated in section 2.5.

### 2.4 Main Result

Our first result states that if the parameters of the Gaussian mixture model in (A1) can be learnt accurately in $\ell_\infty$ norm, then the misclustering rate of the proposed method is small.

**Proposition 2.1.** Assume (A1). For any $\epsilon$, if \[ \max \left( \| \mu_1 - \hat{\mu}_\pi(1) \|_\infty^2, \| \mu_2 - \hat{\mu}_\pi(2) \|_\infty^2, \| \Sigma - \hat{\Sigma} \|_\infty \right) \leq \epsilon \] for some permutation $\pi : \{1, 2\} \rightarrow \{1, 2\}$, and if $\| \beta \|_1 + \sqrt{\epsilon} \leq \lambda$, then

\[ L(\hat{\psi}_\lambda) \leq \phi \left( \max \left( \frac{\Delta_\mu^T \Sigma^{-1} \Delta_\mu - \epsilon_1}{\sqrt{\Delta_\mu^T \Sigma^{-1} \Delta_\mu + \epsilon_2}}, 0 \right) \right) \frac{\epsilon_1 + \epsilon_2}{\sqrt{\Delta_\mu^T \Sigma^{-1} \Delta_\mu}} \]

where $\epsilon_1 = (2\lambda + 3\sqrt{\epsilon}) \| \beta \|_1$, $\epsilon_2 = \epsilon \| \beta \|_1^2 + 3(\lambda + \sqrt{\epsilon}) \| \beta \|_1$, and $\phi$ is the standard normal density.

This result is similar to the classical results in classification error analysis of Fisher’s Linear Discriminant, but with the key difference that the misclustering rate is bounded in terms of the $\ell_\infty$ norms of the errors of the parameter estimates. This is crucial, as it will subsequently allow us to obtain a rate that is only logarithmic in the ambient dimension $d$. Before giving the proof, we notice that the misclustering rate depends on $\Delta_\mu^T \Sigma^{-1} \Delta_\mu$ which can be regarded as the signal energy.

The following result from [60] provides us $\ell_\infty$ control over the GMM parameters.

**Theorem 2.2** (Hardt and Price [60]). Given $\epsilon, \delta > 0$ and $n$ samples from the model (A1), if

\[ n = O \left( \frac{1}{\epsilon^2 \log \left( \frac{d}{\delta \log \left( \frac{1}{\epsilon} \right)} \right)} \right), \]

then, with probability at least $1 - \delta$, Algorithm HARDTPRICE in Section 2.3.1 produces estimates $\hat{\mu}_1, \hat{\mu}_2$ and $\hat{\Sigma}$ such that, for some permutation $\pi : \{1, 2\} \rightarrow \{1, 2\}$,

\[ \max_{i=1,2} \left( \| \mu_i - \hat{\mu}_\pi(i) \|_\infty^2, \| \Sigma - \hat{\Sigma} \|_\infty \right) \leq \epsilon \left( \frac{1}{4} \| \mu_1 - \mu_2 \|_\infty^2 + \| \Sigma \|_\infty \right). \]

Combining Proposition 2.1 and Theorem 2.2, we have the following result under (A2) and (A3). We defer the proof to the supplement.
Corollary 2.3. Assume (A1), (A2), (A3), $\|\Sigma\|_2 \leq D_0$, and $\|\mu_1 - \mu_2\|_\infty^2 < D$. Given $\delta > 0$, there is some constant $c_1$ such that, setting

$$\lambda = c_1 \left( \frac{\log(dn/\delta)}{n} \right)^{1/6} \sqrt{D_0 s (\Delta\Sigma^{-1}\Delta)} \left( \frac{\log(dn/\delta)}{n} \right)^{1/12} + \sqrt{c_1} \left( \frac{\log(dn/\delta)}{n} \right)^{1/12},$$

with probability at least $1 - \delta$,

$$L(\hat{\psi}_\lambda) \leq C_0 \phi \left( \sqrt{D_0 s (\Delta\Sigma^{-1}\Delta)} \left( \frac{\log(dn/\delta)}{n} \right)^{1/6} \sqrt{s} \left( \frac{\log(dn/\delta)}{n} \right)^{1/12} \right),$$

for some constant $C_0$.

Remark: Hence it follows that $n = \Omega \left( s^6 \log(d) \right)$, suppressing the dependence on other parameters.

2.4.1 Recovery of relevant features

We derive a bound for $\|\beta - \hat{\beta}_\lambda\|_\infty$ under (A3) and then guarantee recovery of relevant features under (A4), i.e., when the non-zero components of $\beta$ are large enough.

Theorem 2.4. Assume the conditions of Proposition 2.1 hold. We have

$$\|\beta - \hat{\beta}_\lambda\|_\infty \leq \frac{2\lambda \sqrt{s}}{\eta}.$$

If, in addition, (A4) holds, $\eta > 2/c$, and $\beta_{\text{min}} > 2c\lambda\sqrt{s}$, then

$$\hat{S} = S.$$

Corollary 2.5. Assume (A1)-(A4). Under the conditions of Corollary 2.3, given $\delta > 0$, if

$$\beta_{\text{min}} = \omega \left( s \left( \frac{\log(dn/\delta)}{n} \right)^{1/12} \right),$$

then $\hat{S} = S$ with probability at least $1 - \delta$.

Thus, in order to recover the support, we require at least $n = \Omega \left( (s/\beta_{\text{min}})^{12} \log(d) \right)$, suppressing the dependence on other parameters.

2.4.2 Relaxing model assumptions

When $\Sigma_1 \neq \Sigma_2$, the true clustering $\psi^*$ defined in section 2.2 is no longer equivalent to the Bayes optimal decision rule for the latent classification problem, since the latter is in general quadratic. Hence, our results on the clustering error (Corollary 2.3) do not have a straightforward extension to the unequal covariance case.

However, the quantity $\beta := \Sigma_w^{-1}\Delta$, where now $\Sigma_w = p_1\Sigma_1 + p_2\Sigma_2$, does still have a meaningful interpretation when $\Sigma_1 \neq \Sigma_2$. Namely, it is the Fisher discriminant direction, which has been used in the Gaussian mixture learning literature (e.g. [25]) as a key quantity of interest to estimate, since projecting on this direction maximizes the between cluster variance. Corollary 2.5, which establishes a sample complexity for estimating the support of the vector $\beta$ which is logarithmic in the dimension, depends only on the estimation of $\Delta \Sigma$ and $\Sigma_w$ with $\ell_{\infty}$ norm control on the error. Hardt and Price [60] do provide results with precisely such a guarantee for learning Gaussian mixtures of the more general form $p_1N(\mu_1, \Sigma_1) + p_2N(\mu_2, \Sigma_2)$. Hence, the same proof technique should lead to an analogous result without assuming that $p_1 = p_2 = \frac{1}{2}$ or $\Sigma_1 = \Sigma_2 = \Sigma$. 

14


2.5 Simulation Results

In this section, we show the viability of our proposed method using simulated datasets. We implemented a version of the Hardt and Price algorithm, with minor modifications motivated by practicality, as well as a linear programming formulation for the problem in (2.2). We used the EM-based maximum likelihood Gaussian mixture learning algorithm in the R package EMCluster [38] as the subroutine GMFITLOWDIM in the Hardt and Price algorithm. R code is available in the supplementary material.

The first simulated dataset captures the issue depicted in Figure 2.1. We set \( d = 50 \), \( \Delta \mu = (2, 0, \ldots, 0) \), and \( \Sigma = I \) except for \( \Sigma(1, 2) = \Sigma(2, 1) = 0.85 \). Clearly, the true support of \( \beta = \Sigma^{-1} \Delta \mu \) is \( \{1, 2\} \), so there are \( s = 2 \) relevant variables. After sampling \( n = 200 \), we first apply the EM algorithm in the EMCluster package to the full dimensional dataset, and use the estimated parameters as input to (2.2). The results (Figure 2.2a) clearly show that this approach fails to identify the relevant features. In contrast, using the Hardt and Price parameter estimates (Figure 2.2b), the relevant features stand out for a wide range of values for \( \lambda \).

The second dataset is high-dimensional – we draw \( n = 150 \) samples from a mixture with \( d = 200 \), where we set \( \beta \) to have \( s = 5 \) non-zero coordinates each set to 5, \( \Sigma \) was generated randomly from a Wishart distribution with degrees of freedom \( 2d \) and subsequently rescaled to have eigenvalues in \( [0.5, 2] \), and \( \mu_1 = -\mu_2 = \Sigma \beta \). The results in Figure 2.2c show the importance of the sparsifying effect from using \( \lambda > 0 \) in (2.2), since the coefficients of the relevant features are not isolated until \( \lambda \) is increased to \( \approx 4.5 \), at which point each of the 5 relevant features is identified.

Finally, the third dataset serves to show that the equal component weight and equal component covariance assumptions are not crucial to the proposed method, when using the original algorithm of Hardt and Price rather than the simplified version in section 2.3.1. Specifically, we violate the model in (A1) and draw \( n = 500 \) samples from \( p_1N(\mu_1, \Sigma_1) + p_2N(\mu_2, \Sigma_2) \), where \( p_1 = 0.4 = 1 - p_2 \), \( d = 100 \), \( \Sigma_1 \neq \Sigma_2 \) each generated independently as above, \( \beta \) is set to have \( s = 5 \) non-zero values each of which is 2, and \( \mu_1 = -\mu_2 = (p_1 \Sigma_1 + p_2 \Sigma_2) \beta \). As demonstrated by the results in Figure 2.2d, the proposed method is indeed applicable to this more general setting.

2.6 Discussion

The primary goal of this work was to demonstrate a method for high-dimensional clustering which, in contrast to existing work, provably identifies relevant features that are not distinguished by the marginal separation of component means alone. The method we present is computationally feasible and statis-
cally efficient with sample complexity that primarily depends on the number of relevant dimensions, and only logarithmically on the total number of features. However, this goal was achieved by considering a very simple model - a mixture of two non-spherical Gaussians with same covariance and mixture weights. While we believe that it will be straightforward to adapt our results to allow uneven mixture weights and different covariance matrices, extending our approach to handle more than two components is a difficult problem and is the topic of ongoing work. Theoretically, the bounds we have demonstrated can be tightened in a few places, particularly for support recovery using a primal-dual witness argument. Additionally, it will be interesting to demonstrate matching lower bounds for this problem to establish optimality of the sample complexity.
Chapter 3

Compressive Subspace Estimation

We consider learning the principal subspace of a large set of vectors from an extremely small number of compressive measurements of each vector. Our theoretical results show that even a constant number of measurements per column suffices to approximate the principal subspace to arbitrary precision, provided that the number of vectors is large. This result is achieved by a simple algorithm that computes the eigenvectors of an estimate of the covariance matrix. The main insight is to exploit an averaging effect that arises from applying a different random projection to each vector. We provide a number of simulations confirming our theoretical results. (For technical details including proofs, see [74].)

3.1 Introduction

Low rank approximation is a fundamental preprocessing task in a wide variety of machine learning and signal processing applications. Given a set of vectors \( x_1, \ldots, x_n \in \mathbb{R}^d \), low rank approximation, popularly referred to as principal component analysis, aims to find a low dimensional subspace of \( \mathbb{R}^d \) that captures a large amount of the energy present in the vectors \( \{x_t\}_{t=1}^n \). It is well known that if we concatenate the vectors as columns of a \( d \times n \) matrix \( X \), then the span of the left singular vectors corresponding to the \( k \) largest singular values of \( X \) forms the best \( k \)-dimensional subspace in many senses. For example, it minimizes the squared Euclidean reconstruction error, \( \|X - \Pi X\|_F^2 \), over all \( k \)-dimensional projections \( \Pi \).

There are many situations where obtaining this best subspace is computationally challenging, motivating several different theoretical and empirical studies. A number of clever approaches, ranging from sparsification [2] and column sampling [24], to sketching and streaming techniques [49, 83] have been used to reduce the computational burden associated with computing the principal subspace when \( n \) and \( d \) are large.

An equally pressing concern is the cost of data acquisition, motivating lines of work on low rank approximation from missing data and compressive measurements. The matrix completion literature focuses on finding approximations from randomly or adaptively sampled entries of the data matrix [56, 73, 96, 106]. With compressive measurements, only a few linear combinations of each column of \( X \) are acquired, which mathematically translates to observing \( Y = RX \) where \( R \) is a \( m \times d \) compression matrix with \( m \ll d \). Usually, the columns of \( Y \) are projected back into \( d \)-dimensions and the principal subspace of these vectors is used as an estimator [59, 84, 109]. In this paper, we deviate from this strategy, and propose an algorithm that uses a different sensing operation per column to approximate the principal subspace of the vectors \( \{x_t\}_{t=1}^n \).

This difference, on one hand, adds some complexity to the algorithm, as a new random projection
provided there are enough columns, while existing approaches require $O(k/\epsilon)$ measurements to achieve an $\epsilon$-approximation with target rank $k$ [59].

Before proceeding, let us briefly mention two motivating applications. In time-series analysis, typically we have extremely long data sequences and would like to extract some signal from the sequence to assist in some inferential task. In many of these problems, observing any time-point of the data sequence is expensive; it may require performing some scientific experiment. Our results show that one can significantly undersample each time point of the series, which can tremendously reduce measurement overhead, while still extracting the principal subspace from the data sequence.

A related application is in signal processing over a distributed sensor network with high measurement and communication cost. Suppose each of the $n$ sensors records a $d$-dimensional vector of measurements, and our goal is to make inferences about the $d \times n$ matrix of measurements. Since measurement costs are high, one might be interested in compressive sampling at each sensor, but if the compression operator is shared across the sensors, an expensive synchronization step must be performed before data acquisition. Our method avoids the need for synchronization, since each sensor uses its own compression operator, while guaranteeing a good approximation to the matrix of measurements.

Remark. Since completing this work, we have designed an improved estimator which is consistent even with $m = 1$ measurements per vector, and we have shown matching upper bounds and information-theoretic lower bounds for the convergence rate of the covariance and subspace estimators. We are in the process of preparing these results for presentation.

### 3.2 Results

We first set up some notation used throughout the manuscript. We are interested in recovering the principal $k$-dimensional subspace $\Pi$ of the set of vectors $x_1, \ldots, x_n \in \mathbb{R}^d$ which we will concatenate into a matrix $X \in \mathbb{R}^{d \times n}$. We assume that each vector $x_t$ has bounded norm, that is $\max_{t \in [n]} \|x_t\|_2 \leq \mu^1$. Let $\Sigma = \frac{1}{n} \sum_{t=1}^{n} x_t x_t^T$ be the covariance matrix and notice that the optimal $k$-dimensional subspace $\Pi$, in terms of minimizing squared Euclidean reconstruction error, is spanned by the top-$k$ eigenvectors of $\Sigma$. A key parameter that governs the performance of our algorithm is the eigengap $\gamma_k$, defined as the difference between the $k$th largest eigenvalue and the $k + 1$st largest eigenvalue of the covariance matrix $\Sigma$.

We measure the error of an estimate $\hat{\Pi}$ for the principal subspace in terms of the spectral norm $\|\hat{\Pi} - \Pi\|_2$, where $\|M\|_2$ is the largest singular value of $M$. It is not too hard to show that when $\hat{\Pi}$ and $\Pi$ are projection matrices, $\|\hat{\Pi} - \Pi\|_2$ corresponds to the sine of the largest principal angle between the two subspaces, and therefore it is an appropriate measure of error in our setting.

We are interested in a compressive sensing framework, in which for some $m \geq 1$, we observe each vector $x_t$ through $2m$ compressive measurements $y_{it} = a_{it}^T x_t$, $i \in [2m]$ for vectors $a_i$ drawn independently from the $d$-dimensional unit sphere. This is equivalent to choosing a $2m$-dimensional projection $\Phi_t \in \mathbb{R}^{d \times 2m}$ uniformly at random and observing $y_{it} = \Phi_t x_t$. In our algorithm, we will write this as two $m$-dimensional random projections $\Phi_t$ and $\Psi_t$ per column with observations $y_{it} = \Phi_t x_t$ and $z_t = \Psi_t x_t$. To connect the two representations, let $A_t$ denote the $d \times m$ matrix whose columns are the vectors $a_{it}, i \in [m]$ and define $\Phi_t = P_{A_t}$, the the projection onto the span of the columns $A_t$. Define $B_t$ similarly to $A_t$, but with the second $m$ vectors and let $\Psi_t = P_{B_t}$.

We use $[n]$ to denote $\{1, \ldots n\}$.
Algorithm 1 Compressive Subspace Learning

**Input:** Compression parameter $m$, target rank $k$.

$\hat{\Sigma} = 0 \in \mathbb{R}^{d \times d}$

**for** $x_t \in \mathbb{R}^d$ in the data stream **do**

Let $\Phi_t, \Psi_t$ be $m$-dimensional random projections.

Acquire $y_t = \Phi_t x_t$ and $z_t = \Psi_t x_t$.

Update $\hat{\Sigma} = \hat{\Sigma} + \frac{1}{2}(y_t z_t^T + z_t y_t^T)$

**end for**

**Output:** $\hat{\Pi} = \text{span}(u_1, \ldots, u_k)$ the top $k$ eigenvectors of $\hat{\Sigma}$.

Our algorithm, which we call Compressive Subspace Learning (CSL), is conceptually quite simple (See Algorithm 1). For each vector $x_t$, we observe $y_t = \Phi_t x_t, z_t = \Psi_t x_t$ and form an estimate for $C_t = x_t x_t^T$ with:

$$\hat{C}_t = \frac{1}{2}(y_t z_t^T + z_t y_t^T) \quad (3.1)$$

We estimate the covariance $\Sigma$ with $\hat{\Sigma} = \frac{1}{n} \sum_{t=1}^n \hat{C}_t$ and use the span of top $k$ eigenvectors of $\hat{\Sigma}$ to estimate for the principal subspace $\Pi$. Let $\hat{\Pi}$ denote the output of our algorithm, namely the span of the top $k$ eigenvectors of $\hat{\Sigma}$. Note that $\hat{\Sigma}$ needs to be appropriately rescaled before it provides a reasonable estimate of $\Sigma$, but since we are only interested in the principal subspace, this normalization is only necessary for analysis.

In addition to the statistical guarantee in Theorem 3.1 below, we also mention some practical considerations. The algorithm can be implemented in the streaming model with each column vector $x_t$ being streamed through memory. In this model, the algorithm requires $O(d^2)$ time to process each vector and $O(d^2)$ space in total, to store $\hat{\Sigma}$.

The distributed sensor network model we described earlier is quite appealing for CSL. Recall that we had a network of $n$ sensors, each observing a $d$-dimensional vector $x_t$, with high measurement and communication cost. A naïve procedure that observes and transmits the vectors in full has $O(nd)$ communication overhead, but also $O(nd)$ measurement cost, which can be prohibitively expensive. The traditional compressive sampling approach of using a single random projection needs only $O(md)$ measurements but synchronizing the projection before data acquisition requires $O(nmd)$ communication cost. This is non-negligible overhead as one typically requires $m \asymp k/\epsilon$ to achieve error $\epsilon$ with target rank $k$ [59].

Using CSL, each sensor can generate its own random projection, take $2m$ measurements of its signal $x_t$, and send only two $d$-dimensional vectors (namely $y_t$ and $z_t$) over the network. Thus, CSL makes $O(mn)$ measurements while suffering only $O(nd)$ communication overhead, achieving the best properties of both other approaches. Moreover, provided $n$ is large, CSL can succeed even with $m$ constant, which is significantly smaller than other methods.

Our main result is the following statistical guarantee on the performance of the algorithm:

**Theorem 3.1.** With probability $\geq 1 - \delta$:

$$||\hat{\Pi} - \Pi||_2 \leq \frac{1}{\gamma_k} \left( \sqrt{\frac{44 \mu^2}{nm} \log(d/\delta)} + \frac{2 \mu d^2}{3 m^2 n} \log(d/\delta) \right) \quad (3.2)$$

So that one can achieve spectral norm error $\leq \epsilon$ provided that:

$$n \geq \max \left\{ \frac{176 \mu^2 \log(d/\delta)}{m \gamma_k^2 \epsilon^2}, \frac{4 \mu d^2}{3 \gamma_k \epsilon m^2} \log(d/\delta) \right\} \quad (3.3)$$
Some remarks are in order.

We are interested in the setting where $m$ is small compared to $d$, in which case the second term in Equation 3.2 is active. Thus, in interpreting the theorem, one should focus on the second terms in both bounds to see the dependence on $d$ and $m$. On the other hand, the relationship between $n$ and $\epsilon$ is dictated by the first terms.

Note that since the theorem holds for any $m \geq 1$, even two compressive measurements per column suffice to approximate the principal subspace, provided that $n \gg d$, which is common in a number of signal processing and machine learning applications. This is in sharp contrast with a number of other results on matrix approximation from compressive measurements, where the same projection is used on each column, in which case such a bound is not possible unless $m \geq k/\epsilon$ [59]. The justification for this is that using different random projections allows our algorithm to exploit averaging across the columns to capture the principal directions of the matrix. On the other hand, if the same projection is used, the measurements across columns become highly correlated and one does not experience a law-of-large-numbers phenomenon.

To make this concrete, consider the setting where $X$ is a rank one matrix with identical columns. If the same random projection is used across the columns, then there is no hope of approximating the principal direction with small $m$, irrespective of $n$. On the other hand, our bound says that by leveraging additional randomness in the data acquisition process, we can achieve an $\epsilon$-approximation provided that $n$ is large. In Figure 3.1 we demonstrate this phenomenon.

Results of this flavor are usually stated in terms of the number of measurements per column that suffice to achieve error $\epsilon$. Stated this way, our bound says that:

$$m \approx \sqrt{n} \cdot \frac{d}{\epsilon \gamma_k} \cdot \log(d/\delta)$$

is sufficient. In comparison, Halko et. al only require $m \approx k/\epsilon$ to achieve error $\epsilon$ with target rank $k$ [59] while the state-of-the-art results on matrix completion require observing $k \log(d/\delta)$ coordinates per column [96, 106]. Both sets of results are better than ours when $k$ is small compared to $d/\sqrt{n}$. Note that the method of Halko et. al does not compress the columns via a uniformly distributed projection, but rather projects onto a subspace computed by first compressing the rows of the matrix. While this approach can lead to better approximation, it is unfortunately not possible in many settings, including the distributed sensor network application mentioned above.

Our result is similar in spirit to a recent analysis by Gonen et. al who study the subspace learning problem when only a small number of entries of each column are observed [56]. They also show that one can approximate the principal subspace of a matrix $X$ using only a few observations per column.
Their analysis is somewhat simpler than ours, due to the fact that the distribution induced by the random projection operator is analytically more challenging that the subsampling operator. Interestingly, they also show that one cannot learn the subspace when observing only one coordinate per column, which may be related to why we require two compressive measurements, although of course the sampling paradigm is quite different.

It may seem curious that the target rank $k$ is absent from the bound, but notice that there is an interaction between $k$, the bound on the column norms $\mu$, and the eigengap. The target rank indirectly has some influence on the sample complexity $n$ via both the eigengap $\gamma_k$ and the length bound $\mu$.

### 3.3 Experiments

We complement our theoretical study of the compressive subspace learning algorithm with a number of simulations. Recall that our main goal is to capture the dependence between the error $\epsilon$ and the number of columns $n$. We do not believe that our results are tight in their dependence on the other parameters $d, m, \gamma_k, \mu$. However we do verify qualitatively the influence of the other parameters on the error.

Our first simulation is a comparison between CSL, the compressive singular value decomposition algorithm of Halko et al [59], and the naïve algorithm that uses the same compression operator on each column. The main difference between our algorithm and the other two is that we use a different $m$-dimensional projection per column, which leads to better performance as demonstrated in Figure 3.1. In that Figure, we plot the error $||\hat{\Pi} - \Pi||_2$ as a function of the number of columns $n$, for two different choices of $m$. The matrix is a rank $k = 3$ matrix corrupted with Gaussian noise and $d = 100$.

As predicted by Theorem 3.1, the error for our algorithm converges quickly to zero with $n$, with better convergence for larger $m$. On the other hand, the other two algorithms do not enjoy any performance improvements as $n$ increases. This simulation gives empirical evidence for the averaging effect that is formalized by our theoretical results.

We also verify that the rate of convergence in Theorem 3.1 is substantiated by empirical simulations. To this end, in Figure 3.2, we plot both the error alongside the rescaled error $\sqrt{n}||\hat{\Pi} - \Pi||_2$ as a function of $n$ across several problem settings. In the left two plots we consider the exactly low rank case, while on the right two plots we consider matrices that have some non-zero eigengap $\gamma_k$, but are not exactly low rank.

The first thing to notice is that in all simulations, $\sqrt{n}||\hat{\Pi} - \Pi||_2$ does appear to level out to some constant value, which demonstrates that our algorithm does converge at $n^{-1/2}$ rate. Moreover, in comparing between the trials, we see that increasing the dimensionality degrades the performance of the algorithm. On the other hand, increasing the target rank $k$ seems to have less influence.

We study the effect of $d, m, k$ and $\gamma_k$ in the bottom row of Figure 3.2. Our interpretation of the results is somewhat more qualitative, as we do not expect our theory to precisely capture the dependence between the error and these parameters. In the first figure, we see that increasing the number of measurements per column $m$ significantly improves the performance of the algorithm, as one might expect. It does appear that the dependence is $\epsilon \propto 1/m$, rather than the inverse-quadratic dependence in Theorem 3.1. In terms of problem dimension $d$, plotted in the second figure, the dependence appears to be linear rather than quadratic as predicted by Theorem 3.1.

As we mentioned earlier, there is no explicit dependence on the target rank $k$ in Theorem 3.1, although it does play some role indirectly through $\gamma_k$ and $\mu$. However, in the third plot from the left of the bottom row of Figure 3.2, we clearly see a linear dependence between $k$ and $\epsilon$. We suspect that a more careful analysis of our algorithm can explicitly introduce the dependence on $k$ while possibly removing one factor of $d$, leading to a bound that is more reminiscent of existing matrix completion and compression
Figure 3.2: Top row: Rates of convergence alongside rescaled rates $\sqrt{n} \times ||\hat{\Pi} - \Pi||_2$ across a number of problem settings. Left two plots show exactly low rank case while the right two plots show low rank approximation of high rank matrices. Bottom row: Influence of other problem parameters on error. From left to right: number of measurements per column ($m$), problem dimension ($d$), target rank ($k$), and eigengap ($\gamma_k$).

Lastly we plot the effect of the eigengap $\gamma_k$ on the error. We comment on two main effects: (1) increasing $\gamma_k$ improves the performance of the algorithm, and (2) for fixed eigengap, the performance improves with $n$. This qualitatively justifies $\gamma_k$ in the denominator of Equation 3.2.

3.4 Conclusion

In this manuscript, we demonstrate how one can approximate the principal subspace of a data matrix from very few compressive measurements per column. The main insight is that by using an independent random compression operator on each column, we can effectively take averages across columns, which preserves the signal but diminishes the noise stemming from the compression.

These results are practically and theoretically appealing yet several challenges still remain. First, our simulations suggest that the error should scale linearly with $d$ and $k$ and inversely linearly with $m$, yet our error bound has worse dependences. We would like to improve these dependences so that the theoretical results are more predictive of our experimental findings. Secondly, as we mentioned before, in theory, our algorithm requires at least two compressive measurements per column, although we did not observe such a requirement in simulation. While Gonen et al proved that such a requirement is necessary in the missing data setting, their justification does not immediately carry over to our setting [56]. It would be worthwhile to understand the differences between these sampling paradigms and to show either that two compressive measurements per column are necessary or that one suffices.

Lastly, it is not clear what the fundamental limits are for the compressive subspace learning problem. Given the compressed measurements $y_t, z_t$ for each column $x_t$, is there a lower bound on the error achievable by any algorithm? Is there an algorithm that achieves, or nearly achieves, this bound?

We hope to address these questions in future work.
Chapter 4

Compressive Subspace Detection

We consider the problem of detecting whether a high dimensional vector \( x \in \mathbb{R}^n \) lies in a \( r \)-dimensional subspace \( S \), where \( r \ll n \), given few compressive measurements of the vector. This problem arises in several applications such as detecting anomalies, targets, interference and brain activations. In these applications, the object of interest is described by a large number of features and the ability to detect them using only linear combination of the features (without the need to measure, store or compute the entire feature vector) is desirable. We present a test statistic for subspace detection using compressive samples and demonstrate that the probability of error of the proposed detector decreases exponentially in the number of compressive samples, provided that the energy off the subspace scales as \( n \). Using information-theoretic lower bounds, we demonstrate that no other detector can achieve the same probability of error for weaker signals. Simulation results also indicate that this scaling is near-optimal. (For technical details including proofs, see [10].)

4.1 Introduction

We study the problem of detecting whether a high-dimensional vector \( x \in \mathbb{R}^n \) lies in a known low-dimensional subspace \( S \), given few compressive measurements of the vector. The problem of testing whether a vector lies within a subspace is relevant for several tasks such as anomaly detection [112], medical imaging [4], hyperspectral target detection [75], radar signal processing [104], interference estimation [93], etc.

In high-dimensional settings, it is desirable to acquire only a small set of compressive measurements of the vector instead of measuring every coordinate. This reduces the amount of storage, communication, and computation needed. Recent advances in compressed sensing show that it is possible to reconstruct the original vector from few compressive measurements without significant loss in accuracy, provided that the vector is sparse or lies in a low-dimensional subspace. However, in some applications (as mentioned above) the objective is not to reconstruct the vector, but to detect whether the vector sensed using compressive measurements lies in a low-dimensional subspace or not. In this paper, we address this question and show that it is possible to detect when a compressively sampled vector lies in a known low-dimensional subspace using few compressive measurements. Notice that we cannot first reconstruct the vector and then test whether it lies in the subspace as the reconstruction might be arbitrarily poor if the signal did not lie in that subspace.

While a few papers have considered the problem of detection from compressive samples [47, 62, 99], these typically consider a simple hypothesis test of whether the vector \( x \) is 0 (i.e. observed vector is purely
noise) or a known signal vector \( s \):

\[
\mathcal{H}_0 : x = 0 \quad \text{vs.} \quad \mathcal{H}_1 : x = s
\]  

(4.1)

In this paper, we consider the subspace detection setting, where the subspace is known but the exact signal vector is unknown. This setup leads to the composite hypothesis test:

\[
\mathcal{H}_0 : x \in S \quad \text{vs.} \quad \mathcal{H}_1 : x \notin S
\]

Equivalently, let \( x_\perp \) denote the component of \( x \) that does not lie in \( S \). Now the composite hypothesis test can be stated as:

\[
\mathcal{H}_0 : \|x_\perp\|_2 = 0 \quad \text{vs.} \quad \mathcal{H}_1 : \|x_\perp\|_2 > 0
\]  

(4.2)

In [62], the authors also consider the unknown signal case, but do not consider a subspace setting. The most closely related paper is [16] where the authors consider that a few of the coordinates of the vector \( x \) are missing at random. Note that the missing observation case is a special case of compressive measurements where the rows of the measurement matrix have a single non-zero entry which picks out the corresponding coordinate. Thus, our results generalize that of [16], and additionally characterize the optimal information-theoretic threshold of signal to noise ratio needed to perform the testing reliably in high dimensions. While writing this paper, we also became aware of a paper that recently appeared on arXiv [5] that characterizes similar optimal threshold for detection of a sparse vector using compressive measurements. In this case, the subspace is specified by the span of a subset of the canonical basis vectors and the optimal threshold is shown to scale as \( n/m \), however the probability of error is allowed to decrease arbitrarily slowly in the number of compressive measurements \( m \). We propose detectors for an arbitrary subspace \( S \), and require the probability of error to decay exponentially in the number of compressive measurements. As a result, the optimal threshold for energy off the subspace scales as \( n \) in our setting.

Our results show that it is possible to detect whether a high-dimensional vector \( x \in \mathbb{R}^n \) lies in a \( r \)-dimensional subspace \( S \), where \( r \ll n \) using only \( m = \omega(r) \) \(^1 \) compressive samples, provided the energy off the subspace scales as \( n \). In comparison, in the setting of Eq. (4.1), if the signal vector is known then the energy of the signal vector \( s \) can be a constant \( > 0 \) (with universal random measurements that do not depend on the subspace \( S \)) [47]. Even weaker signals can be detected if both the signal and subspace are known, and the measurement matrix is tailored to the subspace \( S \) [99, 119]. However, this requires knowing the subspace \( S \) at the time of making the compressive measurements. We do not consider that setting here, and instead focus on universal compressive measurements that can be used to test whether a high-dimensional vector lies in any set of known subspaces that do not have to be fixed at the time of measurement collection.

### 4.2 Measurement Model and Test Statistic

Let \( S \) be a known \( r \)-dimensional subspace of \( \mathbb{R}^n \), spanned by the orthonormal columns of a matrix \( U \in \mathbb{R}^{n \times r} \). We are interested in determining whether an unknown vector \( x \in \mathbb{R}^n \) lies in \( S \) or not based only on a small number of compressive measurements. Specifically, for some \( m \geq 1 \), let \( A \in \mathbb{R}^{m \times n} \) be a random matrix with i.i.d. \( \mathcal{N}(0,1) \) entries. Then we observe the \( m \)-dimensional vector \( y = Ax + w \), where \( w \sim \mathcal{N}(0,\sigma^2 I_{n \times n}) \) denotes noise with known variance \( \sigma^2 \) that is independent of \( A \). Notice that this noise model is analogous to the one studied in [27], and different from the more commonly studied case \( y' = Ax + w_m \) where \( w \sim \mathcal{N}(0,\sigma^2 I_{m \times m}) \). In particular, for fixed \( A \) we have \( y \sim \mathcal{N}(Ax,\sigma^2 AA^T) \)

\(^1a = \omega(b)\) means that \( a > b \cdot \text{constant for every constant} > 0.\)
and $y' \sim \mathcal{N}(Ax, \sigma^2 I_{m \times m})$. We focus on the former model as in most applications noise is inherently generated during the measurement process, while compressive linear measurements may be formed later on to optimize storage or data collection.

Define the projection operator $P_U = UU^T$. Then $x_\perp = (I - P_U)x$, where $x_\perp$ is the component of $x$ that does not lie in $S$, and $x \in S$ iff $\|x_\perp\|^2_2 = 0$. Analogously to [16], we define the test statistic $T = \| (I - P_{BU})(AA^T)^{-1/2}y \|^2_2$ based on the observed vector $y$ and study its properties, where $B = (AA^T)^{-1/2}A$. Here $P_{BU}$ is the projection operator onto the column space of $BU$, specifically $P_{BU} = BU((BU)^TBU)^{-1}(BU)^T$, if $((BU)^TBU)^{-1}$ exists.

### 4.3 Main results

For the sake of notational simplicity, throughout this section we directly work with the matrix $B$ and its marginal distribution. Writing $y = B(x + w)$, we have $T = \| (I - P_{BU})y \|^2_2$. Notice that since $A$ is i.i.d. normal, the distribution of the row span of $A$ (and hence $B$) will be uniform over $m$-dimensional subspaces of $\mathbb{R}^n$ [67]. Furthermore, due to the $(AA^T)^{-1/2}$ term, the rows of $B$ will be orthonormal (almost surely). First we show that, in the absence of noise, the test statistic $\| (I - P_{BU})Bx \|^2_2$ is close to $m\|x_\perp\|^2_2/n$ with high probability.

**Theorem 4.1.** Let $0 < r < m < n$, $0 < \alpha_0 < 1$ and $\beta_0, \beta_1, \beta_2 > 1$. With probability at least $1 - \exp[\frac{(1 - \alpha_0 + \log \alpha_0)m}{2}] - \exp[\frac{(1 - \beta_0 + \log \beta_0)m}{2}] - \exp[\frac{(1 - \beta_1 + \log \beta_1)m}{2}] - \exp[\frac{(1 - \beta_2 + \log \beta_2)r}{2}]$

\[
\left( \frac{m}{n} - \beta_1 \beta_2 \frac{r}{n} \right) \|x_\perp\|^2_2 \leq \| (I - P_{BU})Bx \|^2_2 \leq \beta_0 \frac{m}{n} \|x_\perp\|^2_2.
\]

This implies the following corollary.

**Corollary 4.2.** If $m \geq c_1 r \log m$, then with probability at least $1 - c_2 \exp[-c_3 m]$,

\[
d_1 \frac{m}{n} \|x_\perp\|^2_2 \leq \| (I - P_{BU})Bx \|^2_2 \leq d_2 \frac{m}{n} \|x_\perp\|^2_2,
\]

for some universal constants $c_1 > 1$, $c_2 > 0$, $c_3 \in (0, 1)$, $d_1 \in (0, 1)$, $d_2 > 1$.

Corollary 4.2 states that given just over $r$ noiseless compressive measurements, we can estimate $\|x_\perp\|^2_2$ accurately with high probability. In the presence of noise, $t$ is natural to consider the hypothesis test:

$$T = \| (I - P_{BU})y \|^2_2 \leq \eta$$  \hspace{1cm} (4.3)

The following result bounds the false alarm level and missed detection rate of this test (for appropriately chosen $\eta$) assuming a lower bound on $\|x_\perp\|^2_2$ under $H_1$.

**Theorem 4.3.** If the assumptions of Corollary 4.2 are satisfied, and if for any $x \in H_1$

\[
\|x_\perp\|^2_2 \geq \sigma^2 \frac{4e + 2}{d_1} \left( 1 - \frac{r}{m} \right) n,
\]

then

\[P(T \geq \eta | H_0) \leq \exp[-c_4 (m - r)]\]

and

\[P(T \leq \eta | H_1) \leq c_2 \exp[-c_3 m] + \exp[-c_5 (m - r)].\]

where $\eta = e \sigma^2 (m - r)$, $c_4 = (e - 2)/2$, $c_5 = (e + \log(2e + 1))/2$, and all other constants are as in Corollary 4.2.
Figure 4.1: Simulation results. Type II error averaged over 100 runs for different values of $n$. Across all trials, we set $\sigma^2 = 1$, $r = 1$, $m = 5$, $U = (1, 0, \ldots, 0)^T$, and the Type I error is $\alpha = 0.05$. The detection threshold was set to $\eta_m = F^{-1}(1 - \alpha; m - 1)$ ($F^{-1}(\cdot, k)$ is the inverse CDF of $\chi^2_k$).

It is important to determine whether the performance of the test statistic we proposed can be improved further. The following theorem provides an information-theoretic lower bound on the probability of error of any test. A corollary of this theorem implies that the proposed test statistic is optimal, that is, every test with probability of missed detection and false alarm decreasing exponentially in the number of compressive samples $m$ requires that the energy off the subspace scale as $n$.

**Theorem 4.4.** Let $P_0$ be the joint distribution of $B$ and $y$ under the null hypothesis. Let $P_1$ be the joint distribution of $B$ and $y$ under the alternate hypothesis where $y = B(x + w)$, for some fixed $x$ such that $x = x_\perp$ and $\|x\|_2 = M$ for some $M > 0$. If conditions of Corollary 4.2 are satisfied, then

$$
\inf_{\psi} \max_{i=0,1} P_i(\psi \neq i) \geq \frac{1}{8} \exp \left[ - \frac{M^2 m}{2\sigma^2 n} \right]
$$

where the infimum is over all hypothesis tests $\psi$.

**Corollary 4.5.** If there exists a hypothesis test $\psi$ based on $B$ and $y$ such that for all $n$ and $\sigma^2$

$$
\max_{i=0,1} P(\psi \neq i|H_i) \leq C_0 \exp[-C_1(m - r)]
$$

for some $C_0, C_1 > 0$, then there exists some $C > 0$ such that $\|x_\perp\|_2^2 \geq C\sigma^2(1 - r/m)n$ for any $x \in H_1$ and all $n$ and $\sigma^2$.

### 4.4 Simulation results

The bounds in Theorem 4.3 and Corollary 4.5 state that for fixed $r$, $\|x_\perp\|_2$ needs to scale as approximately $\sqrt{n}$ to ensure low error under $H_1$. We perform simulations to demonstrate the effect of this scaling as follows.

We measure detection error as a function of $\|x_\perp\|_2$ for some values of $n$, for fixed false alarm level $\alpha$ (note that the true distribution under the null hypothesis is known, so we can construct an exact level $\alpha$.
test). If the bounds in Theorem 4.3 are tight, we expect to see that for fixed $\|x_\perp\|_2$, larger $n$ leads to larger error; we observe this in Figure 4.1a. Moreover, we expect that if we rescale the x-axis to $\|x_\perp\|_2/\sqrt{n}$, the error becomes independent of $n$, as is the case in Figure 4.1b. Thus, our simulations verify that the proposed test statistic can reliably detect if a $n$-dim vector lies in the given subspace provided the energy off the subspace scales as $n$.

\section*{4.5 Conclusion}

This paper shows that it is possible to detect whether a high-dimensional vector lies in a subspace with very few compressive measurements. We precisely characterized the amount of energy outside the subspace needed for reliable detection and verified this with simulations. The test statistic we propose is optimal in the sense that no other test can detect vectors with smaller energy off the subspace while ensuring that the probability of error decays exponentially with the number of compressive samples. Since the measurement model we consider is not specialized to the problem at hand, the proposed approach is universal and can be used in to detect energy outside any given subspace. This work also has important ramifications for sequential basis learning, where the subspace of interest may not be known a priori, but needs to be learnt from a collection of high-dimensional vectors that are expected to lie in some lower-dimensional subspace. We plan to investigate this direction in future work.
Chapter 5

Density-sensitive Semisupervised Regression

Semisupervised methods are techniques for using labeled data \((X_1, Y_1), \ldots, (X_n, Y_n)\) together with unlabeled data \(X_{n+1}, \ldots, X_N\) to make predictions. These methods invoke some assumption that links the marginal distribution \(P_X\) of \(X\) to the regression function \(f(x)\). For example, it is common to assume that \(f\) is very smooth over high density regions of \(P_X\). Many of the methods are ad-hoc and have been shown to work in specific examples but are lacking a theoretical foundation. We provide a minimax framework for analyzing semisupervised methods. In particular, we study methods based on metrics that are sensitive to the distribution \(P_X\). Our model includes a parameter \(\alpha\) that controls the strength of the semisupervised assumption. We then use the data to adapt to \(\alpha\). (For technical details including proofs, see [12].)

5.1 Introduction

Suppose we have data \((X_1, Y_1), \ldots, (X_n, Y_n)\) from a distribution \(P\), where \(X_i \in \mathbb{R}^d\) and \(Y_i \in \mathbb{R}\). Further, we have a second set of data \(X_{n+1}, \ldots, X_N\) from the same distribution but without the \(Y\)'s. We refer to \(L = \{(X_i, Y_i) : i = 1, \ldots, n\}\) as the *labeled data* and \(U = \{X_i : i = n + 1, \ldots, N\}\) as the *unlabeled data*. There has been a major effort, mostly in the machine learning literature, to find ways to use the unlabeled data together with the labeled data to construct good predictors of \(Y\). These methods are known as semisupervised methods. It is generally assumed that the the \(m = N - n\) unobserved labels \(Y_{n+1}, \ldots, Y_N\) are missing completely at random and we shall assume this throughout.

To motivate semisupervised inference, consider the following example. We download a large number \(N\) of webpages \(X_i\). We select a small subset of size \(n\) and label these with some attribute \(Y_i\). The downloading process is cheap whereas the labeling process is expensive so typically \(N\) is huge while \(n\) is much smaller.

Figure 5.1 shows a toy example of how unlabeled data can help with prediction. In this case, \(Y\) is binary, \(X \in \mathbb{R}^2\) and we want to find the decision boundary \(\{x : P(Y = 1 | X = x) = 1/2\}\). The left plot shows a few labeled data points from which it would be challenging to find the boundary. The right plot shows labeled and unlabeled points. The unlabeled data show that there are two clusters. If we make the seemingly reasonable assumption that \(f(x) = P(Y = 1 | X = x)\) is very smooth over the two clusters, then identifying the decision boundary becomes much easier. In other words, if we assume some link between \(P_X\) and \(f\), then we can use the unlabeled data; see Figure 5.2.

The assumption that the regression function \(f(x) = \mathbb{E}(Y | X = x)\) is very smooth over the clusters is known as the *cluster assumption*. In the special case where the clusters are low dimensional submanifolds,
the assumption is called the *manifold assumption*. These assumptions link the regression function \( f \) to the distribution \( P_X \) of \( X \).

Many semisupervised methods are developed based on the above assumptions although this is not always made explicit. And even with such a link, it is not obvious that semisupervised methods will outperform supervised methods. Making precise how and when these assumptions actually improve inferences is surprisingly elusive and most papers do not address this issue; some exceptions are [107], [110], [77], [95], [19], [111], [18] and [97]. These authors have shown that the degree to which unlabeled data improves performance is very sensitive to the cluster and manifold assumptions. In this contribution, we introduce adaptive semisupervised inference. We define a parameter \( \alpha \) that controls the sensitivity of the distance metric to the density, and hence the strength of the semisupervised assumption. When \( \alpha = 0 \) there is no semisupervised assumption, that is, there is no link between \( f \) and \( P_X \). When \( \alpha = \infty \) there is a very strong semisupervised assumption. We use the data to estimate \( \alpha \) and hence we adapt to the appropriate assumption linking \( f \) and \( P_X \). In addition, we should add that we focus on regression while most previous literature only deals with binary outcomes (classification).

We make the following contributions:

1. We formalize the link between the regression function \( f \) and the marginal distribution \( P_X \) by defining a class of functions spaces based on a metric that depends on \( P_X \). This is called a *density sensitive metric*.
2. We show how to consistently estimate the density-sensitive metric.
3. We propose a semi-supervised kernel estimator based on the density-sensitive metric.
4. We provide some minimax bounds and show that under some conditions the semisupervised method has smaller predictive risk than any supervised method.
5. The function classes depend on a parameter \( \alpha \) that controls how strong the semisupervised assumption is. We show that it is possible to adapt to \( \alpha \).
6. We provide numerical simulations to support the theory.

We now give an informal statement of our main results. In Section 5.5 we define a nonparametric class of distributions \( \mathcal{P}_n \). Let \( 0 < \xi < d - 3 \) and assume that \( m \geq n^{2/(2+\xi)} \). Let \( \mathcal{S}_n \) denote the set of supervised estimators; these estimators use only the labeled data. Let \( \mathcal{SS}_N \) denote the set of semisupervised

![Figure 5.1](image.png)

Figure 5.1: The covariate \( X = (X_1, X_2) \) is two-dimensional. The response \( Y \) is binary and is shown as a square or a circle. Left: The labeled data. Right: Labeled and unlabeled data.
Figure 5.2: Supervised learning (left) uses only the labeled data $\mathcal{L}_n$. Semisupervised learning (right) uses the unlabeled data $\mathcal{U}_N$ to estimate the marginal distribution $P_X$ which helps estimate $f$ if there is some link between $P_X$ and $f$. This link is the semisupervised (SS) assumption.

estimators; these estimators use the labeled data and unlabeled data. Then:

1. (Theorem 5.2.) There is a semisupervised estimator $\hat{f}$ such that

$$
\sup_{P \in \mathcal{P}_n} R_P(\hat{f}) \leq \left(\frac{C}{n}\right)^{\frac{2}{1+\xi}}
$$

(5.1)

where $R_P(\hat{f})$ is the risk of the estimator $\hat{f}$ under distribution $P$.

2. (Theorem 5.3). For supervised estimators $S_n$ we have

$$
\inf_{\hat{f} \in S_n} \sup_{P \in \mathcal{P}_n} R_P(\hat{f}) \geq \left(\frac{C}{n}\right)^{\frac{2}{1+\xi}}.
$$

(5.2)

3. Combining these two results we conclude that

$$
\frac{\inf_{\hat{f} \in SS_n} \sup_{P \in \mathcal{P}_n} R_P(\hat{f})}{\inf_{\hat{f} \in S_n} \sup_{P \in \mathcal{P}_n} R_P(\hat{f})} \leq \left(\frac{C}{n}\right)^{\frac{2(d-3-\xi)}{1+(d-1)} \to 0}
$$

(5.3)

and hence, semisupervised estimation dominates supervised estimation.

**Remark:** We assume, as is standard in the literature on semisupervised learning, that the marginal $P_X$ is the same for the labeled and unlabeled data. Extensions to the case where the marginal distribution changes, are possible but are beyond the scope of the work.

**Related Work.** There are a number of papers that discuss conditions under which semisupervised methods can succeed or that discuss metrics that are useful for semisupervised methods. These include Castelli and Cover [28], Castelli and Cover [29], Ratsaby and Venkatesh [105], Bousquet et al. [23], Singh et al. [110], Lafferty and Wasserman [77], Sinha and Belkin [111], Ben-David et al. [19], Nadler et al. [95], Sajama and Orlitsky [108], Bijral et al. [22], [18], [97] and references therein. Papers on semisupervised inference in the statistics literature are rare; some exceptions include [46], [44] and [82]. To the best of our knowledge, there are no papers that explicitly study adaptive methods that allow the data to choose the strength of the semisupervised assumption.

There is a connection between our work on the semisupervised classification method in [107]. He divides the covariate space $\mathcal{X}$ into clusters $C_1, \ldots, C_k$ defined by the upper level sets $\{p_x > \lambda\}$ of the density $p_X$ of $P_X$. He assumes that the indicator function $I(x) = I(p(y|x) > 1/2)$ is constant over each cluster $C_j$. In our regression framework, we could similarly assume that

$$
f(x) = \sum_{j=1}^{k} f_{\theta_j}(x) I(x \in C_j) + g(x)I(x \in C_0)
$$

where $f_{\theta}(x)$ is a parametric regression function, $g$ is a smooth (but nonparametric function) and $C_0 = \mathcal{X} - \bigcup_{j=1}^{k} C_j$. This yields parametric, dimension-free rates over $\mathcal{X} - C_0$. However, this creates a rather
unnatural and harsh boundary at \( \{ x : p_X(x) = \lambda \} \). Also, this does not yield improved rates over \( C_0 \). Our approach may be seen as a smoother version of this idea.

**Outline.** This chapter is organized as follows. In Section 5.2 we give definitions and assumptions. In Section 5.3 we define density sensitive metrics and the function spaces defined by these metrics. In Section 5.4 we define a density sensitive semisupervised estimator and we bound its risk. In Section 5.5 we present some minimax results. We discuss adaptation in Section 5.6. We provide simulations in 5.7. Section 5.8 contains closing discussion. Many technical details and extensions are contained in the original publication ([12]).

### 5.2 Definitions

Recall that \( X_i \in \mathbb{R}^d \) and \( Y_i \in \mathbb{R} \). Let

\[
L_n = \{(X_1,Y_1), \ldots, (X_n,Y_n)\}
\]  

be an iid sample from \( P \). Let \( P_X \) denote the \( X \)-marginal of \( P \) and let

\[
U_N = \{X_{n+1}, \ldots, X_N\}
\]

be an iid sample from \( P_X \).

Let \( f(x) \equiv f_P(x) = \mathbb{E}(Y|X=x) \). An estimator of \( f \) that is a function of \( L_n \) is called a **supervised learner** and the set of such estimators is denoted by \( S_n \). An estimator that is a function of \( L_n \cup U_N \) is called a **semisupervised learner** and the set of such estimators is denoted by \( SS_N \). Define the risk of an estimator \( \hat{f} \) by

\[
R_P(\hat{f}) = \mathbb{E}_P \left[ \int (\hat{f}(x) - f_P(x))^2 dP(x) \right]
\]

where \( \mathbb{E}_P \) denotes the expectation over data drawn from the distribution \( P \). Of course, \( S_n \subset SS_N \) and hence,

\[
\inf_{\hat{g} \in SS_N} \sup_{P \in \mathcal{P}} R_P(\hat{g}) \leq \inf_{\hat{g} \in S_n} \sup_{P \in \mathcal{P}} R_P(\hat{g}).
\]

We will show that, under certain conditions, semisupervised methods outperform supervised methods in the sense that the left hand side of the above equation is substantially smaller than the right hand side. More precisely, for certain classes of distributions \( \mathcal{P}_n \), we show that

\[
\frac{\inf_{\hat{g} \in SS_N} \sup_{P \in \mathcal{P}_n} R_P(\hat{g})}{\inf_{\hat{g} \in S_n} \sup_{P \in \mathcal{P}_n} R_P(\hat{g})} \to 0
\]

as \( n \to \infty \). In this case we say that semisupervised learning is **effective**.

**Remark:** In order for the asymptotic analysis to reflect the behavior of finite samples, we need to let \( \mathcal{P}_n \) to change with \( n \) and we need \( N = N(n) \to \infty \) and \( n/N(n) \to 0 \) as \( n \to \infty \). As an analogy, one needs to let the number of covariates in a regression problem increase with the sample size, to develop relevant asymptotics for high dimensional regression. Moreover, \( \mathcal{P}_n \) must have distributions that get more concentrated as \( n \) increases. The reason is that, if \( n \) is very large and \( P_X \) is smooth, then there is no advantage to semisupervised inference. This is consistent with the finding in [19] who show that if \( P_X \) is smooth, then “... knowledge of that distribution cannot improve the labeled sample complexity by more than a constant factor.”

**Other Notation.** If \( A \) is a set and \( \delta \geq 0 \) we define

\[
A \oplus \delta = \bigcup_{x \in A} B(x, \delta)
\]
where \( B(x, \delta) \) denotes a ball of radius \( \delta \) centered at \( x \). Given a set \( A \subseteq \mathbb{R}^d \), define \( d_A(x_1, x_2) \) to be the length of the shortest path in \( A \) connecting \( x_1 \) and \( x_2 \).

We write \( a_n = O(b_n) \) if \( |a_n/b_n| \) is bounded for all large \( n \). Similarly, \( a_n = \Omega(b_n) \) if \( |a_n/b_n| \) is bounded away from 0 for all large \( n \). We write \( a_n \propto b_n \) if \( a_n = O(b_n) \) and \( a_n = \Omega(b_n) \). We also write \( a_n \leq b_n \) if there exists \( C > 0 \) such that \( a_n \leq Cb_n \) for all large \( n \). Define \( a_n \geq b_n \) similarly. We use symbols of the form \( c, c_1, c_2, \ldots, C, C_1, C_2, \ldots \) to denote generic positive constants whose value can change in different expressions.

### 5.3 Density-Sensitive Function Spaces

We define a smoothed version of \( P_X \) as follows. (This is needed since we allow the marginal distribution \( P_X \) to be singular.) Let \( K \) denote a symmetric kernel on \( \mathbb{R}^d \) with compact support, let \( \sigma > 0 \) and define

\[
p_{\sigma}(x) \equiv p_{X,\sigma}(x) = \int \frac{1}{\sigma^d} K\left(\frac{||x-u||}{\sigma}\right) dP_X(x).
\]

Thus, \( p_{X,\sigma} \) is the density of the convolution \( P_{X,\sigma} = P_X * \mathbb{K}_\sigma \) where \( \mathbb{K}_\sigma \) is the measure with density \( K_\sigma(\cdot) = \sigma^{-d} K(\cdot/\sigma) \). \( P_{X,\sigma} \) always has a density even if \( P_X \) does not. This is important because, in high dimensional problems, it is not uncommon to find that \( P_X \) can be highly concentrated near a low dimensional manifold. And these are precisely the cases where semisupervised methods are often useful ([19]). Indeed, this was one of the original motivations for semisupervised inference. We define \( P_{X,0} = P_X \). For notational simplicity, we shall sometimes drop the \( X \) and simply write \( p_{\sigma} \) instead of \( p_{X,\sigma} \).

#### 5.3.1 The Exponential Metric

Following previous work in the area, we will assume that the regression function is smooth in regions where \( P_X \) puts lots of mass. To make this precise, we define a **density sensitive metric** as follows. For any pair \( x_1 \) and \( x_2 \) let \( \Gamma(x_1, x_2) \) denote the set of all continuous finite curves from \( x_1 \) to \( x_2 \) with unit speed everywhere and let \( L(\gamma) \) be the length of curve \( \gamma \); hence \( \gamma(L(\gamma)) = x_2 \). For any \( \alpha \geq 0 \) define the **exponential metric**

\[
D(x_1, x_2) \equiv D_{P,\alpha,\sigma}(x_1, x_2) = \inf_{\gamma \in \Gamma(x_1, x_2)} \int_0^{L(\gamma)} \exp\left[-\alpha p_{X,\sigma}(\gamma(t))\right] dt.
\]

In the original publication [12], we also consider a second metric, the **reciprocal metric**. Large \( \alpha \) makes points connected by high density paths closer; see Figure 5.3. Note that \( \alpha = 0 \) corresponds to Euclidean distance. Similar definitions are used in Sajama and Orlitsky [108], Bijral et al. [22] and Bousquet et al. [23].

#### 5.3.2 The Regression Function

Recall that \( f(x) \equiv f_P(x) = E(Y|X = x) \) denotes the regression function. We assume that \( X \in [0,1]^d \equiv \mathcal{X} \) and that \( |Y| \leq M \) for some finite constant \( M \).\footnote{The results can be extended to unbounded \( Y \) with suitable conditions on the tails of the distribution of \( Y \).} We formalize the semisupervised smoothness assumption by defining the following scale of function spaces. Let \( \mathcal{F} \equiv \mathcal{F}(P, \alpha, \sigma, L) \) denote the set functions \( f : [0,1]^d \to \mathbb{R} \) such that, for all \( x_1, x_2 \in \mathcal{X} \),

\[
|f(x_1) - f(x_2)| \leq L \, D_{P,\alpha,\sigma}(x_1, x_2).
\]
Figure 5.3: With a density metric, the points $X$ and $Z$ are closer than the points $X$ and $Y$ because there is a high density path connecting $X$ and $Z$.

Let $P(\alpha, \sigma, L)$ denote all joint distributions for $(X, Y)$ such that $f_P \in F(P, \alpha, \sigma, L)$ and such that $P_X$ is supported on $X$.

### 5.3.3 Properties of the Function Spaces

Let $B_{P, \alpha, \sigma}(x, \epsilon) = \{ z : D_{P, \alpha, \sigma}(x, z) \leq \epsilon \}$ be a ball of size $\epsilon$. Let $S_P$ denote the support of $P$ and let $N_{P, \alpha, \sigma}(\epsilon)$ denote the covering number, the smallest number of balls of size $\epsilon$ required to cover $S_P$. The covering number measures the size of the function space and the variance of any regression estimator on the space $F(P, \alpha, \sigma, L)$ depends on this covering number. Here, we mention a few properties of $N_{P, \alpha, \sigma}(\epsilon)$.

In the Euclidean case $\alpha = 0$, we have $N_{P, 0, \sigma}(\epsilon) \leq (C/\epsilon)^d$. But when $\alpha > 0$ and $P$ is concentrated on or near a set of dimension less than $d$, the $N_{P, \alpha, \sigma}(\epsilon)$ can be much smaller than $(C/\epsilon)^d$. The next result gives a few examples showing that concentrated distributions have small covering numbers. We say that a set $A$ is regular if there is a $C > 0$ such that, for all small $\epsilon > 0$,

$$\sup_{x, y \in A, \|x - y\| \leq \epsilon} \frac{d_A(x, y)}{\|x - y\|} \leq C. \quad (5.11)$$

where $d_A(x_1, x_2)$ is the length of the shortest path in $A$ connecting $x_1$ and $x_2$. Recall that $S_P$ denotes the support of $P$.

**Lemma 5.1.** Suppose that $S_P$ is regular.

1. For all $\alpha, \sigma$ and $P$, $N_{P, \alpha, \sigma}(\epsilon) \leq \epsilon^{-d}$.
2. Suppose that $P = \sum_{j=1}^k \delta_{x_j}$ where $\delta_x$ is a point mass at $x$. Then, for any $\alpha \geq 0$ and any $\epsilon > 0$, $N_{P, \alpha, \sigma}(\epsilon) \leq k$.
3. Suppose that $\dim(S_P) = r < d$. Then, $N_{P, \alpha, \sigma}(\epsilon) \leq \epsilon^{-r}$.
4. Suppose that $S_P = W \oplus \gamma$ where $\dim(W) = r < d$. Then, for $\epsilon \geq C\gamma$, $N_{P, \alpha, \sigma}(\epsilon) \leq \left(\frac{1}{\epsilon}\right)^r$.

### 5.4 Semisupervised Kernel Estimator

We consider the following semisupervised estimator which uses a kernel that is sensitive to the density. Let $Q$ be a kernel and let $Q_h(x) = h^{-d}Q(x/h)$. Let

$$\hat{f}_{h, \alpha, \sigma}(x) = \frac{\sum_{i=1}^n Y_i Q_h \left( \hat{D}_{\alpha, \sigma}(x, X_i) \right)}{\sum_{i=1}^n Q_h \left( \hat{D}_{\alpha, \sigma}(x, X_i) \right)} \quad (5.12)$$
where

$$
\hat{D}_{\alpha,\sigma}(x_1, x_2) = \inf_{\gamma \in \Gamma(x_1, x_2)} \int_0^{L(\gamma)} \exp[-\alpha \hat{p}_\sigma(\gamma(t))] \, dt,
$$

(5.13)

$$
\hat{p}_\sigma(x) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{\sigma^d} K\left(\frac{||x - X_{i+n}||}{\sigma}\right)
$$

(5.14)

and $m = N - n$ denotes the number of unlabeled points. We use a kernel estimator for the regression function because it is simple, commonly used and, as we shall see, has a fast rate of convergence in the semisupervised case.

The estimator $\hat{D}_{\alpha,\sigma}(x_1, x_2)$ is discussed in detail in [12] where we study its properties and we give an algorithm for computing it.

Now we give an upper bound on the risk of $\hat{f}_{h,\alpha,\sigma}$. In the following we take, for simplicity, $Q(x) = I(||x|| \leq 1)$.

**Corollary 5.2.** Suppose that $|Y| \leq M$. Define the event $G_m = \{||\hat{p}_\sigma - p_\sigma||_\infty \leq \epsilon_m\}$ (which depends on the unlabeled data) and suppose that $P(G_m^c) \leq 1/m$. Then, for every $P \in \mathcal{P}(\alpha, \sigma, L)$,

If $N(P,\alpha,\sigma,\delta) \leq C/\delta$ for $\delta \geq 1/2 e^{-\alpha \epsilon_m (2\epsilon_m)^{-\frac{1}{2}}}$ and $N \geq 2n$ then

$$
R_P(f_{h,\alpha,\sigma}) \leq e^{\alpha \epsilon_m (2\epsilon_m)} \left[ L^2 h^2 + \frac{1}{n} \left( \frac{C}{h} \right)^\xi \right] + 4M^2/m.
$$

(5.15)

Hence, if $m \geq n^{2/(2+\xi)}$, and $h \asymp (n e^{\alpha \epsilon_m (2-2\xi)^{-\frac{1}{2}}} - \frac{1}{2+\xi}$ then

$$
\sup_{P \in \mathcal{P}(\alpha,\sigma,L)} R_P(f_{h,\alpha,\sigma}) \preceq \left( \frac{C}{n} \right)^{\frac{2}{2+\xi}}.
$$

(5.16)

### 5.5 Minimax Bounds

To characterize when semisupervised methods outperform supervised methods, we show that there is a class of distributions $\mathcal{P}_n$ (which we allow to change with $n$) such that $R_{SS}$ is much smaller than $R_S$, where

$$
R_S = \inf_{f \in \mathcal{S}_n} \sup_{P \in \mathcal{P}_n} R_P(f) \quad \text{and} \quad R_{SS} = \inf_{f \in \mathcal{SS}_n} \sup_{P \in \mathcal{P}_n} R_P(f).
$$

To do so, it suffices to find a lower bound on $R_S$ and an upper bound on $R_{SS}$. In intuitively, $\mathcal{P}_n$ should be a set distributions whose $X$-marginals are highly concentrated on or near lower-dimensional sets, since this is where semisupervision methods deliver improved performance. Indeed, as we mentioned earlier, for very smooth distributions $P_X$ we do not expect semisupervised learners to offer much improvement.

#### 5.5.1 The Class $\mathcal{P}_n$

Here we define the class $\mathcal{P}_n$. Let $N = N(n)$ and $m = m(n) = N - n$ and define

$$
\epsilon_m \equiv \epsilon(m, \sigma) = \sqrt{\frac{C \log m}{m \sigma^d}}.
$$

(5.17)

Let $\xi \in [0, d-3]$, $\gamma > 0$ and define

$$
\mathcal{P}_n = \bigcup_{(\alpha, \sigma) \in \mathcal{A}_n \times \Sigma_n} Q(\alpha, \sigma, L)
$$

(5.18)
where $Q(\alpha, \sigma, L) \subset P(\alpha, \sigma, L)$ and $A_n \times \Sigma_n \subset [0, \infty]^2$ satisfy the following conditions:

(C1) $Q(\alpha, \sigma, L) = \left\{ P \in P(\alpha, \sigma, L) : N(P, \alpha, \sigma, \epsilon) \leq \left( \frac{C}{\epsilon} \right)^{\xi} \forall \epsilon \geq \left( \frac{1}{n} \right)^{\frac{1}{d+\gamma}} \right\}$

(C2) $\alpha \leq \frac{\log 2}{\epsilon(m, \sigma)}$.

(C3) $\left( \frac{1}{m} \right)^{\frac{1}{d(d+\gamma)}} \leq \sigma \leq \frac{1}{4C_0} \left( \frac{1}{n} \right)^{\frac{1}{d-1}}$

where $C_0$ is the diameter of the support of $K$.

Here are some remarks about $P_n$:

1. (C2) implies that $e^{\alpha \epsilon m} \leq 2$ and hence it can be shown that (C3) implies that $2^{(1/2)} D_{P,\alpha,\sigma}(x_1, x_2) \leq \widehat{D}_{\alpha,\sigma}(x_1, x_2) \leq 2 D_{P,\alpha,\sigma}(x_1, x_2)$ with probability at least $1 - 1/m$.

2. The constraint in (C1) on $N(\epsilon)$ holds whenever $P$ is concentrated on or near a set of dimension less than $d$ and $\alpha/\sigma^d$ is large. The constraint does not need to hold for arbitrarily small $\epsilon$.

3. Some papers on semisupervised learning simply assume that $N = \infty$ since in practice $N$ is usually very large compared to $n$. In that case, there is no upper bound on $\alpha$ and no lower bound on $\sigma$.

The class $P_n$ may seem complicated. This is because showing conditions where semisupervised learning provably outperforms supervised learning is subtle. Intuitively, the class $P_n$ is simply the set of high concentrated distributions with $\alpha/\sigma$ large.

### 5.5.2 Supervised Lower Bound

**Theorem 5.3.** Suppose that $m \geq n \frac{d(d+\gamma)}{\alpha-1}$. There exists $C > 0$ such that

$$R_S = \inf_{\hat{f} \in S_n} \sup_{P \in P_n} R_P(\hat{f}) \geq \left( \frac{C}{n} \right)^{\frac{2}{d+1}}. \tag{5.19}$$

### 5.5.3 Semisupervised Upper Bound

Now we state the upper bound for this class.

**Theorem 5.4.** Let $h = \left( n e^{2(2-\xi)} \right)^{-\frac{1}{2+\gamma}}$. Then

$$\sup_{P \in P_n} R(\hat{f}_{h,\alpha,\sigma}) \leq \left( \frac{C}{n} \right)^{\frac{2}{2+\gamma}}. \tag{5.20}$$

### 5.5.4 Comparison of Lower and Upper Bound

Combining the last two theorems we have:

**Corollary 5.5.** Under the conditions of the previous theorem, and assuming that $d > \xi + 3$,

$$\frac{R_{SS}}{R_S} \leq \left( \frac{1}{n} \right)^{\frac{2(d-3-\xi)}{(2+\xi)(d-1)}} \rightarrow 0 \tag{5.21}$$

as $n \rightarrow \infty$.

This establishes the effectiveness of semi-supervised inference in the minimax sense.

---

2See supplement for [12].

35
5.6 Adaptive Semisupervised Inference

We have established a bound on the risk of the density-sensitive semisupervised kernel estimator. The bound is achieved by using an estimate \( \hat{D}_{\alpha,\sigma} \) of the density-sensitive distance. However, this requires knowing the density-sensitive parameter \( \alpha \), along with other parameters. It is critical to choose \( \alpha \) (and \( h \)) appropriately, otherwise we might incur a large error if the semisupervised assumption does not hold or holds with a different density sensitivity value \( \alpha \). We consider two methods for choosing the parameters.

The following result shows that we can adapt to the correct degree of semisupervisedness if cross-validation is used to select the appropriate \( \alpha, \sigma \), otherwise we might incur a large error if the semisupervised assumption does not hold or holds with a different density sensitivity value \( \alpha \). We consider two methods for choosing the parameters.

For any \( f \), define the risk \( R(f) = \mathbb{E}[(f(X) - Y)^2] \) and the excess risk \( \mathcal{E}(f) = R(f) - R(f^*) = \mathbb{E}[(f(X) - f^*(X))^2] \) where \( f^* \) is the true regression function. Let \( \mathcal{H} \) be a finite set of bandwidths, let \( \mathcal{A} \) be a finite set of values for \( \alpha \) and let \( \Sigma \) be a finite set of values for \( \sigma \). Let \( \theta = (h, \alpha, \sigma) \), \( \Theta = \mathcal{H} \times \mathcal{A} \times \Sigma \) and \( J = |\Theta| \).

Divide the data into training data \( T \) and validation data \( V \). For notational simplicity, let both sets have size \( n \). Let \( \mathcal{F} = \{ \hat{f}_\theta^T \}_{\theta \in \Theta} \) denote the semisupervised kernel estimators trained on data \( T \) using \( \theta \in \Theta \).

For each \( \hat{f}_\theta^T \in \mathcal{F} \) let
\[
\hat{R}^V(\hat{f}_\theta^T) = \frac{1}{n} \sum_{i=1}^{n} (\hat{f}_\theta^T(X_i) - Y_i)^2
\]
where the sum is over \( V \). Let \( Y_i = f(X_i) + \epsilon_i \) with \( \epsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2) \). Also, we assume that \( |f(x)|, |\hat{f}_\theta^T(x)| \leq M \), where \( M > 0 \) is a constant.

**Theorem 5.6.** Let \( \mathcal{F} = \{ \hat{f}_\theta^T \}_{\theta \in \Theta} \) denote the semisupervised kernel estimators trained on data \( T \) using \( \theta \in \Theta \). Use validation data \( V \) to pick
\[
\hat{\theta} = \arg \min_{\theta \in \Theta} \hat{R}^V(\hat{f}_\theta^T)
\]
and define the corresponding estimator \( \hat{f} = \hat{f}_{\hat{\theta}} \). Then, for every \( 0 < \delta < 1 \),
\[
\mathbb{E}[\mathcal{E}(\hat{f}_\theta)] \leq \frac{1}{1 - a} \left[ \min_{\theta \in \Theta} \mathbb{E}[\mathcal{E}(\hat{f}_\theta)] + \frac{\log(J)}{nt} \right] + 4M^2
\]
where \( 0 < a < 1 \) and \( 0 < t < 15/(38(M^2 + \sigma^2)) \) are constants. \( \mathbb{E} \) denotes expectation over everything that is random.

In practice, both \( \Theta \) may be taken to be of size \( n^a \) for some \( a > 0 \). Then we can approximate the optimal \( h, \sigma \) and \( \alpha \) with sufficient accuracy to achieve the optimal rate. Setting \( \delta = 1/(4M^2n) \), we then see that the penalty for adaptation is \( \frac{\log(J)}{nt} + \delta M = O(\log n/n) \) and hence introduces only a logarithmic term.

**Remark:** Cross-validation is not the only way to adapt. For example, the adaptive method in [70] can also be used here.

5.7 Simulation Results

In this section we describe the results of a series of numerical experiments on a simulated data set to demonstrate the effect of using the exponential version of the density sensitive metric for small, labeled
semisupervised methods provably outperform supervised methods, under certain conditions. Semisupervised methods are very powerful but, like all methods, they only work under certain conditions.

5.8 Discussion

Semisupervised methods are very powerful but, like all methods, they only work under certain conditions. We have shown that, under certain conditions, semisupervised methods provably outperform supervised methods. In particular, the advantage of semisupervised methods is mainly when the distribution $P_X$ of $X$ is concentrated near a low dimensional set rather than when $P_X$ is smooth.

We introduced a family of estimators indexed by a parameter $\alpha$. This parameter controls the strength of the semi-supervised assumption. The behavior of the semi-supervised method depends critically on $\alpha$. Finally, we showed that cross-validation can be used to automatically adapt to $\alpha$ so that $\alpha$ does not need to be known. Hence, our method takes advantage of the unlabeled data when the semi-supervised assumption holds, but does not add extra bias when the assumption fails. Our simulations confirm that our proposed estimator has good risk when the semi-supervised smoothness holds.

The analysis in this work can be extended in several ways. First, it is possible to use other density sensitive metrics such as the diffusion distance [79]. Second, we defined a method to estimate the density sensitive metric that works under broader conditions than the two existing methods due to Sajama and
Orlitsky [108] and Bijral et al. [22]. We suspect that faster methods can be developed. Finally, other estimators besides kernel estimators can be used. We will report on these extensions elsewhere.
Chapter 6

Risk Bounds for Mode Clustering

Density mode clustering is a nonparametric clustering method. The clusters are the basins of attraction of the modes of a density estimator. We study the risk of mode-based clustering. We show that the clustering risk over the cluster cores — the regions where the density is high — is very small even in high dimensions. And under a low noise condition, the overall cluster risk is small even beyond the cores, in high dimensions. (For technical details including proofs, see [13].)

6.1 Introduction

Density mode clustering is a nonparametric method for using density estimation to find clusters [7, 30, 40, 42]. The basic idea is to estimate the modes of the density, and then assign points to the modes by finding the basins of attraction of the modes. See Figures 6.1 and 6.5.

In this paper we study the risk of density mode clustering. We define the risk in terms of how pairs of points are clustered under the true density versus the estimated density. We show that the cluster risk over the cluster cores — the high density portion of the basins — is exponentially small, independently of dimension. Moreover, if a certain low noise assumption holds then the cluster risk outside the cluster cores is small. The low noise assumption is similar in spirit to the Tsyabakov low noise condition that often appears in the high dimensional classification literature [9].

It is worth expanding on this last point. Because mode clustering requires density estimation — and because density estimation is difficult in high dimensions — one might get the impression that mode clustering will not work well in high-dimensions. But we show that this is not the case. Even in high dimensions the clustering risk can be very small. Again, the situation is analogous to classification: poor estimates of the regression function can still lead to accurate classifiers.

There are many different types of clustering — $k$-means, spectral, convex, hierarchical — and we are not claiming that mode clustering is necessarily superior to other clustering methods. Indeed, which method is best is very problem specific. Rather, our goal is simply to find bounds on the performance of mode base clustering. Our analysis covers both the low and high-dimensional cases.

Outline. In Section 2 we review mode clustering. In Section 3 we discuss the estimation of the clusters using kernel density estimators. Section 4 contains the main results. After some preliminaries, we bound the risk over the cluster cores in Section 4.3. In Section 4.4 we bound the risk outside the cores under a low noise assumption. In Section 4.5 we consider the case of Gaussian clusters. In Section 4.6 we show a different method to bound the risk in the low dimensional case. Section 5 contains some numerical experiments. We conclude with a discussion in Section 6.
Figure 6.1: Left: a simple dataset. Middle: the kernel density estimator. Right: The four estimated modes and their basins of attractions.

Related Work. Mode clustering is usually implemented using the mean-shift algorithm which is discussed in [40, 42, 53]. The algorithm is analyzed in [7]. [15, 81] introduced mode clustering to the statistics literature. The related idea of clustering based on high density regions was proposed in [61]. [31] and [30] propose several methods for selecting the bandwidth for estimating the derivatives of the density estimator which can in turn be used as a bandwidth selection rule for mode clustering. A method that is related to mode clustering is clustering based on trees constructed from density level sets. See, for example, [35], [71] and [69].

Notation: We let \( p \) denote a density function, \( g \) its gradient and \( H \) its Hessian. A point \( x \) is a local mode (i.e. a local maximum) of \( p \) if \( \|g(x)\| = 0 \) and all the eigenvalues of \( H(x) \) are negative. Here, \( \| \cdot \| \) denotes the usual \( L_2 \) norm. In general, the eigenvalues of a symmetric matrix \( A \) are denoted by \( \lambda_1 \geq \lambda_2 \geq \cdots \). We write \( a_n \leq b_n \) to mean that there is some \( C > 0 \) such that \( a_n \leq C b_n \) for all large \( n \). We use \( B(x, \epsilon) \) to denote a closed ball of radius \( \epsilon \) centered at \( x \). The boundary of a set \( A \) is denoted by \( \partial A \).

6.2 Mode Clustering and Morse Theory

Here we give a brief review of mode clustering, also called mean-shift clustering; more details can be found in [7, 30, 32, 40, 42].

6.2.1 Morse Theory

We will need some terminology from Morse theory. Good references on Morse theory include [17, 51, 89, 94].

Let \( p \) be a bounded continuous density on \( \mathbb{R}^d \) with gradient \( g \) and Hessian \( H \). A point \( x \) is a critical point if \( \|g(x)\| = 0 \). We then call \( p(x) \) a critical value. A point that is not a critical point is a regular point.

The function \( p \) is a Morse function if all its critical values are non-degenerate (i.e. the Hessian at each critical point is non-singular). A critical point \( x \) is a mode, or local maximum, if the Hessian \( H(x) \) is negative definite at \( x \). The index of a critical point \( x \) is the number of negative eigenvalues of \( H(x) \). Critical points are maxima, minima or saddlepoints.

The flow starting at \( x \) is the path \( \pi_x : \mathbb{R} \to \mathbb{R}^d \) satisfying \( \pi_x(0) = x \) and

\[
\pi'_x(t) = \nabla p(\pi_x(t)).
\] (6.1)
Figure 6.2: A Morse function with four modes. Each solid blue dot is a mode. Each red dot is a minimum. Pink dots denote saddle points. The green area is the descending manifold (cluster) for one of the modes.

The flow $\pi_x(t)$ defines the direction of steepest ascent at $x$. The destination and origin of the flow $\pi_x$ are defined by

$$\text{dest}(x) = \lim_{t \to \infty} \pi_x(t), \quad \text{org}(x) = \lim_{t \to -\infty} \pi_x(t).$$  \hspace{1cm} (6.2)

If $x$ is a critical point, then $\text{dest}(x) = x$.

The stable manifold corresponding to a critical point $y$— also called the ascending manifold or the basin of attraction— is

$$\mathcal{C}(y) = \{ x : \text{dest}(x) = y \}. \hspace{1cm} (6.3)$$

In particular, the basin of attraction of a mode $m$ is called a cluster. See Figures 6.2 and 6.3.

Let us mention a few properties of Morse functions that are useful:

1. Excluding critical points, two flow lines are either disjoint or they are the same.
2. The origin and destination of a flow line are critical points (except at boundaries of the domain).

   The set of points $x$ whose destinations are not modes are on the boundaries of clusters and form a set of measure 0.
3. Flow lines are monotonic: $p(x_t)$ is a non-decreasing function of $t$, where $x_t = \pi_x(t)$. Further, $p(\text{dest}(x)) \geq p(\text{org}(x))$ and $\text{dest}(x) \neq \text{org}(x)$ if $x$ is a regular point.
4. The index of $\text{dest}(x)$ is greater than the index of $\text{org}(x)$.
5. The flow has the semi-group property: $\phi(x, t + s) = \phi(\phi(x, t), s)$ where $\phi(x, t) = \pi_x(t)$.
6. Let $\mathcal{C}$ be the basin of attraction of a mode $m$. If $y$ is a critical point in the closure of $\mathcal{C}$ and $y \neq m$, then $y \in \partial \mathcal{C}$.
Figure 6.3: The three large black dots are the three local modes that induce three clusters based on the corresponding basins of attraction. The cluster boundaries, \( D \), consists of the local minima (the square box, \( D_0 \)) and the three thick smooth curves are \( D_1 \). The circles on the boundaries are saddle points. The dotted lines show the flow lines.

### 6.2.2 Clusters

Consider a distribution \( P \) on \( K \subset \mathbb{R}^d \) with density \( p \). We assume that \( p \) is a Morse function with finitely many critical points. The modes of \( p \) are denoted by

\[
\mathcal{M} = \{ m_1, \ldots, m_k \} \quad \text{(6.4)}
\]

The corresponding clusters are \( C_1, \ldots, C_k \) where \( C_j = \{ x : \text{dest}(x) = m_j \} \). Define the clustering function \( c : K \times K \rightarrow \{0, 1\} \) by

\[
c(x, y) = \begin{cases} 
1 & \text{if } \text{dest}(x) = \text{dest}(y) \\
0 & \text{if } \text{dest}(x) \neq \text{dest}(y).
\end{cases}
\]

Thus, \( c(x, y) = 1 \) if and only if \( x \) and \( y \) are in the same cluster.

Let \( X_1, \ldots, X_n \in \mathbb{R}^d \) be random vectors drawn iid from \( P \). Let \( \hat{p} \) be an estimate of the density \( p \) with corresponding estimated modes \( \hat{\mathcal{M}} = \{ \hat{m}_1, \ldots, \hat{m}_\ell \} \), and basins \( \hat{C} = \{ \hat{C}_1, \ldots, \hat{C}_\ell \} \). This defines a cluster function \( \hat{c} \).

In this paper, the **pairwise clustering loss** is defined to be

\[
L = \frac{1}{\binom{n}{2}} \sum_{j<k} I(\hat{c}(X_j, X_k) \neq c(X_j, X_k)) \quad \text{(6.5)}
\]

which is one minus the Rand index. The corresponding clustering risk is \( R = \mathbb{E}[L] \).

### 6.3 Estimated Clusters

Estimating the clusters involves two steps. First we estimate the density then we estimate the modes and their basins of attractions. To estimate the density we use the standard kernel density estimator

\[
\hat{p}_h(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h^d} K \left( \frac{||x - X_i||}{h} \right). \quad \text{(6.6)}
\]
MEAN SHIFT

1. Choose a set of grid points \( G = \{g_1, \ldots, g_N\} \). Usually, these are taken to be the data points.
2. For each \( g \in G \), iterate until convergence:
   \[
g^{(r+1)} \leftarrow \frac{\sum_i X_i K(||g^{(r)} - X_i||/h)}{\sum_i K(||g^{(r)} - X_i||/h)}.
   \]
3. Let \( \hat{\mathcal{M}} \) be the unique elements of \( \{g_1^{(\infty)}, \ldots, g_N^{(\infty)}\} \). Output \( \{g_1^{(\infty)}, \ldots, g_N^{(\infty)}\}, \hat{\mathcal{M}} \) and \( \hat{\text{dest}}(g_j) = g_j^{(\infty)} \).

Figure 6.4: The Mean Shift Algorithm

We will need the following result on the accuracy of derivative estimation. We state the result without proof as it is a simple generalization of the result in [55] which is based on Talagrand’s inequality. In fact, it is essentially a different way of stating the results of Lemmas 2 and 3 in [7].

**Lemma 6.1.** Let \( p_h(x) = \mathbb{E}[\hat{p}_h(x)] \). Assume that the kernel is Gaussian. Also assume that \( p \) has bounded continuous derivatives up to and including third order. Then:

1. **Bias** There exist \( c_0, c_1, c_2 \) such that
   \[
   \sup_x |p_h(x) - p(x)| \leq c_0 h^2, \quad \sup_x ||\nabla p_h(x) - \nabla p(x)|| \leq c_1 h^2, \quad \sup_x ||\nabla^2 p_h(x) - \nabla^2 p(x)|| \leq c_2 h.
   \]

2. **Variance** There exist \( b, b_0, b_1, b_2 \) such that, if \( (\log n/n)^{1/d} \leq h \leq b \) where \( b < 1 \), then,
   \[
   \mathbb{P}(\sup_x |\hat{p}_h(x) - p_h(x)| > \epsilon) \leq e^{-b_0 nh^d \epsilon^2} \]
   \[
   \mathbb{P}(\sup_x ||\hat{\nabla} p_h(x) - \nabla p_h(x)|| > \epsilon) \leq e^{-b_1 nh^{d+2} \epsilon^2} \]
   \[
   \mathbb{P}(\sup_x ||\hat{\nabla}^2 p_h(x) - \nabla^2 p_h(x)|| > \epsilon) \leq e^{-b_2 nh^{d+4} \epsilon^2}.
   \]

**Remark:** It is not necessary to use a Gaussian kernel. Any kernel that satisfies the conditions in [7] will do.

To find the modes of \( \hat{p}_h \) we use the well-known mean shift algorithm. See Figures 6.4 and 6.5. The algorithm approximates the flow defined by (6.1). The algorithm finds the modes, the basins of attractions and the destination \( \hat{\text{dest}}(x) \) of any point \( x \). A rigorous analysis of the algorithm can be found in [7].

### 6.4 Bounding the Risk

We are now ready to bound the clustering risk. We begin by introducing some preliminary concepts.

#### 6.4.1 Stability

To bound the clustering risk, we need to control how much the critical points can change when the density is perturbed. In particular, we need the following result which is Lemma 16 from [37].

**Lemma 6.2.** Let \( p \) be a density with compact support. Assume that \( p \) is a Morse function with finitely many critical values \( C = \{c_1, \ldots, c_L\} \) and that \( p \) has two continuous derivatives on the interior of its
support and non-vanishing gradient on the boundary of its support. Let $q$ be another density and let $\eta = \max\{\eta_0, \eta_1, \eta_2\}$ where

$$\eta_0 = \sup_x |p(x) - q(x)|, \quad \eta_1 = \sup_x ||\nabla p(x) - \nabla q(x)||, \quad \eta_2 = \sup_x ||\nabla^2 p(x) - \nabla^2 q(x)||$$

where $\nabla^2$ is the vec of the Hessian. There are constants $\kappa \equiv \kappa(p)$ and $A \equiv A(p)$ such that, if $\eta \leq \kappa$ then the following is true. The function $q$ is Morse and has $L$ critical points $C' = \{c'_1, \ldots, c'_L\}$. After a suitable relabeling of the indices, $c_j$ and $c'_j$ have the same Morse index for all $j$ and $\max_j ||c_j - c'_j|| \leq A(p)\eta$.

### 6.4.2 The Cluster Cores

An important part of our analysis involves, what we refer to as, the cluster cores. These are the high density regions inside each cluster. Consider the clusters $C = \{C_1, \ldots, C_k\}$. Define

$$\xi_j = \sup_{x \in \partial C_j} p(x)$$

(6.7)

where $\partial C_j$ is the boundary of $C_j$. For any $a \geq 0$ we define the $j^{th}$ cluster core by

$$C_j^\dagger(a) = \left\{x \in C_j : p(x) \geq \xi_j + a\right\}.$$  

(6.8)

See Figure 6.6.

In the sequel, let $p$ be a density function with compact support. Let $C_g \equiv \sup_x ||g(x)|| < \infty$ where $g$ is the gradient of $p$.

### 6.4.3 Bounding the Risk Over the Cores

Now we bound the risk for the data points that are in the cluster cores.

**Theorem 6.3.** Assume that $p$ is a Morse function with finitely many critical values. Denote the modes and clusters by $m_1, \ldots, m_k$ and $C_1, \ldots, C_k$. Let $\hat{p}_h$ be the kernel density estimator. Let $\eta = \max\{\eta_0, \eta_1, \eta_2\}$ where

$$\eta_0 = \sup_x |\hat{p}_h(x) - p(x)|, \quad \eta_1 = \sup_x ||\nabla \hat{p}_h(x) - \nabla p(x)||, \quad \eta_2 = \sup_x ||\nabla^2 \hat{p}_h(x) - \nabla^2 p(x)||.$$
Let \( a = C_g \eta + 2\eta_0 \) and let \( C^\dagger = \bigcup_j C_j^\dagger(a) \) and let \( \mathcal{X} = \{ X_i : X_i \in C^\dagger(a) \} \) be the points in the cores. Let \( \xi_j = \sup \{ p(x) : x \in \partial C_j \} \).

1. If
   \[
p(m_j) > 2A\eta C_g + 2\eta_0 + \xi_j
   \]
   for each \( j \), then \( \tilde{c}(X_i, X_j) = c(X_i, X_j) \) for every \( X_i, X_j \in \mathcal{X} \).
2. Let \( 0 < \delta < 1 \). If \( h_n \to 0 \) and \( h_n \geq n^{-\frac{1-\delta}{2+\delta}} \) then
   \[
   \mathbb{P} \left( \tilde{c}(X_i, X_j) \neq c(X_i, X_j) \text{ for any } X_i, X_j \in \mathcal{X} \right) \leq e^{-n^{\delta b}}
   \]
   for some \( b > 0 \) (independent of \( d \)).

**Remark:** Note that \( \eta, \eta_0, \eta_1, \eta_2 \) are functions of \( n \) but we suppress the dependence for simplicity.

### 6.4.4 Beyond the Cores

Now we bound the risk beyond the cores. Furthermore, we explicitly let \( d = d_n \) increase with \( n \). This means that the distribution also changes with \( n \) so we sometimes write \( p \) as \( p_n \).

Theorem 6.3 shows that the risk over the cores where \( p(x) > \xi + a \) is exponentially small as long as we take \( a = C\eta \) for some \( C > 0 \). The total risk is therefore the exponential bound plus the probability that a point fails to satisfy \( p(x) > \xi + a \). Formally:

**Corollary 6.4.** Assume the conditions of Theorem 6.3. The cluster risk is bounded by

\[
2P(p(X) < \xi + C\eta) + e^{-n^{\delta b}}.
\]

When \( h > 0 \) is fixed, we may take \( \delta = 1 \).

Note that, in the corollary, it is not necessary to let \( h \to 0 \). To further control the risk beyond the cores, we need to make sure that \( P(p(X) < \xi + C\eta) \) is small. To do this, especially in the high-dimensional case, we need to assume that the clusters are well-defined and are well-separated. We call these assumptions “low noise” assumptions since they are similar in spirit to the Tsybakov low noise assumption that is often used in high-dimensional classification [9]. Specifically, we assume that following:

(Low Noise Assumptions:)

1. Let \( \sigma_n \) be the minimal distance between critical points of \( p_n \). We assume that \( \sigma = \lim \inf_{n \to \infty} \sigma_n > 0 \).
2. Let \( m_n \) be the number of modes of \( p_n \). Then \( \lim \sup_{n \to \infty} m_n < \infty \).
3. \( \lim_{n \to \infty} \min_j p_n(m_j) > 0 \).  
4. \( \xi_n \leq n^{-\gamma} \) for some \( \gamma > 0 \) where \( \xi_n = \sup_{x \in D} p_n(x) \) and \( D = \bigcup_j \partial C_j \).
5. For all small \( \epsilon \), \( P(p_n(X) < \epsilon) \leq e^{\beta} \) where \( \beta = \beta_d \) is increasing with \( d \).

Parts 1-3 capture the idea that the clusters are well-defined. It is really parts 4 and 5 that capture the low noise idea. In particular, part 4 says that the density at the cluster boundaries is small. (See Figure 6.6.) Part 5 rules out thick tails. Note that for a multivariate Normal \( N(0, \sigma^2 I) \), we have that, for any fixed small \( \epsilon > 0 \), \( P(p(X) < \epsilon) \leq e^{-d} \) when \( \sigma \) is not too large. So part 5 automatically holds for distributions with Gaussian-like tails.

**Theorem 6.5.** Assume that \( p_n \) is Morse and that the low noise conditions hold. Assume that \( p_n \) has three bounded continuous derivatives. Let \( h_n \asymp n^{-1/(5+d)} \). Then the clustering risk \( R \) satisfies

\[
R \leq \left[ \left( \frac{\log n}{n} \right)^{3(5+d)} \frac{\epsilon^2}{\beta^2} \right] + e^{-n^{\delta b}}
\]

45
Figure 6.6: Left: When clusters are not well separated, $\xi$ is large. In this case, the mass inside the cluster but outside the core can be large. Right: When clusters are well separated, $\xi$ is small. The blue lines correspond to $p(x) = \xi + a$ for $a > 0$. The pink regions are the cluster cores.

with $\delta = 1/(5 + d)$. In particular, $R = O(\sqrt{\log n/n})$ when $\beta_d \geq \max\{(d + 5), 1/(2\gamma)\}$.

Remark: Parts 4 and 5 of the low noise assumption can be replaced by a single, slightly weaker assumption, namely, $P(|p_n(X) - \xi_{n,j}| \leq \epsilon) \leq e^\beta$ where $\xi_{n,j} = \sup_{x \in \partial C_j} p(x)$. The condition only need hold near the boundaries of the clusters.

6.4.5 Gaussian Clusters

Recently, [115] showed that a type of clustering known as convex clustering yields the correct clustering with high probability, even with increasing dimension, when the data are from a mixture of Gaussians. They assume that each Gaussian has covariance $\sigma^2 I$ and that the means are separated by a factor of order $\sqrt{d}$. Here we show a similar result for mode clustering. The clustering is based on a kernel estimator with a small but fixed bandwidth $h > 0$.

Let $X_1, \ldots, X_n \sim \sum_{j=1}^k \pi_j N(\mu_j, \sigma^2 I)$ so that $X_i$ has density

$$p(x) = \sum_{j=1}^k \frac{\pi_j}{\sigma^d(2\pi)^{d/2}} e^{-||x - \mu_j||^2/(2\sigma^2)}.$$ 

Theorem 6.6. Let $X_1, \ldots, X_n \sim \sum_{j=1}^k \pi_j N(\mu_j, \sigma^2 I)$. Let $\hat{p}_h$ be the kernel density estimator with fixed bandwidth $h > 0$ satisfying

$$0 < h < \frac{1}{2} \min_j \left( \frac{\pi_j^d}{\sqrt{2\pi}\sigma e^{16}} \right)^d.$$ 

Let $D = \bigcup_j \partial C_j$ and define $\Gamma = \min_j d(\mu_j, D)$. Suppose that $p$ is Morse,

$$\Gamma > \sigma \sqrt{32d + 2 \log \left( \frac{1}{\min_j \pi_j} \right)}.$$ 

46
and that
\[ \min_{j \neq k} |\mu_j - \mu_k| > 16\sigma \sqrt{d}. \quad (6.11) \]

Then, for all large \( n \),
\[
P\left( \hat{c}(X_j, X_k) \neq c(X_j, X_k) \text{ for some } j, k \right) \leq 2e^{-8d} + e^{-nb}.
\]

**Remark:** The theorem implies the following. As long as the means are separated from each other and from the cluster boundaries by at least \( \sqrt{d} \), then a kernel estimator has cluster risk \( 2e^{-8d} + e^{-nb} \). It is not necessary to make the bandwidth tend to 0.

### 6.4.6 Low Dimensional Analysis

In this section we assume that the dimension \( d \) is fixed. In this case, it is possible to use a different approach to bound the risk. We do not make the low noise assumption. The idea is to use results on the stability of dynamical systems (Chapter 17 of Hirsch, Smale and Devaney 2004). As before \( p \) is a Morse function and \( \tilde{p} \) is another function. Define \( \eta, \xi, C_g \) and \( C^\dagger(a) \) as in the previous sections.

Let \( C \) be a cluster with mode \( m \). Choose a number \( a \) such that
\[ 0 < a < p(m) - A\eta C_g - \xi. \quad (6.12) \]

For any \( x \) in the interior of \( C \), let
\[ t(x) = \inf \left\{ t : \pi_x(t) \in C^\dagger(a) \right\}. \quad (6.13) \]

If \( x \in \partial C \) then \( t(x) = \infty \) since \( \pi_x(t) \) converges to a saddlepoint on the boundary. But for any interior point, \( t(x) < \infty \). For \( x \in C^\dagger(a) \) we define \( t(x) = 0 \).

Our first goal is to control the difference \( ||\hat{\pi}_x(t(x)) - \pi_x(t(x))|| \). And to do this, we first need to bound \( t(x) \). Let
\[ \Delta(x) = \inf_{0 \leq t \leq t(x)} ||g(\pi_x(t))||. \quad (6.14) \]

Now, \( \Delta(x) > 0 \) for each \( x \notin \partial C \). However, as \( x \) gets closer to the boundary, \( \Delta(x) \) approaches 0. We need an assumption about how fast \( \Delta(x) \) approaches 0 as \( x \) approaches \( \partial C \) which is captured in the following assumption:

(B) Let \( B_\delta = \{ x \in C : d(x, \partial C) = \delta \} \). There exists \( \gamma > 0 \) such that, for all small \( \delta > 0 \),
\[ x \in B_\delta \text{ implies that } \Delta(x) \geq c\delta^\gamma. \quad (6.15) \]

We now have the following result.

**Theorem 6.7.** Assume condition B. Let \( \eta_1 = \sup_x ||\nabla p(x) - \nabla \tilde{p}(x)|| \). Let
\[ \delta = \left( \frac{\kappa_2 \sqrt{d} p(m)}{\log(\kappa_2 \sqrt{d}/\sqrt{\eta_1})} \right)^{\frac{1}{2}}. \quad (6.16) \]

Let \( x, y \in C \). Suppose that \( d(x, \partial C) \geq \delta \) and \( d(y, \partial C) \geq \delta \). Also, suppose that \( \eta_1 < a^2/C_g \). Then, for all small \( \eta \), \( \text{dest}(x) = \text{dest}(y) \).
Now let $\hat{p}_h$ be the kernel density estimator with $h = h_n \asymp n^{-1/(5+d)}$. In this case $\eta_1 = O_P(n^{-2/(6+d)})$ so, with $\delta$ defined as in (6.16), we have

$$\delta \equiv \delta_n \asymp (\log n)^{-1/(2\gamma)}.$$ 

By the previous theorem, there are no clustering errors for data points $X_i$ such that $d(X_i, D) \geq \delta_n$ where $D = \bigcup_j \partial C_j$ as long as $\eta_1 = \sup_x ||\nabla p(x) - \nabla \hat{p}(x)|| < a^2/C_0 \gamma$ which holds except on a set of exponentially small probability (Lemma 1). Hence, we have:

**Corollary 6.8.** Assume that $p$ is a Morse function with finitely many critical values. Denote the modes and clusters by $m_1, \ldots, m_k$ and $C_1, \ldots, C_k$. Suppose that condition (B) holds in each cluster. Let $\hat{p}_h$ be the kernel density estimator. Let

$$\eta = \max \{\eta_0, \eta_1, \eta_2\}$$

where

$$\eta_0 = \sup_x |\hat{p}_h(x) - p(x)|, \quad \eta_1 = \sup_x ||\nabla \hat{p}_h(x) - \nabla p(x)||, \quad \eta_2 = \sup_x ||\nabla^2 \hat{p}_h(x) - \nabla^2 p(x)||.$$ 

Let $D = \bigcup_j \partial C_j$ and let

$$\mathcal{X} = \{X_i : d(X_i, D) \geq \delta_n\}$$

where

$$\delta_n = \left(\frac{\kappa_2 \sqrt{dp(m)}}{\log(\kappa_2 \sqrt{d}/\eta_1)}\right)^{\frac{1}{2\gamma}} \asymp (\log n)^{-1/(2\gamma)}.$$ 

If $h_n \to 0$ and $nh_n^{d+4} \to \infty$, then

$$\mathbb{P}\left(\hat{c}(X_i, X_j) \neq c(X_i, X_j) \text{ for any } X_i, X_j \in \mathcal{X}\right) \leq e^{-nb} \tag{6.17}$$

for some $b > 0$.

Thus, the clustering risk is exponentially small if we exclude points that are close to the boundary.

### 6.5 Experiments

An example of highly non-spherical mode clusters in two dimensions is given in Figure 6.7, left panel. The true density (contours shown in blue) has two modes, with the corresponding basins of attraction shown in blue and green. Mean shift (using a Gaussian kernel with bandwidth 1) is applied to the 1000 points sampled from the density as plotted, and all but the points shown in red are correctly clustered. All but 1% of points are correctly clustered, despite a total variation distance of about 0.29 between the true and estimated densities.

Our theoretical results show that mean shift clustering should perform well even in high dimensions, assuming the bulk of the basins of attraction are well-separated by low density regions. We simulate such a setting in 10 dimensions, where we measure the performance of mean shift clustering on samples drawn from a mixture of two equal weight Gaussian components. The norm of the difference between the means is 5, and each component has randomly generated non-spherical covariance matrix with eigenvalues between 0.5 and 2. The center panel of Figure 6.7 shows the average clustering error as a function of the sample size $n$ and bandwidth $h$, after 75 replications of the procedure. With only 50 samples, an average error of 0.05 is achieved with the appropriate bandwidth.

The effect of component separation is demonstrated further in the right panel of Figure 6.7. Here, we draw $n = 300$ samples from an equal weight mixture of two unit covariance Gaussians in two dimensions, and measure the clustering error of mean shift (averaged over 35 replications).
6.6 Conclusion

Density mode clustering — also called mean-shift clustering — is very popular in certain fields such as computer vision. In statistics and machine learning it is much less well known. This is too bad because it is a simple, nonparametric and very general clustering method. And as we have seen, it is not necessary to estimate the density well to get a small clustering risk. Because of this, mode clustering can be effective even in high dimensions.

We have developed a bound on the pairwise risk of density mode clustering. The risk within the cluster cores — the high density regions — is very small with virtually no assumptions. If the clusters are well-separated (low noise condition) then the overall risk is small, even in high dimensions.

Several open questions remain such as: how to estimate the risk, how to choose a good bandwidth and what to do when the low noise condition fails. Regarding the last point, we believe it should be possible to identify regions where the low noise conditions fail. These are essentially parts of the cluster boundaries with non-trivial mass. In that case, there are two ways to reduce the risk. One is to merge poorly separated clusters. Another is to allow ambiguous points to be assigned to more than one cluster. For research in this direction, see [39, 81].
Chapter 7

Proposed and Exploratory Problems

In this chapter we give some background for the problems stated as proposed and exploratory work in subsections 1.2 and 1.3.

7.1 Proposed Work

7.1.1 Sparse Gaussian Mixture Learning with \( K > 2 \) Components

The main difficulty in extending the method described in Chapter 2 for learning sparse Gaussian mixtures to more than \( K = 2 \) mixture components is the nature of the Hardt and Price algorithm it is based on, which is highly specific to the two component case. (The subsequent step of finding a sparse Fisher discriminant can be handled with existing methods.) Below we explore the main challenge involved in extending the Hardt and Price method to multiple components. We also provide two possible approaches to overcoming this challenge. The first appears to require an \( O(K) \) increase in sample complexity, and a \( O(d^K) \) increase in computational complexity, which can be prohibitively expensive in a high dimensional setting for \( K \) even slightly larger than 2. The second can seemingly only give an approximation of the true mixture, even with infinite data, although the approximation in question may be useful under mild assumptions on the true mixture.

Recap of Hardt and Price Method

The dimension reduction portion of the Hardt and Price paper assumes the availability of an algorithm to accurately learn the parameters of an up to 4 dimensional mixture of up to 2 arbitrary Gaussians with arbitrary mixture weights.

The (original, \( K = 2 \)) algorithm relies on the following idea (ignoring finite sample issues). Fix some enumeration of the \( D = d + \frac{d(d+1)}{2} \) parameters of a \( d \)-dimensional normal, and let \( \theta_1, \theta_2 \in \mathbb{R}^D \) be the parameters of the two components. First, note that by applying the low-dimensional algorithm to up to 2-dimensional marginals, the unordered (and possibly identical) pairs \( \theta_1(i), \theta_2(i) \) for each \( i \in [D] \) can be learned (a 2-dimensional marginal is necessary exactly when parameter \( i \) corresponds to an off-diagonal element of the covariance). Use the low-dimensional algorithm to find \( i \in [D] \) such that \( \theta_1(i) \neq \theta_2(i) \), which must exist if the components are not identical and weights are non-zero. Fix some ordering of \( \theta_1(i), \theta_2(i) \), and record the component weights. The goal then is to order the pairs \( \theta_1(j), \theta_2(j) \) for each \( j \neq i \) in such a way that is consistent with the order chosen for \( \theta_1(i), \theta_2(i) \). This can be done by applying the low-dimensional algorithm to whichever marginal depends both on parameters \( i \) and \( j \). This marginal can be 2, 3, or 4 dimensional, determined as follows:
• if parameters $i$ and $j$ both correspond to elements of the mean vector or the diagonal of the covariance, then a bivariate marginal is sufficient;

• if $i$ and $j$ correspond to $\Sigma(\ell_1, \ell_2)$ and $\Sigma(\ell_3, \ell_4)$ where $\ell_k$ are all distinct, then a 4-dimensional marginal is needed;

• if $i$ and $j$ correspond to $\Sigma(\ell_1, \ell_2)$ and $\Sigma(\ell_2, \ell_3)$ where $\ell_k$ are all distinct, then a 3-dimensional marginal is needed.

(The actual algorithm treats the means and covariances somewhat separately.)

What Breaks When $K > 2$?

In the $K > 2$ case, if we found some parameter $i$ so that all the values $\theta_1(i), \ldots, \theta_K(i)$ are distinct, then a very similar approach would work – for each $j \neq i$, we could use the up to 4 dimensional marginal to find a permutation$^1$ of the (not necessarily distinct) parameters $\theta_1(j), \ldots, \theta_K(j)$ that is consistent with some fixed ordering of $\theta_1(i), \ldots, \theta_K(i)$.

However, in general, there may not exist any parameter $i$ that differentiates all $K$ components.

Next, we describe two possible approaches to overcoming this issue.

Approach 1: Multiple Pivots

As stated above, there may not exist any parameter $i$ that differentiates all $K$ components. However, there has to exist$^2$ some set of parameters $I \subseteq [D]$ which jointly differentiates all $K$ components, i.e. so that $\theta_k(i) \neq \theta_l(i)$ for all $k \neq l$ (where $\theta_k(i) \in \mathbb{R}^D$). We could then use the whole set $I$ simultaneously to find consistent permutations of the values for parameters $j \notin I$, as described in the previous section for the case $I = \{i\}$.

However, this approach seems to suffer from two drawbacks, since the smallest such set $I$ can be as large as $O(K)$. First, it seems inevitable that the “low”-dimensional algorithm would need to handle up to $O(K)$-dimensional problems. Second, it is not clear if there is a way to find such an $I$ other than brute force search over $\{I \subseteq [D] : \ |I| \leq \text{an upper bound on the size of the smallest } I\}$ (the upper bound in question is on the order of $K$).

Approach 2: Pairwise Constraint Reconciliation

Instead of relying on a single (set of) parameters which differentiate every component, we might try to use every/many pairs of parameters $i, j$ to find constraints on the possible permutations of the learned values for each parameter. I.e., where the Hardt and Price method can exactly infer the mixture parameters by only considering pairs of parameter positions $i, j$, we might seek some, not necessarily unique, set of mixture parameters $\theta_1, \ldots, \theta_K$ which are consistent with all observed constraints. Specifically, in essence this approach requires that we solve the following problem (ignoring finite sample issues).

Suppose each parameter $i \in [D]$ takes on unique values $\tau_1(i), \ldots, \tau_{K_i}(i)$ for some $1 \leq K_i \leq K$ in the mixture (i.e., there are $K_i$ different values among $\theta_1(i), \ldots, \theta_K(i)$). Let $W$ be a $K_1 \times \ldots \times K_D$ tensor so that, for any $i \in [K_1] \times \ldots \times [K_D]$, $W(i)$ is the weight of the component $k$ that has $\theta_k(i) = \tau_{\ell_k}(i)$ for each $i \in [D]$ (if such a component doesn’t exist, then $W(i) = 0$). In other words, instead of giving the mixture weights of parameters $\theta_1, \ldots, \theta_K$ directly, $W$ gives the mixture weights of parameters $\theta_1', \ldots, \theta_{K_1 \times \ldots \times K_D}'$ obtained by forming all possible combinations of parameter values $\tau_1(i), \ldots, \tau_{K_i}(i)$ at each position $i \in [D]$.  

$^1$In fact, we are searching not for permutations, but permutations with repeated elements.

$^2$Trivially, $I = [D]$ is valid.
By observing only marginals corresponding to pairs of parameter positions \( i, j \in [D] \), we observe the tensor \( W \) by summing out all but 2 of the dimensions of \( W \). Clearly, this is insufficient to fully recover \( W \) – we need to solve a system of \( \sum_{i,j \in [D]} K_i K_j \) equations for \( K_1 \times \ldots \times K_D \) unknowns.

However, it is reasonable to assume that \( K \) is small. If \( K_1 \times \ldots \times K_D \) is large, then we know that most of the weights in the true weight tensor \( W \) are zero. Hence, a reasonable approximation to the mixture might be possible to obtain by solving for the sparsest \( W \) which is consistent with all the linear constraints described above.

In practice, finite sample effects mean that the \( i, j \) marginals can only be resolved to some accuracy, hence the constraints on \( W \) should be enforced with some slack. The question then is whether the form of these constraints are such that solving this approximate sparse problem can give a good estimate of the true \( W \) (e.g., if the set of linear measurements which form the constraints satisfy some form of restricted isometry property).

### 7.1.2 Feature Selection for Mode Clustering

Incorporating feature selection in mode clustering is non-trivial, firstly since it is not immediately obvious what notion of feature irrelevance to use. We begin by proposing a marginal modality screening approach. Since mode clusters are defined by their modes, it may be reasonable to consider a feature to be relevant only if its marginal density is multimodal. We can apply any univariate multimodality test on the marginal projections of the data, select features which are marginally multimodal, and pass those on to a typical mode clustering algorithm like mean shift.

This notion of relevance is quite restrictive, as it excludes features which are marginally unimodal but which help increase the separation between modes separated along another direction (see, e.g., the example in Section 2.1). This might be rectified by an additional expansion stage, where features which are marginally dependent (as determined by an independence test) on any of the features selected in the initial marginal multimodality screening stage are also included. If we assume that the true density \( p(x) \) can be written as a product \( p(x) = p_m(x_m)p_u(x_u) \), where \( x = (x_m, x_u) \), \( p_m \) is multimodal, and \( p_u \) is unimodal, then it is easy to see that the above procedure would not select any of the features falling in \( x_u \). On the other hand, further conditions are needed to ensure that each feature in \( x_m \) is selected.

### 7.2 Exploratory Work

#### 7.2.1 Convex Sparse Clustering

We begin by briefly introducing the convex clustering approach, and then describe our sparse convex clustering method and the questions it raises.

**Background: Convex Clustering**

Recently, several authors have proposed very similar schemes for *convex clustering*, i.e., clustering by means of optimizing a convex objective \([63, 85, 100]\). This is in contrast to many clustering methods such as K-means, which require solving non-convex (and often NP-hard) optimization problems. These approaches are developed as follows. Given the data matrix \( X \in \mathbb{R}^{n \times d} \) (whose rows are comprised of the points we wish to cluster), we begin by considering a formulation of clustering in terms of the (non-
convex) optimization problem

\[
\min_{C \in \mathbb{R}^{n \times d}} \frac{1}{2} \|X - C\|^2_F \\
\text{subject to } \sum_{i,j=1}^n 1_{C_i \neq C_j} \leq M
\]

where \(C_i\) are the rows of \(C\). Unlike K-means, in this formulation clustering is not achieved by explicitly optimizing over a set number of centroids and cluster assignments. Instead, each data point \(X_i\) is represented by its own “cluster” center \(C_i\), but the constraint \(\sum_{i,j=1}^n 1_{C_i \neq C_j} \leq M\) limits the number of different values that \(C_1, \ldots, C_n\) can take. Subsequently, any points \(X_i\) and \(X_j\) such that \(C_i = C_j\) are considered to be in the same cluster. Here, the parameter \(M\) controls the number of clusters obtained – the higher \(M\) is, the more clusters are allowed (although there is no simple way of predicting how many clusters will be obtained using a given value of \(M\)).

The convex clustering objectives are then obtained by replacing the indicators with some convex relaxation. For instance, using the 2-norm, and expressing the problem in its Lagrangian form, we have

\[
\min_{C \in \mathbb{R}^{n \times d}} \frac{1}{2} \|X - C\|^2_F + \lambda \sum_{i,j=1}^n 1_{C_i \neq C_j} \|C_i - C_j\|_2
\]

where \(\lambda \geq 0\) sets the number of clusters. When \(\lambda = 0\), the optimal solution is to set \(C = X\), and so each data point is assigned its own cluster. As \(\lambda\) increases, the centroids \(C_i\) and \(C_j\) are “shrunk” towards each other, until eventually for some finite value of \(\lambda\) all centroids become identical, i.e. only a single cluster is left.

However, in practice this objective often gives uninteresting results. In particular, the term \(\sum \|C_i - C_j\|_2\) penalizes the differences between distant points too strongly. Ideally, we wish the number of clusters to slowly decrease as \(\lambda\) is increased, with nearby points being placed into clusters early on, then more distant points, etc. On the contrary, the above objective often tends to shrink all points simultaneously towards the single common center, so that either all points are in different clusters, or all points are in one cluster. We show an example of this in Figure 7.1 (b), for the two moons data in subfigure (a). To address this issue, the literature proposes to weigh the penalties \(\|C_i - C_j\|_2\) higher the closer the points \(X_i\) and
X_j are. Hence, we arrive at a typical convex clustering objective

$$\minimize_{C \in \mathbb{R}^{n \times d}} \frac{1}{2} \|X - C\|^2_F + \lambda \sum_{i,j=1}^{n} w_{ij} \|C_i - C_j\|^2$$

for given weights $w_{ij} \geq 0$, which in practice are usually set according to the $k$-nearest neighbor graph (for some $k$), i.e. $w_{ij} = 1$ if $i$ and $j$ are $k$-nearest neighbors, and $w_{ij} = 0$ otherwise. The effect of these weights for the two moons example is shown in Figure 7.1 (d).

This type of convex clustering objective was analyzed theoretically by [115], who provided finite sample bounds on the error of the resulting clustering and characterized conditions which guarantee non-trivial solutions (i.e., a number of clusters that is larger than one and smaller than the number of samples).

**Convex Sparse Clustering**

We arrive at our proposed convex sparse clustering objective by introducing a convex sparsifying penalty on the cluster centers. Namely, we modify the convex clustering objective described above by adding an L2 penalty (often referred to as the group LASSO penalty) on the columns of the cluster center matrix, which correspond to the dimensions of the problem:

$$\minimize_{C \in \mathbb{R}^{n \times d}} \frac{1}{2} \|X - C\|^2_F + \lambda_1 \sum_{i=1}^{d} \|C_{(i)}\|^2 + \lambda_2 \sum_{i,j=1}^{n} w_{ij} \|C_i - C_j\|^2$$

where $C_{(i)}$ are the columns of the matrix $C$. Here, $\lambda_1$ is the sparsity parameter – the higher it is, the more columns of $C$ will be exactly zero in the optimizer of this objective, and so the more feature-sparse the corresponding clustering.

Solving this objective is non-trivial, however we derived an alternating minimization algorithm (AMA) [116] approach to optimizing it, analogous to a similar approach for non-sparse convex clustering [41].

Preliminary results using this algorithm indicate that this objective does indeed effectively perform feature selection for clustering. However, its performance quickly degrades with the dimensionality of the input. It appears that the algorithm is very sensitive to the choice of penalization weights $w_{ij}$ – the method seems to succeed whenever the true clusters in the data are captured in the structure of the graph given by the weights $w_{ij}$, and fail otherwise. This is problematic, since in a high dimensional setting the k-nearest neighbor graph, or any other graph used to set the weights, may be too noisy for the clusters to be distinguishable. A direction of further inquiry is to explore other designs of the sparse convex clustering objective (and penalization weight setting) which result in better performance in high dimensions, and to establish statistical limits of the dimensionality of clustering problems which can be consistently resolved with any such approach to sparse convex clustering.

### 7.2.2 Detecting Arbitrary Non-Gaussian Directions

There exist several methods, such as projection pursuit [65, 66, 87, 101, 102] and Non-Gaussian Component Analysis [48], which, given a multivariate dataset, aim to find directions in it along which the data distribution is clearly non-Gaussian. The overarching motivation for this type of exploratory data analysis is that, in some sense, the Gaussian distribution can be thought of as the “least interesting”, and any deviation from Gaussian indicates that some structure is present [64].
Despite the wide range of approaches to this problem in the literature [48, 66, 102], to our knowledge none of the existing methods are guaranteed to consistently discover arbitrary deviations from normality (at least within reasonable limits, such as smoothness conditions on the non-Gaussian distribution).

Recent results on the maximum mean discrepancy (MMD) statistic have shown that it can be used to perform nonparametric two sample testing that is consistent against all alternatives [57]. The MMD is analogous to the covariance of a pair of random variables $X$ and $Y$ in a Reproducing Kernel Hilbert Space (RKHS), and is related to the Distance Covariance [114] which has many similar properties. We propose to develop a method for searching for non-Gaussian directions in multivariate data based on either of these statistics, and hence to establish an estimator which is consistent against all non-Gaussian alternatives. We also note that by using a Gaussian copula transform approach similar to [86], such a method can have applications in searching for interesting interactions between features of multivariate datasets, where the variables need not be marginally Gaussian.

### 7.2.3 Approximation Algorithms for Sparse K-means

Recently, several methods have been proposed for sparse clustering which modify the K-means clustering objective to incorporate various feature selection strategies [34, 113, 117]. For example, [117] consider optimizing the objective

$$
\max_{w \in \mathbb{R}^d, C \in \mathbb{R}^{K \times d}, L \in [K]^n} \sum_{j=1}^{d} w_j \left( \sum_{i=1}^{n} X_{i(j)}^2 - \sum_{k=1}^{K} \sum_{i: L_i = k} (X_{i(j)} - C_{k(j)})^2 \right)
$$

subject to $\|w\|_2^2 \leq 1$, $\|w\|_1 \leq s$, $w_j \geq 0 \forall j \in [d]$

where $C$ and $L$ are the typical cluster centers and cluster assignments in K-means, $w$ is a vector of weights assigned to each feature, and $s$ is the parameter that controls the degree of sparsity of $w$. Here, feature selection is achieved when $w_j = 0$ for some feature $j$. (Note that fixing $w = 1$ reduces the above problem to the between-clusters sum of squares formulation for ordinary K-means clustering.)

Clearly, this objective is in general NP-hard to solve, since it contains K-means as a special case (which is itself NP-hard). The existing algorithms for approximately solving this objective, and the other sparse K-means objectives in the literature, rely on coordinate descent-type procedures starting at some initialization for either the cluster centers, the cluster assignments, or the feature weights (similar to Lloyd’s algorithm for K-means). While this is a reasonable approach, there are several questions to be considered:

1. What constitutes a good initialization for these algorithms? Is there an analogy to the K-means++ [8] initialization method which provides guarantees on the quality of the resulting approximation?
2. Are there other approximation algorithms, either for one of the existing sparse K-means formulations, or for some new formulation, which provide better guarantees?
3. In case such guarantees prove elusive, can it be proven under any assumptions on the data generating distribution that any approximation to sparse K-means is likely to be computationally intractable, in the framework of computational lower bounds on statistical algorithms [20, 52]?
Bibliography


[16] L. Balzano, B. Recht, and R. Nowak. High dimensional matched subspace detection when data are missing. In ISIT, 2010. 1, 4.1, 4.2


[22] Avleen Bijral, Nathan Ratliff, and Nathan Srebro. Semi-supervised learning with density based distances. In 27th Conference on Uncertainty in Artificial Intelligence, 2011. 5.1, 5.3.1, 5.8


[33] Yao-ban Chan and Peter Hall. Using evidence of mixed populations to select variables for clustering very high-dimensional data. Journal of the American Statistical Association, 105(490), 2010. 2.1

[34] Xiangyu Chang, Yu Wang, Rongjian Li, and Zongben Xu. Sparse K-Means with $\ell_{\infty}/\ell_0$ Penalty for High-Dimensional Data Clustering. ArXiv e-prints 1403.7890, March 2014. 2.1, 7.2.3


[53] Keinosuke Fukunaga and Larry D. Hostetler. The estimation of the gradient of a density function,


[56] Alon Gonen, Dan Rosenbaum, Yonina Eldar, and Shai Shalev-Shwartz. The sample complexity of subspace learning with partial information. *arXiv:1402.4844*, 2014. 1, 3.1, 3.2, 3.4


[60] Moritz Hardt and Eric Price. Sharp bounds for learning a mixture of two gaussians. *ArXiv e-prints 1404.4997*, April 2014. 2.1, 2.3, 2.3.1, 2.4, 2.2, 2.4.2


[70] Samory Kpotufe. k-nn regression adapts to local intrinsic dimension. *NIPS*, 2011. 5.6


[73] Akshay Krishnamurthy and Aarti Singh. Low-rank matrix and tensor completion via adaptive
sampling. *Advances in Neural Information Processing Systems*, 2013. 1, 3.1


[110] Aarti Singh, R. D. Nowak, and X. Zhu. Unlabeled data: Now it helps, now it


