The Dissertation Committee for Christopher Garrett Kennedy
certifies that this is the approved version of the following dissertation:

Fast High Dimensional Approximation via Random Embeddings

Committee:

Rachel Ward, Supervisor

François Baccelli

Andrew Blumberg

Eric Price

Fast High Dimensional Approximation via Random Embeddings

by

Christopher Garrett Kennedy

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2018

Acknowledgments

First and foremost, I thank my advisor Rachel Ward for her continuous support throughout my graduate career. I also thank my committee for agreeing to supervise this thesis, and for the time and feedback they dedicated to it.

I thank my parents Tom and Diana, for their unending guidance. Thank you to all the friends who have made my days better in every way.

Many UT staff have been invaluable and timely throughout this process, I would especially like to thank Elisa Armendariz, Dan Knopf, Sandra Catlett, and Tim Perutz.

Lisa, nothing in my life would be possible without you.

Fast High Dimensional Approximation via Random Embeddings

Publication No. ____________________

Christopher Garrett Kennedy, Ph.D.
The University of Texas at Austin, 2018

Supervisor: Rachel Ward

In the big data era, dimension reduction techniques have been a key tool in making high dimensional geometric problems tractable. This thesis focuses on two such problems - hashing and parameter estimation. We study locality sensitive hashing(LSH), which is a framework for randomized hashing that efficiently solves an approximate version of nearest neighbor search. We propose an efficient and provably optimal hash function for LSH that builds on a simple existing hash function called cross-polytope LSH. In the context of parameter estimation, we focus on regression, for which the well-known LASSO requires precise knowledge of the unknown noise variance. We provide an estimator for this noise variance when the signal is sparse that is consistent and faster than a single iteration of LASSO. Finally, we discuss notions of distance between probability distributions for the purposes of quantization and propose a distance metric called the Rényi divergence, that achieves both large and small scale bounds.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Often for problems involving large amounts of data in very high dimension, computing even simple properties of the data set, for example pairwise distances, can be prohibitively expensive to compute and store. To illustrate this, consider a collection of 1 billion vectors of length 10000 (for example, a database of images). Computing the distance between two vectors in MATLAB on my laptop takes roughly $3 * 10^{-5}$ seconds, which seems fast, but to even find the nearest neighbor (by computing all pairwise distances with a point) would take 5 and a half hours. There are two fundamental approaches to dealing with these kinds of bottlenecks:

- Develop a better algorithm to the problem or a relaxation of the problem that doesn't require brute force computation.

- Reduce the dimensionality of the data to improve storage and computation time.

In this work we explore applications of both these techniques using modern tools from random matrix theory. The main problems we are solving, detailed below, are nearest neighbor search and regression. In both cases we assume the dimension of the data is very high and, in the case of nearest neighbor search, there are a lot of data points.

Common techniques for reducing the dimensionality of data include principal components analysis (PCA) and all of its' variants, as well as non-linear methods such as self-organizing maps, autoencoders, kernel-PCA, etc. However, these methods suffer from inefficiency on high-dimensional data, either in the preprocessing or embedding step. In this work, we use subsampled, fast matrices such as the Fast Fourier Transform (FFT). Although this choice of matrix is classical and well-studied, the theory behind its' use as a tool in dimension reduction is much more recent. We also employ the following principle, from [37], which extends the use of these matrix ensembles:

Reducing the dimension of a finite point set and reducing the dimension of the set of all sparse vectors, while preserving pairwise distances, are "nearly" equivalent.

"Nearly" has a precise quantitative meaning which we will see later. This principle allows us to use fast dimension reducing matrices like subsampled FFTs in problems like regression, where the underlying structure of some estimator is sparse. In particular, we use the (fast) matrix ensembles used in compressed sensing because they preserve distances on sparse vectors, can also be used as dimension reducing matrices. This allows us to show our methods are both (i-) provably efficient and (ii-) provably work for their corresponding geometric problems.

1.1 Nearest Neighbor Search

Suppose we are given a set of points $P = \{x_1, ..., x_n\} \subset X$ in a metric space $(X, \mathcal{D})$. The fundamental task we are trying to solve is to find the nearest neighbor in our dataset.

Definition 1. ***(Exact Nearest Neighbor)*** *Given a query point $p \in P$, return the point $q \in P$ that minimizes*

$$q := \operatorname{argmin}_{p_0 \in R} \mathcal{D}(p_0, p).$$

The naive algorithm simply computes the pairwise distances $\mathcal{D}(p_0, p)$ for every $p_0 \in P$ and keeps a running index of the minimum. For euclidean distance in $X = \mathbb{R}^d$, this algorithm runs in time $\mathcal{O}(nd)$ and becomes prohibitively expensive when $n, d \gg 0$. The typical problem that occurs when trying to improve this bound, in other words to achieve sublinear query time, is that the storage requirements scale on the order of $n^{\mathcal{O}(d)}$. This exponential growth is suspected to be unavoidable except in problems with additional structure (i.e.if the data lies in a low-dimensional manifold), and is a manifestation of the "curse of dimensionality."

In order to circumvent this, one needs to relax the problem we are trying to solve. Specifically we replace the exact nearest problem with approximate nearest neighbor search. The meaning of "approximate" can vary, but for our purposes it means the following.

Definition 2. ***(Approximate Nearest Neighbor)*** *Given $p \in P$ and $c, R > 0$ and suppose that $\exists p_0 \in P$ s.t. $\mathcal{D}(p_0, p) < R$. Return (with high probability) $q \in P$ s.t. $\mathcal{D}(p, q) < cR$.*

It should be clear that this problem is a relaxation of nearest neighbor search in the case that $c > 1$, and in general the performance of any algorithm should degrade as $c \searrow 1$. In fact, for $c = 1$, one can formulate an algorithm for nearest neighbor search by varying $R > 0$. For now, we fix our metric space $X \subset \mathbb{R}^d$ with the Euclidean metric.

One way to approach approximate nearest neighbors is to instead do a preprocessing step that involves randomly hashing each point in P, then looking for collisions in the hash maps. This requires careful choice of hash map, because for this algorithm to work we need the hash function to be more likely to map close points to the same hash value - so called **Locality Sensitive Hashing** (LSH). The cost for improved query time is in precomputing the hash values of every point.

From the above discussion, it should be clear that there is a tradeoff between:

(i-) Query time

(ii-) Number of hash computations for each point

(iii-) Time to hash individual points.

Items (i-) and (ii-), as we will see later, can be simultaneously minimized using a parameter called "sensitivity," which quantifies how well our hash function detects whether points are close. In fact, hash functions have been constructed that achieve the (asymptotically) optimal lower bound in terms of sensitivity, as $d \to \infty$, but they are either hard to implement or have inefficient hash computations.

As suggested in the first section, our approach to develop a more efficient algorithm for LSH is to project our points via some FFT-type matrix $M : \mathbb{R}^d \to \mathbb{R}^m$ where $m \ll d$, then hash our points in dimension m in such a way that the sensitivity is preserved. The main advantage is that we replace a typically dense matrix from $\mathbb{R}^d \to \mathbb{R}^d$ with a fast, structured projection matrix. This allows us to do the hashing operation on m-dimensional points, which is significantly faster when $m = \mathcal{O}(\operatorname{polylog} d)$. Our algorithm is adapted from a previously studied scheme called cross-polytope LSH. It is easy to implement and has properties that a straightforward to analyze using results on high-dimensional Gaussians. This allows us to bootstrap our analysis on previous results in a very straightforward way.

1.2 Regression

We now turn to the problem of regression analysis. In it's simplest form, the problem is the following.

Definition 3. *Given $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$ known and some unknown signal $\beta \in \mathbb{R}^p$ such that $y = X\beta + \eta$ where $\eta \in \mathbb{R}^n$ is noise of unknown variance,* *return* $\widehat{\beta}$ *that minimizes*

$$\widehat{\beta} := \operatorname{argmin}_{\beta_0 \in \mathbb{R}^p} \|\beta - \beta_0\|_r,$$

for some r.

Obviously if the noise is unknown the above problem is in general intractable. We make the simplifying assumption that $\eta \sim \mathcal{N}(0, \sigma^2)$ is Gaussian with small variance. Of course when $n \leq p$, $r = 2$, and there is no noise, the problem can be solved exactly as $\widehat{\beta} = X^{\dagger} y$, where $X^{\dagger} := (X^T X)^{-1} X^T$ is the pseudoinverse.

We work in the case where $\mathbf{p} \gg \mathbf{n}$, where least squares is less effective. Note that least squares cannot distinguish between solutions modulo the null space of X, and can potentially lose a lot of information about the desired β. For the problem to be more tractable, we make the assumption that $\beta \in \mathbb{R}^p$ is s-sparse. We can think of s as having very small growth compared to the other variables in the problem. Typical methods for solving the regression problem when $p \gg n$ need an accurate estimation of the variance of the noise. Intuitively, this estimate should tell the algorithm how much it should try to fit the estimate β to the transformed signal y. The LASSO solves a convex optimization problem that balances fitting the signal and a regularization term (typically some norm of the estimated β) to prevent overfitting. In particular, we minimize

$$\widehat{\beta} = \operatorname{argmin}_{\beta} \|X\beta - y\|_2^2 + \lambda \|\beta\|_1, \tag{1.1}$$

where λ is tuned according to the variance of the noise. If the exact noise level is known, there are a large class of results called **oracle inequalities** that provide bounds on $\|\widehat{\beta} - \beta\|_r$ (where $\widehat{\beta}$ is the solution to the LASSO objective) for various values of $r \geq 1$.

Recall that β is s-sparse (with unknown level of sparsity), so ideally we would use the ℓ_0 norm in (1.1) to promote sparsity. However, even if the level of sparsity is known this penalization requires checking all $\mathcal{O}(n^s)$ possible supports of β, which is intractable for even moderately sized s. On the other hand, we can replace the ℓ_1 norm with $\|\beta\|_2^2$, which actually leads to a closed form solution. This is called ridge regression or Tikhonov regularization. However, as we will see later, the solution to this penalization also has undesirable properties. There is a variant of ridge regression that uses a linear combination of the ℓ_2 and ℓ_1 penalty terms, called the elastic net [73].

Our approach deviates from the LASSO penalization, and seeks to exploit the sparse structure of β using short, fast matrices typically used for dimension reduction. We first compute $X^T y$ (where X is the design matrix from above) and then take averages of the small entries of the resulting vector. The hope is that, provided the matrix X^T behaves "well" with respect to the sparse vector β, the small entries capture information about the variance (remember the above assumption that the noise variance is small compared to the signal). The bulk computational step is a simple matrix/vector multiplication. This is beneficial both because it is extremely simple to implement and also because this operation is highly optimized in most languages and easily parallelizable.

1.3 Robust Quantization

For the above problems, we are attempting to recover simple geometric information based on incoming data. In the case of LASSO, we want to recover the variance of the noise in the received vector, and in LSH we want to efficiently recover nearest neighbors from a given query point. Typically we want to evaluate the performance of the algorithm - in the case of LASSO, the variance and corresponding λ parameter, and for LSH the randomized hash maps. In this section we generalize this framework to evaluate the performance of a pre-trained algorithm on some incoming data set.

The algorithmic framework we use that is very natural in the geometric setting is quantization. Formally, quantization will partition the space into various regions and hash it according to this partition, mapping the continuous space to a discrete set of partition elements. We make the additional simplifying assumption that our partition scheme is a Voronoi partition, i.e. that each partition element is the set of points closest to a given point according to some metric. Thus, a quantization scheme corresponds to an indexed set of pairs $\mathcal{Q} = \{(P_i, w_i)\} \subset \mathbb{R}^d \times \mathbb{R}^d$, where each w_i is a point in $\mathbb{R}^d$ and $P_i = \{x \in \mathbb{R}^d : \mathcal{D}(x, w_i) \leq \mathcal{D}(x, w_j) \, \forall j\}$.

Fix our distance to be the Euclidean metric, and assume our quantization scheme is well adapted to some distribution $\mathbb{P}_1$. To quantify this, define the quantization error

$$E_{\mathbb{P}_1,\mathcal{Q}} := \int_{\mathbb{R}^d} \ell(x, \operatorname{argmin}_{w_i} \|x - w_i\|_2) d\mathbb{P}_1(x),$$

for some loss function ℓ. This is enough to guarantee a hash function based on a voronoi partition (as is common in LSH) is well-adapted to a distribution $\mathbb{P}_1$. In addition to this **global** geometric property, we would like to also capture **local** bounds on small probability events.

Suppose we receive samples $q_i \sim \mathbb{P}_2$ from a new distribution $\mathbb{P}_2$. There are two fundamental questions we investigate:

- What notion of closeness between $\mathbb{P}_1$ and $\mathbb{P}_2$ will make the quantization scheme $\mathcal{Q}$ have low error for both distributions (also, can we extend these to finite sample bounds).
- What notion of closeness will ensure that small probability events according to $\mathbb{P}_1$ will also have small probability in $\mathbb{P}_2$, i.e. scale invariant bounds.

Note that the second item above allows for analysis of nearest neighbors in the context of a quantization scheme.

There are various divergence and transportation based distance metrics between probability distributions. Among the most popular are the Wasserstein distance and the Kullback-Leibner(KL) divergence. We will see in section 5, both of these notions are insufficient for satisfying both conditions listed above. Instead, we use a generalized KL divergence known as the Rényi divergence,

$$R_\alpha(\mathbb{P}_1 \| \mathbb{P}_2) := \frac{1}{\alpha - 1} \ln \left(\mathbb{E}_{\mathbb{P}_2} \frac{d\mathbb{P}_1}{d\mathbb{P}_2}^{\alpha} \right).$$

It can be shown that this approaches the Rényi divergence in the limit $\alpha \downarrow 1$. This notion of distance turns out to be appropriate for all of the purposes outlined above, and classifies proximity between the distributions $\mathbb{P}_1$ and $\mathbb{P}_2$ in both global and local properties.

1.4 Main Results

1.4.1 Fast Cross-Polytope LSH

There are many choices of hash functions in LSH that have various advantages/disadvantages many of which we will mention later. One that we will focus on because it has optimal asymptotic sensitivity and a very simple formulation is **cross-polytope LSH**:

$$h(x) = \operatorname*{argmin}_{u=\{\pm e_i\}} \left\| \frac{\mathcal{G}x}{\|\mathcal{G}x\|_2} - u \right\|_2. \tag{1.2}$$

Here, $\mathcal{G} \in \mathbb{R}^{d \times d}$ is a Gaussian matrix with i.i.d. $\mathcal{N}(0,1)$ entries, so that x is rotated uniformly at random, then rounded to the nearest ± unit vector e_i. This function has the obvious disadvantage that it takes time $\mathcal{O}(d^2)$ to compute. We replace the matrix $\mathcal{G}$ with a fast alternative $\mathcal{G}' : \mathbb{R}^d \to \mathbb{R}^{d'}$ which supports matrix/vector multiplication in time $\mathcal{O}(d \ln d)$. Our new hash function becomes

$$h(x) = \operatorname*{argmin}_{u=\{\pm e_i\}} \left\| \frac{\mathcal{G}'x}{\|\mathcal{G}'x\|_2} - u \right\|_2. \tag{1.3}$$

The choice of d' can vary and can be tuned for desired sensitivity/runtime, but for now we choose $d = d'$. Our main result about this version of cross-polytope LSH is the following.

Theorem 4. *Consider the hash family $\mathcal{H}$ defined by (1.3). Then, $\mathcal{H}$ has the optimal rate of convergence in sensitivity as $d \to \infty$ and supports time $\mathcal{O}(d \ln d)$ hash computation.*

Using a construction from [33], we can replace our subsampled FFT with a matrix that has only $\mathcal{O}(\operatorname{polylog} d)$ random bits. With this and a similar construction, we can extend our result.

Theorem 5. *There is a hash family $\widehat{\mathcal{H}}$ that has the optimal rate of convergence in sensitivity as $d \to \infty$, supports time $\mathcal{O}(d \ln d)$ hash computation, and only requires $\mathcal{O}(\operatorname{polylog} d)$ bits of randomness.*

One big omission thus far is the precise meaning of sensitivity, and also the optimal rate of convergence. These definitions, as well as proofs of theorems 4 and 5 will be given in section 4. In section 4.6, we illustrate the collision probability of our scheme compared to regular cross-polytope LSH for various distances. We also highlight the collision probabilities by dimension, versus the optimal rate mentioned above.

1.4.2 Regression

Our estimator for noise variance has a very simple formula for the case where $p = n$ and $X \in \mathbb{R}^{p \times p}$ is orthogonal. Note that for the purposes of estimating the noise variance, X is the identity w.l.o.g., so that we receive a noisy signal $y = \beta + \eta$. We state the orthogonal estimator here because of it's simplicity:

```
1: Compute the window estimators $S_j = \frac{1}{L}\sum_{i\in\Omega_j} |y_i|^2, j \in \{1, 2, \dots, p/L\}$.
2: Let $\widehat{\sigma^2} = (1 + \frac{1}{\log(p)})\frac{2L}{p}\sum_{j=1}^{p/(2L)} S_{(j)}$, where $\{S_{(j)}\}_j$ is a non-decreasing arrangement of $\{S_j\}_j$.
```

The above estimator is extremely efficient to compute, in fact the most expensive step is computing the window estimators (assuming the number of Ω_j is relatively small, so that sorting is cheap). Additionally, it works well in typical regimes.

Theorem 6. *Suppose* $y = X\beta + \eta$ *where* $X \in \mathbb{R}^{p\times p}$ *is orthogonal,* $\eta_j \sim \mathcal{N}(0, \sigma^2)$ *are independent and* β *is s-sparse. For window size* $L = \mathcal{O}(\text{polylog}(p))$ *and sufficiently small s, the above variance estimator satisfies*

$$|\widehat{\sigma^2} - \sigma^2| \leq \frac{6\sigma^2}{\log p},$$

with probability $1 - \frac{2}{p}$.

We can extend our estimator to the case where $p > n$. In this case, the estimator is nearly identical, but first we preprocess the vector y via multiplication by X^T where X is the design matrix, and then apply the above method. As we will see in section 3, assuming X is sufficiently "nice" we can use this preprocessing step to exploit the sparsity of β. This is additionally a use of the principle mentioned earlier this section, that matrices typically used for dimension reduction also preserve distance of sparse vectors. For this estimator, we have the following result.

Theorem 7. *Suppose* $y = X\beta + \eta$ *where* $X \in \mathbb{R}^{n\times p}$ *is sufficiently well-behaved,* $\eta_j \sim \mathcal{N}(0, \sigma^2)$ *are independent and* β *is* s*-sparse. For window size* $L = \mathcal{O}(\operatorname{polylog}(p))$, $n \geq L$ *and sufficiently small* s*, then the generalized variance estimator satisfies*

$$|\widehat{\sigma}^2 - \sigma^2| \leq \frac{C_{\delta,\beta,p}}{\log(p)}(\sigma^2 + 1).$$

We hide the constant $C_{\delta,\beta,p}$ in the above theorem for simplicity, but it approaches 0 at a rate of $\mathcal{O}(1/\operatorname{polylog}(p))$ in typical regimes.

The full version of the above estimator and theorem will be given in section 3. The proofs rely on combining properties of the matrix X^T when applied to the sparse vector β, and standard concentration results. We also provide an empirical comparison to other standard variance estimators in section 3.4 as well as results on well known genomics data sets.

1.4.3 Quantization

Using the Rényi divergence mentioned above, we can formulate our first result, a scale invariant bound between two probability distribution $\mathbb{P}_1$ and $\mathbb{P}_2$. Note that for the Rényi divergence to exist, $\mathbb{P}_1$ and $\mathbb{P}_2$ must be mutually absolutely continuous, but this is a necessary property for such a bound. Our first result in the follow.

Proposition 8. *Suppose $\mathbb{P}_1$ and $\mathbb{P}_2$ are probability distributions that are mutually absolutely continuous. Then, for all events E,*

$$\begin{aligned}&\mathbb{P}_2(E)^{(\alpha-1)/\alpha}\exp[-(\alpha-1)R_\alpha(\mathbb{P}_2\|\mathbb{P}_1)] \leq \mathbb{P}_1(E)\\ &\leq \mathbb{P}_2(E)^{\alpha/(\alpha-1)}\exp[(\alpha-1)R_\alpha(\mathbb{P}_1\|\mathbb{P}_2)].\end{aligned}$$

This result also implies a multiplicative bound between the quantization error of $\mathbb{P}_2$, $E_{\mathbb{P}_2,\mathcal{Q}}$ and the quantization error of $\mathbb{P}_1$, $E_{\mathbb{P}_1,\mathcal{Q}}$ according to some scheme $\mathbb{P}_2$ (we drop the subscript $\mathcal{Q}$ in our notation and assume this scheme is fixed):

$$\exp[-(\alpha-1)R_\alpha(\mathbb{P}_2\|\mathbb{P}_1)]E_{\mathbb{P}_2} \leq E_{\mathbb{P}_1} \leq \exp[(\alpha-1)R_\alpha(\mathbb{P}_1\|\mathbb{P}_2)]E_{\mathbb{P}_2}.$$

Making a further assumption that the distributions are supported in a ball of radius $R > 0$, we can use concentration inequalities to get finite sample bounds on the quantization error of an incoming set of point $q_i \sim \mathbb{P}_2$.

Proposition 9. *Suppose that $\mathbb{P}_1$ and $\mathbb{P}_2$ are mutually absolutely continuous, such that*
$R_\alpha(\mathbb{P}_1\|\mathbb{P}_2), R_\alpha(\mathbb{P}_2\|\mathbb{P}_1) \leq \delta$, for some $\alpha > 1$, $\delta > 0$. Suppose also that we have some fixed quantization scheme $\mathcal{Q}$ with error $E_{\mathbb{P}_1} < \epsilon$ on distribution $\mathbb{P}_1$ and $\epsilon > 0$ small, and we receive $\{q_i\}_{i=1}^N \sim \mathbb{P}_2$. Then,

$$\mathbb{P}_2(|\widehat{E_{\mathbb{P}_2}} - E_{\mathbb{P}_1}| \geq t + C_{R,\delta,\epsilon,\alpha}) \leq 2\exp\left(\frac{-2Nt^2}{R^2}\right). \tag{1.4}$$

The constant $C_{R,\delta,\epsilon,\alpha} \to 0$ as $\delta \to 0$. Roughly, the proposition says that the sample quantization error of points from $\mathbb{P}_2$ approaches the true sample error as the number of points goes to infinity, and the Rényi divergence between the distributions goes to 0.

We note that these results can be leveraged to achieve bounds on more precise events, such as the probability that the k nearest neighbors land in the same Voronoi cell (i.e. partition element of the quantization scheme) for some k, but the parameters become unmanageable. These considerations are beyond the scope of this work.

Chapter 2

Background

2.1 Dimension Reduction Techniques

A key technique throughout this work and a common theme across all high-dimensional data analysis is to first reduce the dimension of the data while preserving some important property of the data set (pairwise distances, ordering, etc). Typically, given a set of points $P = \{x_1, ..., x_n\} \subset \mathbb{R}^d$, one finds a linear map $A : \mathbb{R}^d \to \mathbb{R}^m$ where $m = \mathcal{O}(\text{polylog}(n))$ that is sufficiently "nice" for practical purposes, for example

$$(1-\delta)\|x_i - x_i\|_2^2 \leq \|A(x_i - x_j)\|_2^2 \leq (1+\delta)\|x_i - x_j\|_2^2, \quad \forall i \neq j. \tag{2.1}$$

We call a map satisfying the above property a **Johnson-Lindenstrauss (JL) Transform**. This property is especially important when the problem we are solving involves some geometrical property of the data, since pairwise distances are approximately preserved. The most natural map we can choose is to take A to be a uniformly random projection onto an n-dimensional subspace of $\mathbb{R}^d$. A typical way to do this is to generate a matrix $\mathcal{G} \in \mathbb{R}^{n \times d}$ with i.i.d. $\mathcal{N}(0,1)$ entries and then compute $A = UV^T$, where $\mathcal{G} = U\Sigma V^T$ is the SVD of $\mathcal{G}$. In particular, we have the following deterministic result:

Theorem 10. *[32] For any $0 < \delta < 1/2$, any point set $P \subset \mathbb{R}^d$ of size n, and $m = \mathcal{O}(\log(n)\delta^{-2})$ there is a map $A : \mathbb{R}^d \to \mathbb{R}^m$ such that (2.1) holds.*

Moreover, there is a construction where the map f is linear and the choice $m = \mathcal{O}(\log(n)\delta^{-2})$ is optimal over all linear maps ([4], [38]). Much work has gone into finding fast JL-transforms that get as close as possible to this bound. Following a strong line of work ([1], [2], [37], among others), a fast JLT can be constructed as follows:

$$A = H_S D_b.$$

Here, $D_b : \mathbb{R}^d \to \mathbb{R}^d$ is diagonal with i.i.d. Rademacher entries on the diagonal, and $H_S \in \mathbb{R}^{m \times d}$ is a partial Hadamard matrix restricted to a uniformly random subset of $|S| = m$ rows. Many versions of this transform occur in the literature, sometimes replacing the Hadamard matrix with a different orthogonal matrix (fast fourier tranform, e.g.), replacing the row subset with a sparse Gaussian, and without or without the diagonal Rademacher matrix.

2.1.1 Restricted Isometry Property/Connections

We now develop a distinct but related notion, the Restricted Isometry Property. As the name suggests, this property ensures our matrix is a near isometry on a restricted subset of $\mathbb{R}^d$, notably all sparse vectors up to a certain order.

Definition 11. *A matrix $X \in \mathbb{R}^{m \times d}$ satisfies the* ***Restricted Isometry***

Property *of (integer) order $s_0 > 0$ and level $\delta > 0$ if*

$$(1-\delta)\|x\|_2 \leq \|Ax\|_2^2 \leq (1+\delta)\|x\|_2^2 \text{ for all } s_0\text{-sparse } x \in \mathbb{R}^d.$$

It should be clear that in general, we can't say that a RIP matrix will be a near isometry on any given point set (this would imply it is an isometry on the whole space). However, if instead our matrix comes from a distribution that satisfies RIP of some order/level with high probability, then it is a probabilistic JL-transform. Specifically, we say a matrix $A \in \mathbb{R}^{m\times d}$ sampled from some distribution is a **probabilistic JL-transform** if

$$\mathbb{P}\left[(1-\delta)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1+\delta)\|x\|_2^2)\right] \geq 1 - 2\exp(-c_0\delta^2 m), \qquad (2.2)$$

for fixed $x \in \mathbb{R}^n$ and some constants c_0,$\delta > 0$ (c_0 may have mild dependencies on the other parameters). Note that this statement can easily be translated to a concentration inequality over a set of points via union bounds. We now have the following deep result from [11].

Theorem 12. *(Theorem 5.2 from [11]) Fix $m, d, \delta > 0$. Suppose that $A \in \mathbb{R}^{m\times d}$ comes from a distribution satisfying 2.2 for some $c_0 > 0$. Then, there are absolute constants c_1, c_2 such that with probability $1 - 2\exp(-c_2\delta^2 m)$ the matrix A satisfies RIP of order $s_0 \leq c_1\delta^2 m/\ln(d/s_0)$ and level $\delta > 0$.*

Thus, the probabilistic JL condition satisfies RIP with high probability. There is also a partial converse:

Theorem 13. *(Theorem 1.3 from [37]) Fix $\eta, \delta > 0$ and some finite set $E \subset \mathbb{R}^d$ of cardinality $|E| = p$. Suppose that A satisfies RIP of order $s_0 \leq$*

$40\ln(4p/\eta)$ and level δ. Let $b \in \{-1,1\}^d$ be an i.i.d. Rademacher sequence. Then, with probability $\geq 1-\eta$,

$$(1-\delta/4)\|x\|_2^2 \leq \|AD_b x\|_2^2 \leq (1+\delta/4)\|x\|_2^2$$

for all $x \in E$.

This theorem should give justification to the construction in the previous section, since a Hadamard matrix restricted to a uniformly random row subset satisfies RIP with high probability, thus multiplying it by a diagonal matrix with i.i.d. signs on the diagonal makes it also act as a JL transform on some fixed subset with high probability. These two theorems give a precise quantitative version of the principle from the introduction, that reducing the dimension of a finite point set and the set of all sparse vectors are "nearly" equivalent.

2.2 Regression and the LASSO

Consider the following setting: suppose $\beta \in \mathbb{R}^p$ is s-sparse, and that we are given a noisy, transformed version of this signal,

$$y = X\beta + \eta.$$

Here $\eta \in \mathbb{R}^n$ has i.i.d. Gaussian entries $\eta_j \sim \mathcal{N}(0,\sigma^2)$ and $X \in \mathbb{R}^{n\times p}$ is a known design matrix. The regression problem is to find an estimator $\widehat{\beta}$ that minimizes the mean squared error (MSE),

$$\mathrm{MSE}(\widehat{\beta}) := \mathbb{E}_{\widehat{\beta}}\left[\|X\widehat{\beta} - y\|_2^2\right]. \tag{2.3}$$

A key observation is that the above error can be decomposed into bias and variance terms,

$$\mathrm{MSE}(\widehat{\beta}) = \mathrm{Var}(X\widehat{\beta}) + \mathrm{Bias}(X\widehat{\beta}) + \sigma^2. \tag{2.4}$$

Minimizing both terms simultaneously is a classical problem in statistics - we don't want to overfit by minimizing bias and get a high variance estimator, or underfit by minimizing variance resulting in higher bias.

Suppose, in the above setting, that $p \leq n$, so that our matrix $X \in \mathbb{R}^{n\times p}$ is potentially overdetermined. In this case we can directly minimize the bias,

$$\widehat{\beta} = \mathrm{argmin}_\beta \, \|X\beta - y\|_2^2 = X^\dagger y,$$

where $X^\dagger = (X^TX)^{-1}X$ is the pseudoinverse of X. The obvious drawback of the above method is that the variance of this estimator can be large, and also if $p > n$, it is not uniquely defined. Instead, it is common to add a regularization term to prevent overfitting:

$$\widehat{\beta} = \mathrm{argmin}_\beta \, \|X\beta - y\|_2^2 + \lambda f(\beta). \tag{2.5}$$

The parameter $\lambda > 0$ is tuned to the amount of counterbalance we want, and the function f can vary based on application. In general, this problem no longer has a closed form solution, and for some choices of f it can be highly non-convex. We focus on the case where $f(\beta) = \|\beta\|_1$ so that the above objective is convex, and the minimization is called LASSO (least absolute shrinkage and selection operator). In the vein of many ideas from Compressed Sensing, one can think of the choice of ℓ_1 norm as a proxy for the ℓ_0 norm to promote sparsity (however, choosing $f(\beta) = \|\beta\|_0$ makes the problem intractable). Intuitively, the LASSO is selecting the "important" entries of β that will improve generalization error.

Remark 14. *(Ridge Regression) If instead we take $f(\beta) = \|\beta\|_2^2$, the resulting minimization is called ridge regression or Tikhonov regularization. Instead of promoting sparsity, the euclidean norm imposes a large penalty for large entries, which shrinks the entries of β to prevent overfitting. However, unlike the LASSO this method doesn't tell us which entries are important (useful in the case where $p \gg n$), which makes the resulting estimator less interpretable.*

Chapter 3

Variance Estimation for the LASSO

This section is based on work that appears in the preprint [35]. Recall that LASSO attempts to recover a noisy, transformed signal $Y = X\beta + \eta$ where $\eta \in \mathbb{R}^n$ has i.i.d. Gaussian entries $\eta_j \sim \mathcal{N}(0, \sigma^2)$, $X \in \mathbb{R}^{n \times p}$ is known, and $\beta \in \mathbb{R}^p$ is s-sparse, using the following minimization,

$$\widehat{\beta} = \operatorname{argmin}_\beta \|X\beta - y\|_2^2 + \lambda\|\beta\|_1.$$

The standard analysis of the LASSO is conditioned on the event $\{\lambda : \lambda/4 \geq \|X^T\eta\|_\infty/n\}$ (see [14]). In particular, for the case that η is Gaussian with variance σ^2 and $X \in \mathbb{R}^{n \times n}$ is orthogonal, with high probability we have $\|X^T\|_\infty/n = \Theta(\sigma^2 \log(n)/n)$. Thus, with the choice $\lambda = 4\sigma^2 \log(n)/n$, the LASSO will provably produce a good estimate β.

However, in applications, the variance σ, and hence a proper choice of λ, is not known a priori. We consider the case where σ is not known in advance, and needs to be estimated from the signal y. It should be clear from the above observations that precision in estimating the parameter σ improves recovery of the true signal.

3.1 Standard Methods

A good review of variance estimators for LASSO is given in [53], where variance estimation using cross-validated LASSO is highlighted as particularly strong in many sparsity regimes. This method typically uses 5 or 10-fold cross-validation to train the hyperparameters in LASSO and analysis relies on the restricted eigenvalue condition on the design matrix. The above work was later complemented by a theoretical analysis of a slightly modified variant of cross-validated LASSO in [18] (see also [23] [29], e.g.). The method of moments (see [21]) is a reasonable alternative to cross-validated LASSO. It relies on the assumption that the design matrix is Gaussian and exploits statistical properties to formulate an estimator. It is consistent with a good rate of convergence [21], but the design matrix has to be Gaussian which is restrictive. We should also mention a variant of the LASSO - the square-root LASSO (see [13]) - whose penalty level doesn't depend on the variance of the noise. However, the resulting estimator is formulated as a conic programming problem which can be inefficient in practice and is beyond the scope of this work.

3.2 Greedy Variance Estimation – The Orthonormal Case

For the moment we focus on the case where $X \in \mathbb{R}^{p \times p}$ is an orthonormal matrix ($p = n$) and the problem reduces to recovering the noisy signal $y = \beta + \eta$ (by rotational invariance of the Gaussian). In this regime, the LASSO has the closed form solution

$$\widehat{\beta}_i = \operatorname{sign}(y_i)(|y_i| - \lambda)_+,$$

where $\widehat{\beta}_i = \widehat{\beta}_i(\lambda)$ implicitly depends on λ. A standard approach is to minimize the cross-validation error:

$$\min_{\lambda} \|y - \widehat{\beta}(\lambda)\|_2,$$

which has nice practical and theoretical properties (see [36] e.g.). Moreover, given the optimal λ one can infer a good estimate of the variance as $\|\widehat{\beta} - y\|_2/p$. However, this approach still requires one to compute the LASSO minimizer over a range of λ values, whereas one would like to perform a single computation to estimate the variance (and thus optimal λ). We formulate a method to estimate the variance which only needs a single pass over the input y.

Algorithm 1 Greedy Variance Estimator – Orthonormal Design Matrix

```
1: Compute the window estimators $S_j = \frac{1}{L}\sum_{i\in\Omega_j} |y_i|^2, j \in \{1, 2, \ldots, p/L\}$.
2: Let $\widehat{\sigma^2} = (1 + \frac{1}{\log(p)})\frac{2L}{p}\sum_{j=1}^{p/(2L)} S_{(j)}$, where $\{S_{(j)}\}_j$ is a non-decreasing arrangement of $\{S_j\}_j$.
```

The basic idea behind the above algorithm is that we want to capture a noise estimator that avoids the entries of y affected by signal (hence in the second step we take the average of the smaller 50% of the window estimates). We multiply the resulting estimator by $1 + \frac{1}{\log(p)}$ to correct the downward bias that results from averaging only over the smallest windows.

Remark 15. *(Total variation denoising) Suppose we receive image-type data and instead of taking the LASSO minimizer we want to instead want to regu-*

larize by the total variation seminorm:

$$\widehat{\beta} = arg\min_{\beta} \|\beta - y\|_2^2 + 2\lambda TV(\beta), \tag{3.1}$$

where $TV(\beta) := \sum_n \|\beta_n - \beta_{n-1}\|$. *The typical assumption in this model is that the discrete derivative of true signal is sparse, which is promoted by the above objective. In this case, we can apply our estimator to the discrete derivative (which as observed is essentially a sparse signal plus noise) to get a reasonable estimate of the variance of the noise in this setting. This approach originally appeared in [56] and statistical guarantees on the resulting estimator* $\widehat{\beta}$ *have been developed in [44], [47], [67], culminating most recently in [30]. These papers give a framework that allows one to generalize the estimator (3.1) to when the signal is 2-D image data. We note that our estimators can also be easily adapted to 2-D image data by replacing window estimates with box estimates.*

We have the following result which guarantees accuracy of the estimator $\widehat{\sigma^2}$.

Theorem 16. *Suppose* $y = X\beta + \eta$ *where* $X \in \mathbb{R}^{p\times p}$ *is orthonormal,* $\eta_j \sim \mathcal{N}(0, \sigma^2)$ *are independent, and* β *is* s*-sparse. Consider window size* $L \geq \log^3(p)$, *and suppose that* $s \leq \frac{p}{2L}$. *Then the Greedy Variance Estimator produced by Algorithm 1 satisfies*

$$|\widehat{\sigma^2} - \sigma^2| \leq \frac{6}{\log p}\sigma^2,$$

with probability $1 - \frac{2}{p}$.

3.3 Greedy Variance Estimation – RIP Design Matrix

We now turn to the more general case where the design matrix $X \in \mathbb{R}^{n\times p}$ is possibly underdetermined $n \leq p$, but satisfies the Restricted Isometry Property with the appropriate constants (indeed this is a more general case, as an orthonormal matrix satisfies the RIP with constant $\delta = 0$). We define the regularized design matrix as $Z := [Z_{\Omega_1}, ..., Z_{\Omega_{p/L}}]$ where each $Z_{\Omega_i} \in \mathbb{R}^{n\times L}$,

$$Z_{\Omega_i} := U_i I_{n\times L} V_i \quad \text{such that} \tag{3.2}$$

$$X_{\Omega_i} = U_i \Sigma_i V_i \text{ is the SVD of } X_{\Omega_i}.$$

Then, we run a conditioning step based on the (block orthonormal) matrix Z and then run the algorithm similar to the orthonormal case:

Algorithm 2 Greedy Variance Estimator

```
1: Compute $\tilde{y} = Z^T y$.
2: Compute the window estimators $S_j = \frac{1}{L}\sum_{i\in\Omega_j} |\tilde{y}_i|^2, j \in \{1, 2, \ldots, n/L\}$.
3: Let $\widehat{\sigma^2} = (1 + \frac{1}{\log(p)})\frac{2L}{p}\sum_{j=1}^{p/(2L)} S_{(j)}$, where $\{S_{(j)}\}_j$ is a non-increasing arrangement of the window estimators $\{S_j\}_j$.
```

In practice, we use the matrix X instead of Z, however using Z allows us to do a more streamlined theoretical analysis. To see why this should work intuitively, assume that we precondition just on X that satisfies RIP for a large enough sparsity level s_0. Note that $X^T y = X^T X\beta + X^T\eta$, so the obstruction to estimating the noise is the $X^T X$ term. Then, $\|X\beta\|_2 = \|X_{\Omega_\beta}\beta\|_2 \approx \|\beta\|_2$, and if we assume our window set Ω_j is disjoint from Ω_β, RIP implies the restricted matrices $X_{\Omega_j}^T$, X_{Ω_β} satisfy $\|X_{\Omega_j}^T X_{\Omega_\beta}\| \leq \delta$ for $\delta > 0$ small. Thus, for a "good" window estimator, we only see the noise $X^T\eta$.

Theorem 17. *Suppose $y = X\beta + \eta$, $X \in \mathbb{R}^{n\times p}$, $\eta_j \sim \mathcal{N}(0, \sigma^2)$ are i.i.d., and β is s-sparse. Assume that $L \geq \log^3(p)$, $n \geq L$, $s \leq \frac{n}{2L}$, and that X satisfies (RIP) with order $s_0 = 2\max\{L, s\}$ and level $\delta > 0$. Then, the variance estimator from the above algorithm satisfies*

$$\left|\widehat{\sigma}^2 - \sigma^2\right| \leq$$
$$(1 + \frac{1}{\log(p)})\left(2\delta\frac{\|\beta\|^2}{L} + \frac{6\sigma^2}{\log(p)} + \frac{1}{L}\max(4\sigma^2\log(p), 8\sqrt{\delta}\sigma\|\beta\|^2\sqrt{\log(p)})\right)$$

with probability $1 - \frac{4}{p}$.

Remark 18. *The constants in Theorem 17 are chosen for neatness of presentation and are in no way optimized.*

Remark 19. *Although the right hand side of Theorem 17 contains factors involving $\|\beta\|_2$ (as opposed to $\|\beta\|_1$ which one finds in typical LASSO results), we do not expect this to be a problem in practice. In particular, one can assume $c\sigma \leq |\beta_j| \leq C\sigma$ for all j and some absolute constants $C, c > 0$. If the $|\beta_j|$ are below this threshold, they are essentially noise and difficult to detect in general (this is called the beta-min assumption). On the other hand, one can naturally expect the entries of β to have a uniform upper bound even as the problem size goes to infinity. Since $\|\beta\|_2 \leq s\sqrt{C\sigma}$, we just need that $\delta < \frac{1}{s}$ which will hold for our sparsity regime and standard matrix models (i.i.d. normalized Gaussian entries, for example) with high probability.*

3.4 LASSO Experiments

Our experimental methodology is based off of the results in [53]. In particular, we generate a design matrix $X \in \mathbb{R}^{n\times p}$ with i.i.d. entries $X_{ij} \sim \mathcal{N}(0, n^{-1/2})$ so that X satisfies RIP with sufficiently small constants with high probability. The sparsity level $s = \lceil n^{\alpha} \rceil$, with $\alpha < 1$, and the non-zero entries of β (chosen uniformly at random) are distributed according to a Laplace(1) distribution. The resulting β is scaled to have the specified norm. The experiments are over the following grid of parameter values, where $n = 100$ in all experiments.

- $p = 100, 200, 500, 1000$,
- $\|\beta\|_2 = 0.1, 1, 2, 5, 10$,
- $\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$.

We use the following estimators in our analysis:

- oracle: the oracle estimator $\hat{\beta} = \|\eta\|_2/\sqrt{n}$.
- window: the standard window estimator with the transformation $\tilde{y} = X^T y$.
- window-svd: the theoretical window estimator with the transformation $\tilde{y} = Z^T y$ where Z is given by (3.2).
- cv-lasso: 10-fold cross-validated LASSO (computed using the R package glmnet [24]).

- moment: method of moments estimator (see [21]).

We include the cross-validated LASSO because it was shown to be the most robust to changes in sparsity/dimension by [53] and the method of moments estimator because it aims to be a fast replacement for cv-LASSO.

Remark 20. *The glmnet package mentioned above ([24]) uses a version of cyclic coordinate descent instead of vanilla gradient descent. Consequently, it doesn't share the type of theoretical result contained in this paper, that also holds for regular cv-LASSO. Nonetheless, it performs well in practice, and scales to a problem size appropriate for comparing to our estimators.*

The window size is chosen based on an inflection point in the values of the estimator for a specific set of parameters as the window size varies.
Figure 3.1 shows performance for our estimators with window size based on an inflection point, $p = 1000$. Signal-less ($\|\beta\| = 0$), low SNR ($\alpha = 0.1$, $\|\beta\| = 1$), medium SNR ($\alpha = 0.1$, $\|\beta\| = 5$), high SNR ($\alpha = 0.1$, $\|\beta\| = 10$) are shown respectively, top to bottom. As we can see in Figure 3.1, the window and window-svd estimators have reasonable performance compared to the cv-LASSO with slightly larger biases. In particular, we do quite well for $\alpha = 0.1$, $\beta = 1$, performing similarly to cv-Lasso, and with a much smaller variance than the method of moments.

Remark 21. *We only include results for* $\alpha = 0.1$ *because the algorithm performs similarly for* $\alpha \leq 0.5$. *Moreover our theory only covers up to roughly*

$\alpha = 0.5$ *for reasonable choices of window size. The performance for dense signal* $\alpha = 0.9$ *is covered in its own section below.*

3.4.1 Optimal Window Size

It is notable to see how well our method can perform when the window size is optimized. Here, we give some representative plots (Figure 3.2) to show what happens to performance when replacing the window size with the optimal window size using prior knowledge of the variance. In all experiments, n=100 and p=1000. For the low SNR regimes, we see a similar downward bias to the oblivious choice of window size, although with a smaller bias. Similarly, for high SNR, the upward bias is also smaller than when choosing an oblivious window size. Table 3.1 shows optimal window sizes as a function of α and $\|\beta\|_2$ for $p = 200$. The optimal window size was found by a grid search over all possible window sizes using knowledge of the true variance. We note that the optimal window size is generally decreasing as a function of both the signal to noise ratio and the sparsity. Moreover, choosing the maximal window size is optimal in modest regimes.

Figure 3.2 shows the various lasso estimators with optimal window size, $p = 1000$. Top to bottom: Signal-less ($\|\beta\| = 0$), low SNR ($\alpha = 0.1$, $\|\beta\| = 1$), high SNR ($\alpha = 0.1$, $\|\beta\| = 10$) respectively.

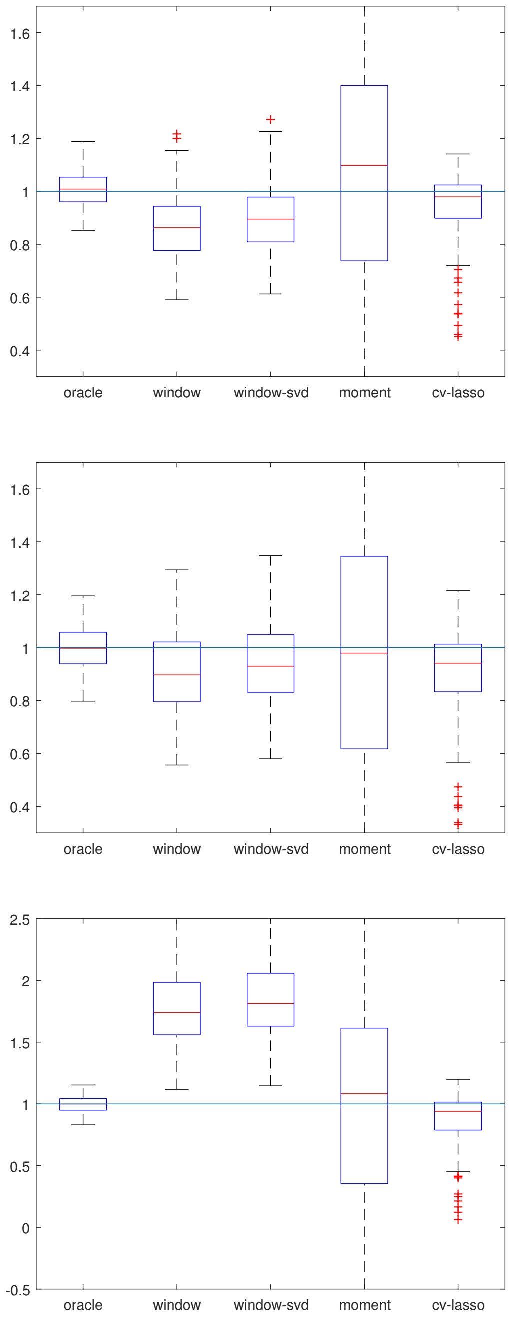

Figure 3.1: LASSO estimators with window size based on inflection point.

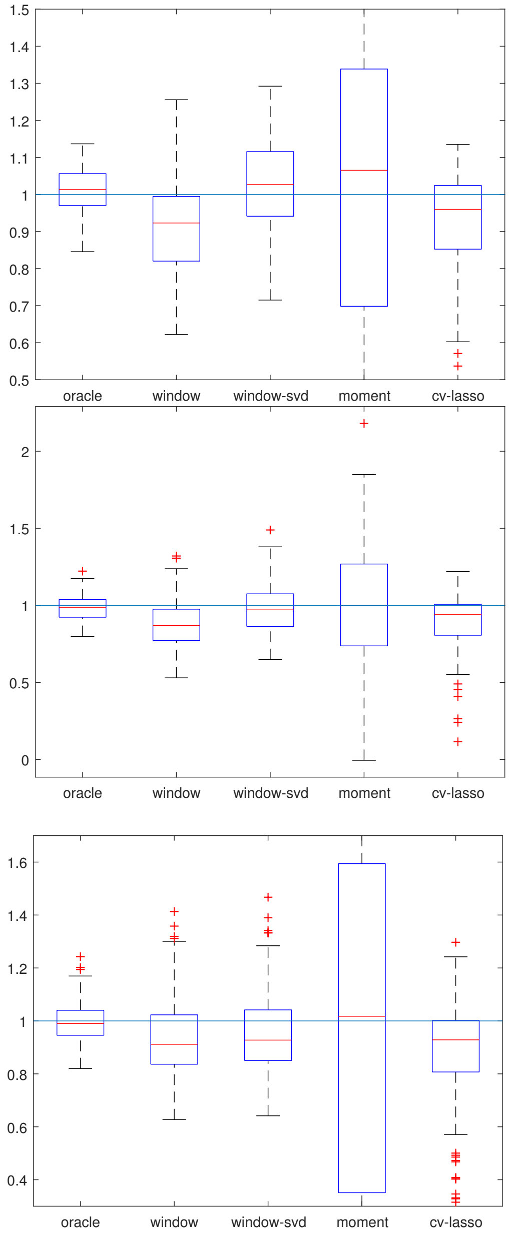

Figure 3.2: LASSO estimators with optimal window size.

		$\|\beta\|_2$				
		0.1	1	2	5	10
α	0.1	100	100	100	20	4
	0.3	100	100	100	22	4
	0.5	100	100	100	18	4
	0.7	100	100	100	18	4
	0.9	100	100	100	14	3

Table 3.1: Optimal window sizes as a function of α, $\|\beta\|_2$.

3.4.2 High Dimension

In this section we highlight the regime in which our estimator is most useful - when $p \gg n$ is large. In particular, we chose $n = 100$, $p = 100000$ in all experiments. In this regime, it is inefficient to even compute an optimal box size based on an inflection point in the value of the estimator, so instead the choice $L = 25$ was fixed for all experiments. The results are shown in Figure 3.3, $p = 100000$ $L = 25$. Top to bottom: Signal-less ($\|\beta\| = 0$), low SNR ($\alpha = 0.1$, $\|\beta\| = 1$), high SNR ($\alpha = 0.1$, $\|\beta\| = 10$) respectively. Although the bias remains, the estimator performs well, especially in low SNR regimes. This is likely due to the strength of the compressed sensing properties for the design matrix as the dimension grows. The bias increases with higher SNR, however our estimator maintains a lower variance than cv-LASSO.

3.4.3 Orthogonal Design Matrix

We find our estimator performs quite well in the case where the design matrix is orthogonal, as shown in Figure 3.4, $p = n = 200$. Top to bottom: Signal-less ($\|\beta\| = 0$), low SNR ($\alpha = 0.1$, $\|\beta\| = 1$), high SNR ($\alpha = 0.1$,

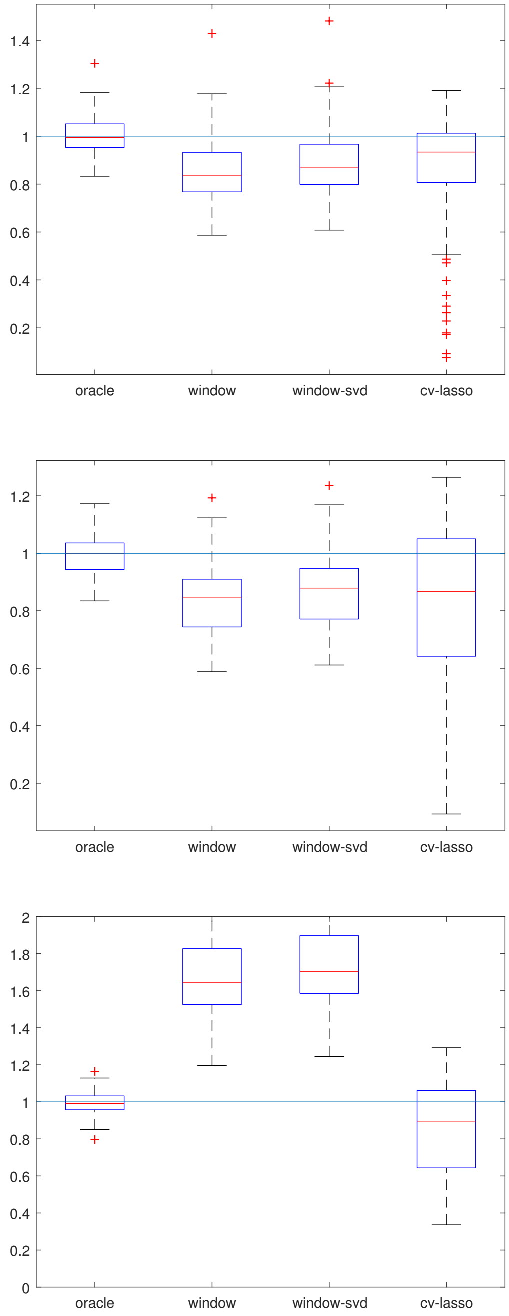

Figure 3.3: LASSO estimators in the high dimensional regime.

$\|\beta\| = 10$) respectively. In all experiments, the window size is chosen via inflection point in the value of the estimator. The method of moments still performs reasonably well, but suffers a strong upwards bias for large SNR. We note that in all regimes, our estimator performs better than cross-validated LASSO. Moreover, it is more robust to changes in SNR than when the design matrix is RIP (but not necessarily orthogonal).

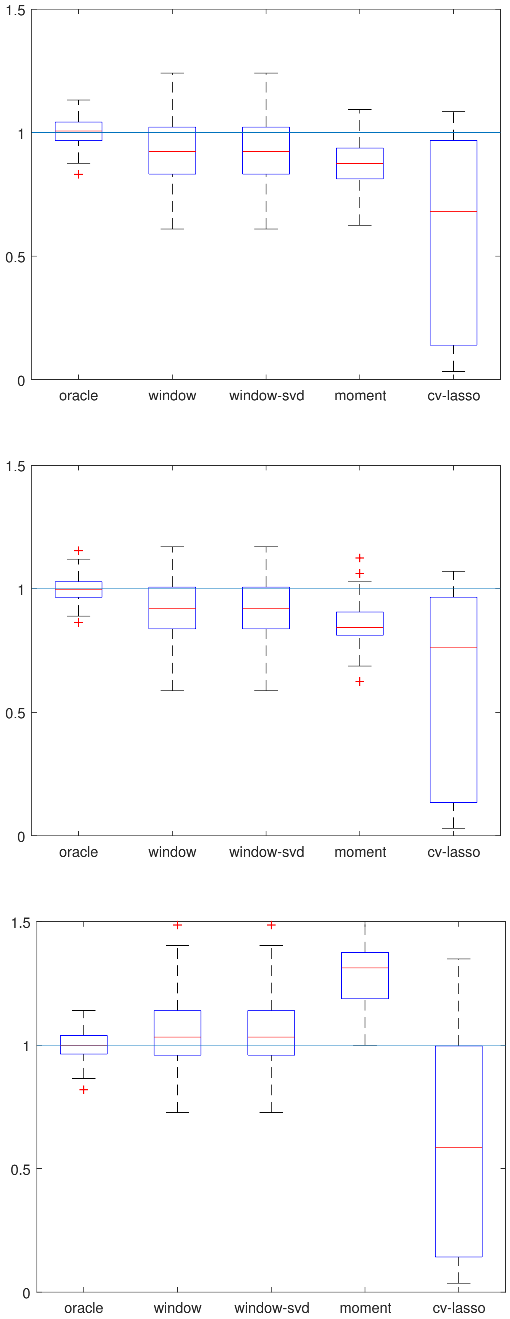

Figure 3.4: LASSO estimators with orthogonal design matrix.

3.4.4 Dense Signal

Our theory does not cover high sparsity levels ($\alpha \geq 0.9$), but nonetheless our estimator performs well. Although more prone to high levels of SNR, we are still competitive with cv-LASSO in low SNR regimes as seen in Figure 3.5. $p = 200$, top to bottom: Low SNR ($\alpha = 0.9$, $\|\beta\| = 1$), medium SNR ($\alpha = 0.9$, $\|\beta\| = 5$), high SNR ($\alpha = 0.9$, $\|\beta\| = 10$), respectively.

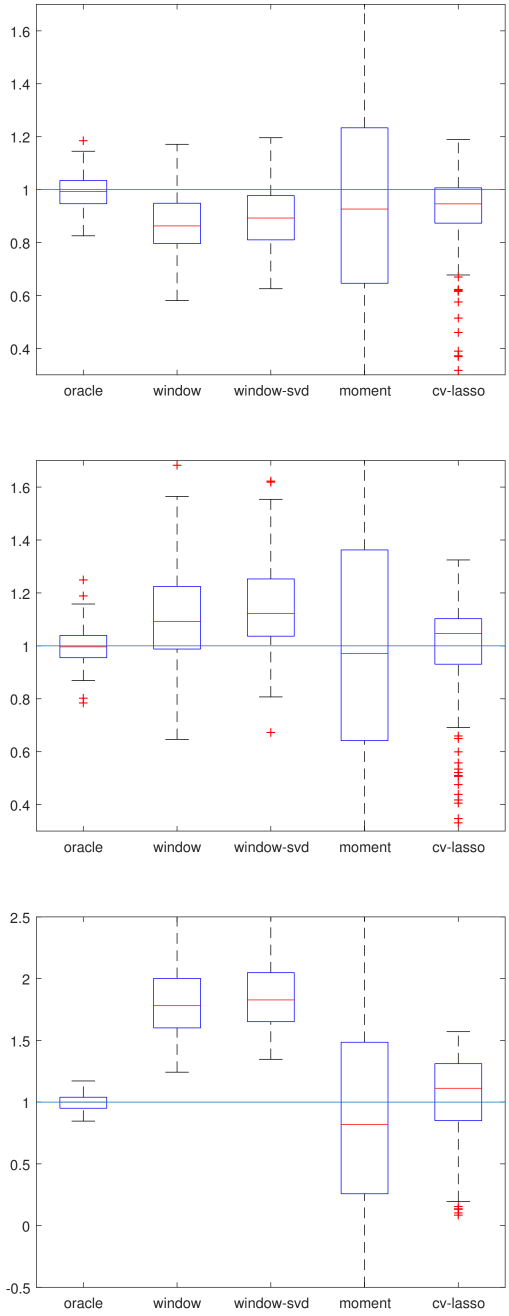

Figure 3.5: LASSO estimators with dense signal.

3.5 Real Data

In this section we report results on real data sets well suited for LASSO. Typical data sets where $p \gg n$ involve genetics data, where the amount of genetic data recorded is much larger than the number of patients sampled.

The first data set is from [57] and corresponds to gene expression data. It is presented as a 102x6033 matrix, where each row is a sample from a single subject, and the columns are expression levels. We defer to the original paper for how precisely these values were computed. This data is regressed against a length 102 vector with 52 cancer patient (1) and 50 healthy patients(0). We also consider the well-know Golub data set [27], which is a gene expression data set from subjects with human acute myeloid(AML) and acute lymphoblastic leukemias(ALL). It is represented as 3571 expression levels over 72 patients, with 47 ALL subjects and 25 AML. The final data set is from Alon et al. [5], a 62x2000 matrix of gene expression data from colon tissue, 40 tumor 22 normal. Note that in all cases we have a small number of subjects (< 102) and thousands of gene expressions for each subject.

Since we have no knowledge of the true noise of the variance in real world data, we instead compare the noise variance computed for 10 fold CV-LASSO to that of our estimators, as well as the resulting λ parameter. These results are tabulated in table 3.2. We note that with the refined version of our scheme, the estimated variance and resulting λ parameter are close to the corresponding λ value for 1 standard error in CV-LASSO.

Data	σ CV-LASSO	σ GVE	σ Fast-GVE	λ 1-SE	λ GVE
[57]	0.4854	0.7254	105.279	0.05295	0.00429
[27]	0.8132	0.6772	375.82	0.0637	0.0276
[5]	0.7788	1.212	8.31E+09	0.1503	0.0861

Table 3.2: σ and λ values for real data sets.

We also plot, in figures 3.6-3.8 the corresponding curves for the mean squared error of the LASSO solution, using the λ parameters from table 3.2.

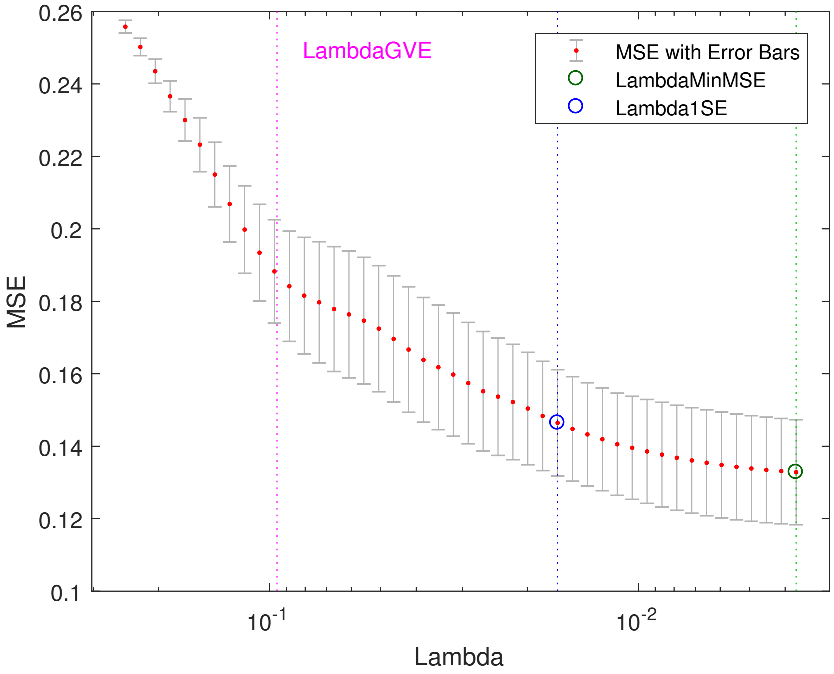

Figure 3.6: MSE for 10-fold CV LASSO using data from [57], with the λ value given by the estimator from Algorithm 2 marked in magenta.

3.6 LASSO Proofs

3.6.1 Proof Ingredients

Proposition 22. *(Lemma 1 in [39]) Suppose Z has a chi-squared distribution with d degrees of freedom. Then,*

$$\Pr[d - 2\sqrt{dt} \leq Z \leq d + 2\sqrt{dt} + 2t] \geq 1 - 2e^{-t} \qquad \forall t \geq 0. \tag{3.3}$$

Proposition 23. *(Proposition 2.5 in [52]) Suppose $\Omega_u \cap \Omega_v = \emptyset$, and that $X \in \mathbb{R}^{n \times p}$ satisfies RIP of order s_0 and level $\delta > 0$ with $s_0 = |\Omega_u| + |\Omega_v|$. Then,*

$$\|X_{\Omega_u}^T X_{\Omega_v}\|_{2\to 2} \leq \sqrt{\delta} \tag{3.4}$$

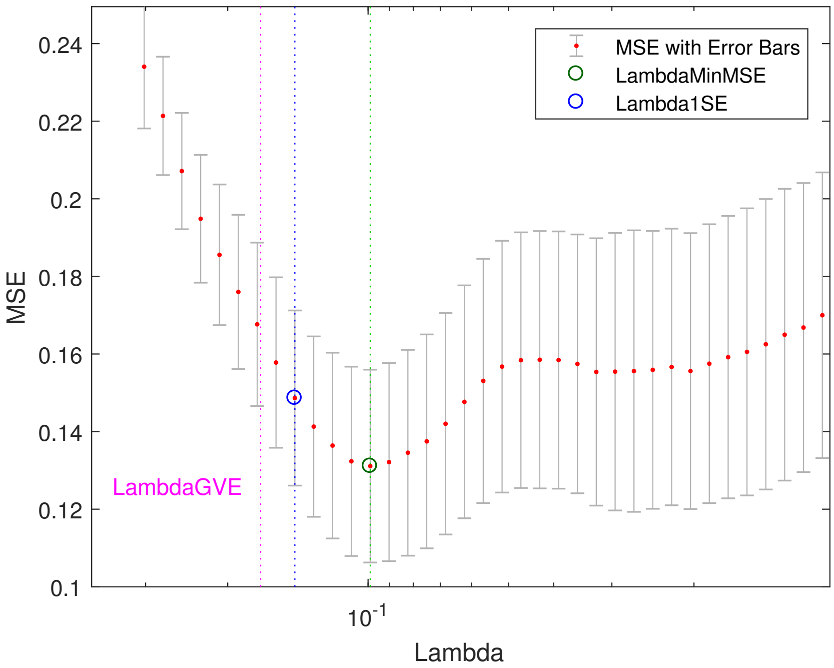

Figure 3.7: MSE for 10-fold CV LASSO using data from [5].

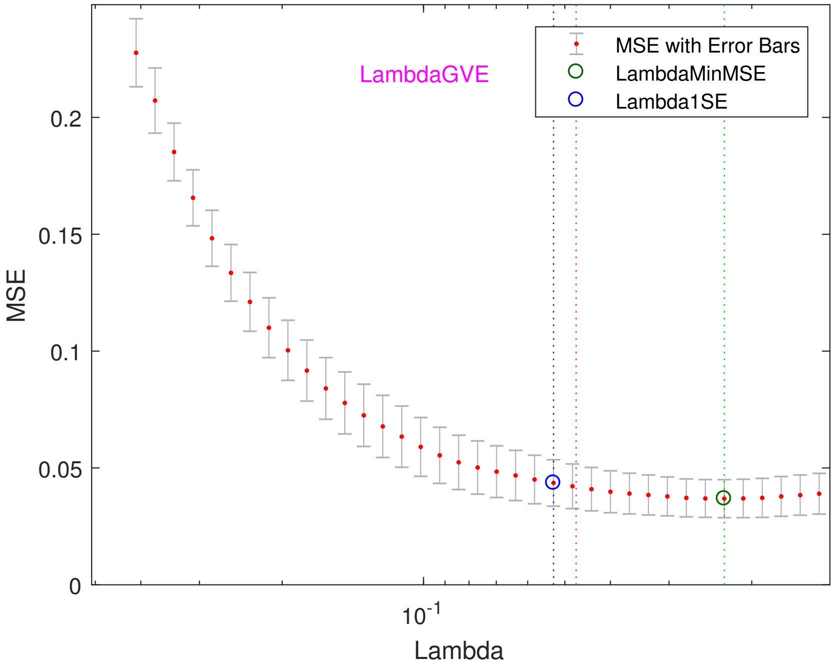

Figure 3.8: MSE for 10-fold CV LASSO using data from [27].

Proposition 24. *(Equation (5.5) in [63]) Let X be a Gaussian random variable with mean 0, variance σ. Then,*

$$\Pr[|X| > t] \leq 2e^{-t^2/2\sigma^2}, \qquad t \geq 1.$$

3.6.2 Proof of Theorem 16

Consider the window estimators

$$\begin{aligned} S_j &= \frac{1}{L}\sum_{i\in\Omega_j} |y_i|^2 \\ &= \frac{1}{L}\sum_{i\in\Omega_j} |\beta_i + \eta_i|^2 \\ &= \frac{1}{L}\sum_{i\in\Omega_j} |\beta_i|^2 + \frac{1}{L}\sum_{i\in\Omega_j} |\eta_i|^2 + 2\frac{1}{L}\sum_{i\in\Omega_j} \beta_i\eta_i. \end{aligned}$$

Set $E_j := \frac{1}{L}\sum_{i\in\Omega_j} |\eta_i|^2$. E_j is a sum of L independent squares of $\mathcal{N}(0, \sigma^2)$ random variables. Then E_j concentrates strongly around its expected value,

$$\mathbb{E}(E_j) = \sigma^2.$$

Note that E_j has a chi-squared distribution with L degrees of freedom, so by (3.3) with the choice $t = \log(p)^2$ and after a union bound over all p/L windows, we get that with probability at least $1 - \frac{2}{p}$,

$$\left(1 - \frac{5}{\log(p)}\right)\sigma^2 \leq E_j \leq \left(1 + \frac{5}{\log(p)}\right)\sigma^2,$$

holds uniformly for all $j \in \{1, 2, \dots, p/L\}$, assuming that $L \geq \log^3(p)$.

Since $L \leq \frac{p}{2s}$ by assumption, the pigeon hole principle implies that at least $\frac{p}{2L}$ windows do not overlap Ω_β. On any such "good" window k we have

$\|\beta_{k:k+L-1}\|_2^2 = 0$ and hence

$$|S_k - \sigma^2| \leq \frac{5\sigma^2}{\log(p)}. \tag{3.5}$$

Thus, if $\overline{S}$ is the average over a subset of the good windows, then also $|\overline{S}-\sigma^2| \leq \frac{5\sigma^2}{\log(p)}$.

Now, to bound the estimator above on *any* window, we need some control on the cross term $\sum_{i\in\Omega_j} \beta_i\eta_i$. Note that this quantity is just a sum of i.i.d. Gaussians with mean zero and with variance $\|\beta_{\Omega_j\cap\Omega_\beta}\|_2^2\sigma^2$; thus, by concentration, we have that with probability at least $1-2/p$, the following holds uniformly over all windows:

$$\sum_{i\in\Omega_j} \beta_i\eta_i \leq \frac{2\sigma\|\beta_{\Omega_j\cap\Omega_\beta}\|_2\sqrt{\log(p)}}{L}. \tag{3.6}$$

Hence, for any *any* window,

$$\begin{aligned}
S_j &\geq \frac{1}{L}\|\beta_{\Omega_j\cap\Omega_\beta}\|_2^2 + E_j - \frac{2}{L}\sum_{i\in\Omega_j\cap\Omega_\beta} \beta_i\eta_i \\
&\geq \frac{1}{L}\|\beta_{\Omega_j\cap\Omega_\beta}\|_2^2 + \left(1-\frac{5}{\log(p)}\right)\sigma^2 - \frac{\|\beta_{\Omega_j\cap\Omega_\beta}\|_2}{\sqrt{L}}\frac{2\sigma\sqrt{\log p}}{\sqrt{L}} \\
&\geq \left(1-\frac{5}{\log(p)}\right)\sigma^2 - \frac{\sigma^2\log(p)}{L} \\
&\geq \left(1-\frac{6}{\log(p)}\right)\sigma^2,
\end{aligned} \tag{3.7}$$

where the final inequality holds because $\log^2(p) \leq L$.

Now, consider the surrogate estimator $\widehat{\sigma_S^2} = \frac{2L}{p}\sum_{j=1}^{p/(2L)} S_{(j)}$. By construction, $\widehat{\sigma_S^2} \leq \overline{S}$, where $\overline{S}$ is the average over any $p/(2L)$ "good" windows. From the above analysis, we have that with probability exceeding $1-\frac{4}{p}$,

$$|\widehat{\sigma_S^2} - \sigma^2| \leq \frac{6}{\log(p)}\sigma^2.$$

Thus, for our final estimator, $\widehat{\sigma_S^2} = (1 + \frac{1}{\log(p)})\widehat{\sigma^2}$, we have

$$|\widehat{\sigma^2} - \sigma^2| \leq \left(\frac{7}{\log(p)} + \frac{6}{(\log(p))^2}\right)\sigma^2$$

3.6.3 Proof of Theorem 17

Recall that $\tilde{y} := Z^T y \in \mathbb{R}^p$. Consider the window estimate

$$\begin{aligned} S_j &= \frac{1}{L}\sum_{i\in\Omega_j} |\tilde{y}_i|^2 \\ &= \frac{1}{L}\sum_{i\in\Omega_j} |(Z^T X\beta)_i|^2 + \frac{1}{L}\sum_{i\in\Omega_j} |Z_i^T \eta|^2 + \frac{2}{L}\sum_{i\in\Omega_j} (Z^T X\beta)_i (Z^T\eta)_i \\ &= \frac{1}{L}\|Z_{\Omega_j}^T X_{\Omega_\beta}\beta\|_2^2 + \frac{1}{L}\|Z_{\Omega_j}^T \eta\|_2^2 + \frac{2}{L}\sum_{i\in\Omega_j} (Z^T X\beta)_i (Z^T\eta)_i \end{aligned} \tag{3.8}$$

The first term is small if Ω_j and Ω_β have disjoint support, since X has the RIP, the center term gets close to its expectation σ^2 due to standard concentration inequalities, and the third term is also small due to standard concentration inequalities. More concretely, if we assume that S_j is a “good” window, meaning that Ω_j and Ω_β have disjoint support, by equation (3.4)

$$\frac{1}{L}\|X_{\Omega_j}^T X_{\Omega_\beta}\beta\|_2^2 \leq \frac{\delta\|\beta\|_2^2}{L}. \tag{3.9}$$

All of the diagonal entries of Σ_j are in the range $[\sqrt{1-\delta}, \sqrt{1+\delta}]$, hence by (3.9)

$$\begin{aligned} \frac{1}{L}\|Z_{\Omega_j}^T X_{\Omega_\beta}\beta\|_2^2 &\le \frac{1+\delta}{L}\|X_{\Omega_j}^T X_{\Omega_\beta}\beta\|_2^2 \\ &\le \frac{\delta(1+\delta)\|\beta\|_2^2}{L} \\ &\le \frac{2\delta\|\beta\|_2^2}{L} \end{aligned} \tag{3.10}$$

For the center term, note that $\|Z_{\Omega_j}\eta\|_2^2 = \|P_L\eta\|_2^2$ where P_L is projection onto the first L coordinates. Next, we know that $\|P_L\eta\|_2^2$ has a chi-squared distribution with L degrees of freedom, so by (3.3) with $t = \log(p^2)$,

$$\Pr\left[|\|P_L\eta\|_2^2 - L\sigma^2| \le 2\sigma^2\left(\sqrt{L\log(p)} + \log(p)\right)\right] \ge 1 - \frac{2}{p^2}.$$

Hence by a union bound, with probability at least $1 - \frac{2}{p}$, the following holds uniformly over all windows:

$$\begin{aligned} |\|Z_{\Omega_j}^T\eta\|_2^2/L - \sigma^2| &= |\|P_L\eta\|_2^2/L - \sigma^2| \\ &\le 2\sigma^2\sqrt{\frac{\log(p)}{L}} + 2\sigma^2\frac{\log(p)}{L} \\ &\le \frac{5\sigma^2}{\log(p)} \end{aligned} \tag{3.11}$$

For the final term in 3.8, note that $\frac{2}{L}\sum_{i\in\Omega_j}(Z^TX\beta)_i(Z^T\eta)_i$ is a Gaussian random variable with variance $2\sigma\|Z_{\Omega_j}X_\beta\beta\|_2/L$. Thus, by Proposition 24 and (3.10), the following holds uniformly over all windows with probability at least $1 - \frac{1}{p}$:

$$\frac{2}{L}\sum_{i\in\Omega_j}(Z^T X\beta)_i(Z^T\eta)_i \le \frac{4\sigma\|Z_{\Omega_j}X_\beta\beta\|_2\sqrt{\log(p)}}{L} \tag{3.12}$$

$$\le \frac{8\sqrt{\delta}\sigma\|\beta\|_2\sqrt{\log(p)}}{L}, \tag{3.13}$$

Thus, averaging over any set of $p/2L$ "good" windows, using (3.10) (3.11) and (3.13) we have

$$\left|\frac{2L}{p}\sum_j S_j - \sigma^2\right| \le \frac{2\delta\|\beta\|_2^2}{L} + \frac{5\sigma^2}{\log p} + \frac{8\sqrt{\delta}\sigma\|\beta\|_2\sqrt{\log p}}{L} \tag{3.14}$$

with probability at least $1 - \frac{4}{p}$. Thus, by construction, the estimator $\widehat{\sigma_S^2} = \frac{2L}{p}\sum_j S_{(j)}$ also satisfies

$$\widehat{\sigma_S^2} \le \sigma^2 + \frac{2\delta\|\beta\|_2^2}{L} + \frac{5\sigma^2}{\log p} + \frac{8\sqrt{\delta}\sigma\|\beta\|_2\sqrt{\log p}}{L}.$$

It remains to show that the window estimator $\widehat{\sigma_S^2}$ cannot be too small. The inequalities (3.12) and (3.11) hold uniformly over all windows, not just good windows; hence, for any window S_j,

$$\begin{aligned}
S_j &\ge \frac{1}{L}\|Z_{\Omega_j}^T X_{\Omega_\beta}\beta\|_2^2 + \frac{1}{L}\|Z_{\Omega_j}^T\eta\|_2^2 - \frac{2}{L}\|Z_{\Omega_j}^T X_{\Omega_\beta}\beta\|_2\|X_{\Omega_j}^T\eta\|_2 \\
&\ge \frac{1}{L}\|X_{\Omega_j}^T X_{\Omega_\beta}\beta\|_2^2 + \sigma^2 - \frac{5\sigma^2}{\log(p)} - \frac{8\sigma\|X_{\Omega_j}^T X_{\Omega_\beta}\beta\|_2\sqrt{\log p}}{L} \\
&\ge \sigma^2 - \frac{5\sigma^2}{\log(p)} - \frac{4\sigma^2\log(p)}{L}.
\end{aligned}$$

Combining the bounds,

$$\begin{aligned}
-\frac{5\sigma^2}{\log(p)} - \frac{4\sigma^2\log(p)}{L} &\le \frac{2L}{p}\sum_j S_{(j)} - \sigma^2 \\
&\le \frac{2\delta\|\beta\|_2^2}{L} + \frac{5\sigma^2}{\log p} + \frac{8\sqrt{\delta}\sigma\|\beta\|_2\sqrt{\log p}}{L}
\end{aligned}$$

For our final estimator $\widehat{\sigma^2} = (1+\frac{1}{\log(p)})\widehat{\sigma^2_S}$, we have

$$\begin{aligned}
&|\widehat{\sigma^2} - \sigma^2| \le \\
&(1+\frac{1}{\log(p)})\left(2\delta\frac{\|\beta\|^2}{L} + \frac{6\sigma^2}{\log(p)} + \frac{1}{L}\max(4\sigma^2\log(p), 8\sqrt{\delta}\sigma\|\beta\|^2\sqrt{\log(p)})\right)
\end{aligned}$$

Chapter 4

Locality Sensitive Hashing

This section is based on work that appears in the publication [34]. **Nearest neighbor search (NN)** is a recently popular task of retrieving the nearest point in some point set to a given query point. The typical regime of this problem is that there are many points and they are in very high dimension. To be more precise: given a metric space $(X, \mathcal{D})$ and a set of points $P = \{x_1, ..., x_n\} \subset X$, for a query point $x \in P$ find $y = \operatorname{argmin}_{x_i \in P \setminus \{x\}} \mathcal{D}(x_i, x)$. Typically, $X = \mathbb{R}^d, S^{d-1}$, or $\mathbb{F}_2^n$ and $\mathcal{D}$ is some ℓ_p, cosine similarity, or χ^2 distance. The above problem is also known as exact nearest neighbor search, because we want to know the single minimal nearest neighbor in P. In particular, it was shown in [68] that when d is large, popular partioning/clustering techniques are outperformed by brute force search (that is, computing the pairwise distance of every point to the query point).

In order to improve performance, it is often enough in practice to solve an approximate version of nearest neighbor search named (R, c) **nearest neighbor search** (**(R, c)-NN**): given a query point $x \in P$ and the assurance of a point $y' \in P$ such that $\mathcal{D}(y', x) < R$, find $y \in P$ such that $\mathcal{D}(y, x) < cR$. Note that instead of solving the exact nearest neighbors problem (which degrades to linear search in high dimensions), by solving approximate nearest neighbor search we can achieve *sublinear* query time using **locality sensitive hashing** (LSH). The idea in LSH is to specify a function from the domain X to a discrete set of hash values – a *hash function* – which sends closer points to the same *hash value* with higher probability than points which are far apart. Then, for a set of points $P = \{x_1, ..., x_n\} \subset X$ and a query point $x \in P$, search within its corresponding hash bucket for a nearest neighbor.

The above discussion begs the obvious question: what makes LSH good for (R, c)-NN, and how can we quantify this? First we need a notion of sensitivity for our hash functions.

Definition 25. *For $r_1 \leq r_2$ and $p_2 \leq p_1$, a hash family $\mathcal{H}$ is (r_1, r_2, p_1, p_2)-**sensitive** if for all $x, y \in S^{d-1}$,*

- *If $\|x - y\|_2 \leq r_1$, then $\Pr_{\mathcal{H}}[h(x) = h(y)] \geq p_1$.*
- *If $\|x - y\|_2 \geq r_2$, then $\Pr_{\mathcal{H}}[h(x) = h(y)] \leq p_2$.*

Intuitively, this measures how often a hash function maps close points to the same value, and far points to different values. The more sensitive a hash function is (i.e. p_1 is close to 1, p_2 is close to 0 for some fixed r_1, r_2), the more effective it should be for the (R, c)-NN problem. We primarily care about the case where $r_1 = R$, $r_2 = cR$, in which case we study the parameter

$$\rho = \frac{\ln(p_1^{-1})}{\ln(p_2^{-1})}, \tag{4.1}$$

which quantifies sensitivity. The key result, which directly links the sensitivity of a hash family to how well it performs for (R, c)-NN search is the following (a result of this type first appeared in [31] but we use a more recent version with improved bounds).

Theorem 26. *(Theorem 1 in [20]) Given an (R, cR, p_1, p_2)-sensitive hash family $\mathcal{H}$, then there exists an algorithm that solves $(R, c) - NN$ with constant probability, using $O(dn+n^{1+\rho})$ space, with query time $O(n^\rho)$, and $O(n^\rho \ln_{1/p_1} n)$ evaluations of hash functions from $\mathcal{H}$.*

The above algorithm stores L hash tables from the family $\mathcal{G}$, where each $g \sim \mathcal{G}$ is given by $g(x) = (h_1(x), ..., h_k(x))$, and $h_i \sim \mathcal{H}, i = 1...k$. Then, given a query point $x \in X$, the algorithm looks for collisions in the buckets $g_1(x), ..., g_L(x)$. The choice of parameters $k = n^\rho$, $L = \ln_{1/p_1} n$ ensure that the algorithm solves (R, c)-NN with constant probability.

4.1 LSH Schemes

It should be clear from above that the correlation between ρ and R is a key feature in determining how effective a hash function is for LSH. To see this in a simple example, consider the hash function for $X = \mathbb{S}^{d-1}$, $\mathcal{D}$ is the angular distance, defined by

$$h(x) = \text{sign}(\langle \mathcal{G}, x \rangle),$$

where $\mathcal{G} \in \mathbb{R}^{d \times d}$ is a random Gaussian matrix with i.i.d. $\mathcal{N}(0, 1)$ entries (equivalently, A could be chosen according to the Haar measure on the rotation group

$SO(d)$). This rounding map traces back to Goemans and Williamson [26] and was introduced in the context of LSH by Charikar [17]. It has the advantage of being incredibly easy to implement, however has two main drawbacks: the $O(d^2)$ matrix/vector multiplication is slow, and it has suboptimal sensitivity. To see this, observe that the above hash function is equivalent to if we first project onto a random 2-dimensional hyperplane, then hash to a line with uniformly random angle with the x-axis. We can compute

$$\Pr[h(x) = h(y)] = 1 - \theta(x, y)/\pi,$$

where $\theta(x, y)$ is the angular distance between x and y. Consequently, for this scheme, if $\|x - y\|_2 = R$ and for fixed $c > 0$,

$$\rho = \frac{\ln(1 - R)}{\ln(1 - cR)} \leq \frac{1}{c}. \tag{4.2}$$

Moreover, $\rho \uparrow \frac{1}{c}$ as $R \to 0$. However, for the case of the unit sphere with euclidean metric, the optimal sensitivity $\rho = \frac{1}{c^2}$ is given in [48]. Spherical lsh ([8], [9]) has been shown to satisfy this, however the corresponding hash functions are not practical to compute. The work [7] showed the existence of an LSH scheme with optimally sensitive hash functions which are practical to implement; namely, the *cross-polytope* LSH scheme which has been previously proposed in [58] (see also [10], [48], [46]). Given a Gaussian matrix $\mathcal{G} \in \mathbb{R}^{d \times d}$ with i.i.d. $\mathcal{N}(0, 1)$ entries, the cross polytope hash of a point $x \in S^{d-1}$ is defined as

$$h(x) = \operatorname*{argmin}_{u = \{\pm e_i\}} \left\| \frac{\mathcal{G}x}{\|\mathcal{G}x\|_2} - u \right\|_2, \tag{4.3}$$

where $\{e_i\}_{i=1}^d$ is the standard basis for $\mathbb{R}^d$. Specifically, the name "cross polytope" arises (as with a few other hashing schemes) as the convex hull of the vertex set $\{\pm e_i\}_{i=1}^d$. A recent paper of Andoni, Indyk, Laarhoven, and Razenshteyn [7] gives the following collision probability for cross-polytope LSH.

Proposition 27 (Theorem 1 in [7])**.** *Suppose $x, y \in S^{d-1}$ are such that $\|x - y\|_2 = R$, with $0 < R < 2$, and $\mathcal{H}$ is the hash family defined in (4.3). Then,*

$$\ln\left(\frac{1}{\Pr_{\mathcal{H}}[h(x) = h(y)]}\right) = \frac{R^2}{4 - R^2}\ln d + \mathcal{O}_R(\ln(\ln d)). \tag{4.4}$$

Consequently,

$$\rho = \frac{1}{c^2}\frac{4 - c^2R^2}{4 - R^2} + o(1).$$

Remark 28. *The above proposition that cross-polytope LSH is asymptotically optimal with respect to ρ. In fact, the coefficient $\frac{4-c^2R^2}{4-R^2} < 1$ for every choice of $c > 1$ and $0 < R < 2$, but this does not break the lower bound given in [48] since the lower bound $\rho = \frac{1}{c^2}$ only holds for a particular sequence $R = R(d)$. For cross-polytope LSH and the schemes that follow, any sequence $R(d) \to 0$ suffices.*

Still, this scheme is limited in efficiency by the $\mathcal{O}(d^2)$ computation required to compute a dense matrix-vector multiplication in (4.3). To reduce this computation, [7] proposed to to use a pseudo-random rotation in place of a dense Gaussian matrix, namely,

$$h(x) = \operatorname{argmin}_{u=\{\pm e_i\}} \|HD_bHD_{b'}HD_{b''}x - u\|_2\,, \tag{4.5}$$

where $H \in \mathbb{R}^{d\times d}$ is a Hadamard matrix and $D_b, D_{b'}, D_{b''} \in \mathbb{R}^{d\times d}$ are independent diagonal matrices with i.i.d. Rademacher entries on the diagonal. This scheme has the advantage of computing hash functions in time $\mathcal{O}(d \ln d)$, and was shown in [7] to *empirically* exhibit similar collision probabilities to cross-polytope LSH, but provable guarantees on the asymptotic sensitivity of this fast variant of the standard cross-polytope LSH remain open.

4.1.1 Fast cross-polytope LSH with optimal asymptotic sensitivity

While we do not prove theoretical guarantees regarding the asymptotic sensitivity of the particular fast variant (4.5), we construct a different variant of the standard cross-polytope LSH (defined below in (4.6)) which also enjoys $\mathcal{O}(d \ln d)$ matrix-vector multiplication, and for which we are able to prove optimal asymptotic sensitivity $\rho = \frac{1}{c^2}$:

$$h_F(x) = \operatorname*{argmin}_{u=\{\pm e_i\}} \left\| \frac{\mathcal{G}(H_S D_b x)}{\|\mathcal{G}(H_S D_b x)\|_2} - u \right\|_2 ; \tag{4.6}$$

Here, $D_b : \mathbb{R}^d \to \mathbb{R}^d$ is a diagonal matrix with i.i.d. Rademacher entries on the diagonal, $H_S \in \mathbb{R}^{m\times d}$ is a partial Hadamard matrix restricted to a random subset $S \subset [d]$ of $|S| = m = \mathcal{O}(\log(d))$ rows, and $\mathcal{G} : \mathbb{R}^m \to \mathbb{R}^{d'}$ is a Gaussian matrix that lifts and rotates in dimension d' in the range $m \leq d' \leq d$. There is nothing special about lifting to dimension d, and indeed one could lift to dimension $d' > d$, but if d' grows faster than d, the hash computation no longer takes time $\mathcal{O}(d \ln d)$.

The embedding $H_S D_b x$ acts as a Johnson-Lindenstrauss (JL) transform[1], and embeds the points in dimension $m \approx \ln d$.

It is straightforward that the hash computation $x \to h_F(x)$ takes $\mathcal{O}(d'm)$ time from the Gaussian matrix multiplication and $\mathcal{O}(d \ln d)$ time from the JL transform. We will show that optimal asymptotic sensitivity is still achieved without lifting, $d' = m$, but we observe both empirically and theoretically that the *rate of convergence* to the asymptotic sensitivity improves by lifting to higher dimension; taking d' closer to d results in empirically closer results to the standard cross-polytope scheme (see section 4.6 for more details). Moreover, our scheme achieves the lower bound given by Theorem 2 in [7] for the fastest rate of convergence among all hash families which has to d' values.

4.1.2 Fast cross-polytope LSH with optimal asymptotic sensitivity and few random bits

Aiming to construct a hash family with similar guarantees which also uses as little randomness as possible, we also consider a discretized version of the fast hashing scheme (4.6) in which the Gaussian matrix $\mathcal{G} \in \mathbb{R}^{d' \times m}$ is replaced by a matrix $\widehat{\mathcal{G}} \in \mathbb{R}^{d' \times m}$ whose entries are i.i.d. discrete approximations of a Gaussian; in place of the "standard" fast JL transform $H_S D_b$, we consider $Z \in \mathbb{R}^{d \times m}$ a low-randomness JL transform that we will clarify later. Then, the discrete fast hashing scheme we consider is

[1] Formally, given a finite metric space $(X, \|\cdot\|) \subset \mathbb{R}^d$, a JL transform is a linear map $\Phi : \mathbb{R}^d \to \mathbb{R}^m$ such that for all $x \in X$, $(1-\delta)\|x\|^2 \leq \|\Phi x\|^2 \leq (1+\delta)\|x\|^2$, with $m \ll d$ close to the optimal scaling $m = C\delta^{-2}\ln(|X|)$ [32] [4] [38].

$$h_D(x) = \operatorname*{argmin}_{u=\{\pm e_i\}} \left\| \frac{\widehat{\mathcal{G}}(Zx)}{\|\widehat{\mathcal{G}}(Zx)\|_2} - u \right\|_2 . \tag{4.7}$$

Also for this scheme, the hash computation $x \to h(x)$ takes $\mathcal{O}(d'm)$ time from the Gaussian matrix multiplication and $\mathcal{O}(d \ln d)$ time from the JL transform. Our scheme has several advantages, due to the fact that the choice of d' in the range $d \leq d' \leq m$ is flexible: To summarize our main contributions, we prove for both the fast cross-polytope LSH and the fast discrete cross-polytope LSH,

- For each d' in the range $m \leq d' \leq d$, this scheme achieves the asymptotically optimal ρ. Moreover, for $d' = d$, the rate of convergence to this ρ is optimal over all hash families with d hash values.
- With the choice $d' = d$, the scheme computes hashes in time $\mathcal{O}(d \ln d)$ and performs well empirically compared to the standard cross-polytope with dense Gaussian matrix.
- With the choice $d' = m$, and by discretizing the Gaussian matrix, we arrive at a scheme that has only $\mathcal{O}(\ln^9(d))$ bits of randomness and still has optimal asymptotic sensitivity.

Table 4.1 contains the construction of the original cross-polytope LSH scheme, our fast cross-polytope scheme, as well as the discretized version.

Table 4.1: Various LSH Families and corresponding Hash Functions.

LSH Family	Hash Function
Cross-Polytope LSH	$h(x) = \operatorname*{argmin}_{u=\{\pm e_i\}} \left\lVert \frac{\mathcal{G}x}{\lVert \mathcal{G}x \rVert_2} - u \right\rVert_2,$ $\mathcal{G} \in \mathbb{R}^{d \times d}$
Fast **Cross-Polytope LSH**	$h_F(x) = \operatorname*{argmin}_{u=\{\pm e_i\}} \left\lVert \frac{\mathcal{G}(H_S D_b x)}{\lVert \mathcal{G}(H_s D_b x) \rVert_2} - u \right\rVert_2,$ $\mathcal{G} \in \mathbb{R}^{d' \times m}$
Fast Discrete **Cross-Polytope LSH**	$h_D(x) = \operatorname*{argmin}_{u=\{\pm e_i\}} \left\lVert \frac{\widehat{\mathcal{G}}(Zx)}{\lVert \widehat{\mathcal{G}}(Zx) \rVert_2} - u \right\rVert_2,$ $\widehat{\mathcal{G}} \in \mathbb{R}^{d' \times m}$

4.2 LSH Results

We now formalize the intuition about how our scheme behaves relative to cross-polytope LSH.

Theorem 29. *Suppose $\mathcal{H}$ is the family of hash functions defined in (4.6) with the choice $m = \mathcal{O}(\ln^5(d) \ln^4(\ln d))$, and ρ is as defined in (4.1) for this particular family. Then we have*

(i-)

$$\rho = \frac{1}{c^2} \frac{4 - c^2 R^2}{4 - R^2} + o(1).$$

and this hashing scheme runs in time $\mathcal{O}(d \ln d)$.

Moreover, we have the optimal rate of convergence,

(ii-)

$$\rho = \frac{1}{c^2} \frac{4 - c^2 R^2}{4 - R^2} + \mathcal{O}\left(\frac{1}{\ln d'}\right).$$

The lower bound given by Theorem 2 in [7] verifies the above rate of convergence is in fact optimal. We remark that when hashing n points simultaneously, the embedded dimension m picks up a factor of $\ln(n)$. Assuming that n is polynomial in d, the result in Theorem 29 still holds simultaneously over all pairs of points.

In addition to creating a fast hashing scheme, one can reduce the amount of randomness involved. In particular, we show that a slight alteration of the scheme still achieves the optimal ρ-value while using only $\mathcal{O}(\ln^9 d)$ bits of randomness. The idea is to replace the Gaussian matrix by a matrix of i.i.d. discrete random variables. Some care is required in tuning the size of this matrix so that the correct number of bits is achieved. As a consequence the number of hash values for this scheme is of order $\mathcal{O}(m)$ (i.e. we lift up to a smaller dimension), which lowers performance in practice, but does not affect the asymptotic sensitivity ρ. We additionally use a JL transform developed by Kane and Nelson [33] that only uses $\mathcal{O}(\ln(d)\ln(\ln d))$ bits of randomness. Specifically, the hash function for this scheme is

$$h_D(x) = \operatorname*{argmin}_{u=\{\pm e_i\}} \left\| \frac{\widehat{\mathcal{G}}(Zx)}{\|\widehat{\mathcal{G}}(Zx)\|_2} - u \right\|_2$$

where $\widehat{\mathcal{G}} \in \mathbb{R}^{d' \times m}$ is a matrix with i.i.d. copies of a discrete random variable X which roughly models a Gaussian, and $Z \in \mathbb{R}^{d \times m}$ is the JL transform constructed in [33]. Our analysis allows us to pick the threshold value $d' = m$ to minimize the number of random bits.

Theorem 30. *There is a hash family $\mathcal{H}$ with $\mathcal{O}(\ln^9 d)$ bits of randomness that* *achieves the bound*

$$\rho = \frac{1}{c^2}\frac{4 - c^2R^2}{4 - R^2} + o(1),$$

and runs in time $\mathcal{O}(d \ln d)$.

4.3 Theorem 29 Proof Outline

First we state an elementary limit result that we will apply to the proofs of both Theorem 29 and Theorem 30.

Lemma 31. *Suppose* $m_d(a), m_d(b)$ *are positive functions,* $\lim_{d\to\infty} m_d(a) = a$*,* $\lim_{d\to\infty} m_d(b) = b$*, and that* $f(d), g(d)$ *are also positive,* $\lim_{d\to\infty} f(d) = \lim_{d\to\infty} g(d) = \infty$*,* $\lim_{d\to\infty} \frac{f(d)}{g(d)} = \infty$*. Then,*

$$\lim_{d\to\infty} \frac{m_d(a)f(d) + g(d)}{m_d(b)f(d) + g(d)} = \frac{a}{b}$$

Proceeding to the proof of Theorem 29, the key observation is that for $x, y \in S^{d-1}$, $\mathcal{G}\tilde{x} = \mathcal{G}_0 \begin{bmatrix} \tilde{x} \\ 0 \end{bmatrix}$, where $\mathcal{G}_0 \in \mathbb{R}^{d' \times d'}$ is a square Gaussian matrix. Thus,

$$\Pr[h_f(x) = h_f(y)] = \Pr\left[h\left(\begin{bmatrix} \tilde{x} \\ 0 \end{bmatrix}\right) = h\left(\begin{bmatrix} \tilde{y} \\ 0 \end{bmatrix}\right)\right],$$

recalling that h_f is the fast cross-polytope hash function and h is the standard version. It then follows that, provided the distance between $\tilde{x}$ and $\tilde{y}$ is close to the distance between x and y, we can apply proposition 27 to control the above probability. We start with a lemma for our chosen JL transform that combines a recent improvement on the *restricted isometry property* (RIP) for partial

Hadamard matrices [28] with a reduction from RIP to Johnson-Lindenstrauss transforms in [37]; we defer the proof to the sequel.

Lemma 32. *Suppose $\gamma > 0$, $x, y \in S^{d-1}$, $\widetilde{x} = H_S D_b x$, $\widetilde{y} = H_S D_b y$ and $H_S \in \mathbb{R}^{m\times d}$ is such that $m = \mathcal{O}(\gamma \ln^4(d)\ln^4(\ln d))$. Then with probability $1 - \mathcal{O}(d^{-\gamma})$,*

$$\left(1 - \frac{1}{\ln d}\right) \leq \|\widetilde{x}\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right), \tag{4.8}$$

$$\left(1 - \frac{1}{\ln d}\right) \leq \|\widetilde{y}\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right), \tag{4.9}$$

$$\left(1 - \frac{1}{\ln d}\right)\|x-y\|_2^2 \leq \|\widetilde{x} - \widetilde{y}\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right)\|x-y\|_2^2 \tag{4.10}$$

We apply the above lemma with the choice $\gamma = \ln d$ to get that

$$\frac{\|x-y\|_2^2}{\left(1-\frac{1}{\ln d}\right)} - \frac{5}{\ln d - 1} \leq \left\| \frac{\widetilde{x}}{\|\widetilde{x}\|_2} - \frac{\widetilde{y}}{\|\widetilde{y}\|_2} \right\|_2^2 \leq \frac{\|x-y\|_2^2}{\left(1+\frac{1}{\ln d}\right)} + \frac{5}{\ln d + 1}. \tag{4.11}$$

with probability $1 - \mathcal{O}(d^{-\ln d})$. Combining this fact with proposition 27 we get that

$$\Pr[h_f(x) = h_f(y)] = C(d')^{\frac{-\tilde{R}^2}{4-\tilde{R}^2}} \ln^{-1}(d'),$$

where $\tilde{R} = \|\tilde{x} - \tilde{y}\|^2$ (by equation (4.11)) goes to R as $d \to \infty$, and C is bounded in the dimension. We then apply lemma 31 to see that

$$\begin{aligned} \rho &= \frac{\frac{\tilde{R}^2}{4-\tilde{R}^2}\ln(d') + \ln\ln(d') + C}{\frac{c^2\tilde{R}^2}{4-c^2\tilde{R}^2}\ln(d') + \ln\ln(d') + C} \\ &= \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2} + o(1). \end{aligned}$$

4.3.1 Proof of Theorem 29 Part (ii-)

Let $\rho_{R,c}$ be the exponent for standard cross-polytope lsh in dimension d', and $\rho_{R,c}^{fast}$ be the exponent for fast cross-polytope lsh lifted to dimension d'. Suppose that

$$\rho_{R,c} - \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2} \leq C(R,c)F(d'),$$

where $F(d') \to 0$ as $d' \to \infty$ and $C(r,c)$ is constant in the dimension d'. Assume that $H_sD_b : \mathbb{R}^d \to \mathbb{R}^m$ is a δ-isometry on $x-y$, i.e.

$$||x-y||_2^2 \leq R^2 \implies ||\tilde{x}-\tilde{y}||_2^2 \leq (1+\delta)R^2 \tag{4.12}$$

$$||x-y||_2^2 \geq c^2R^2 \implies ||\tilde{x}-\tilde{y}||_2^2 \geq (1-\delta)c^2R^2. \tag{4.13}$$

The next observation is that $h_f(x)$ applies the standard cross-polytope lsh scheme on H_sD_bx, so conditioned on H_sD_bx being a δ-isometry, we can analyze the fast scheme in terms of the standard scheme as follows:

$$\rho_{R,c}^{fast} \leq \rho_{R',c'},$$

where $R' = R\sqrt{1+\delta}$, $c' = \sqrt{\frac{1-\delta}{1+\delta}}c$. Now, we can say

$$\begin{aligned}\rho_{R,c}^{fast} - \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2} &\leq [\rho_{R,c}^{fast} - \rho_{R',c'}] + \left[\rho_{R',c'} - \frac{1}{(c')^2}\frac{4-(c')^2(r')^2}{4-(R')^2}\right] \\ &+ \left[\frac{1}{(c')^2}\frac{4-(c')^2(R')^2}{4-(R')^2} - \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2}\right] \\ &\leq C(R',c')F(d) + \left[\frac{1}{(c')^2}\frac{4-(c')^2(R')^2}{4-(R')^2} - \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2}\right].\end{aligned}$$

The difference in the last equation can be bounded as

$$
\begin{aligned}
&\frac{1}{(c')^2}\frac{4-(c')^2(R')^2}{4-(R')^2} - \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2} \\
&= \left(\frac{1+\delta}{c^2(1-\delta)}\right)\frac{4-(1-\delta)c^2R^2}{4-(1-\delta)R^2} - \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2} \\
&\leq \frac{(1+\delta)(4-(1-\delta)c^2R^2)(4-R^2)-(4-c^2R^2)(1-\delta)(4-(1-\delta)R^2)}{\frac{c^2}{2}(4-R^2)^2} \\
&= \delta\mathcal{O}(R,c) + \frac{(1+\delta)(4-c^2R^2)(4-R^2)-(1-\delta)(4-c^2R^2)(4-R^2)}{\frac{c^2}{2}(4-R^2)^2} \\
&= \delta D(R,c),
\end{aligned}
$$

so it follows that $\rho^{fast}_{R,c} - \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2} \leq \delta D(R,c) + C(R',c')F(d')$ conditioned on the fact that H_sD_b is a δ-isometry on $x-y$. Note that for d' large enough, $C(R',c')$ is bounded above by a constant independent of the dimension. We can make the choice $\delta = \frac{1}{\ln(d)}$, so that the isometry condition holds with probability $1-\mathcal{O}(d^{-\ln d})$, so if ρ is the true exponent without conditioning, we get that

$$
\begin{aligned}
\rho &\leq \frac{p_1}{p_2 + C\ln(1-d^{-\ln d})} \\
&\leq \frac{p_1}{p_2 - Cd^{-\ln d}} \\
&\leq \frac{p_1}{p_2}(1 + Cd^{-\ln d}/p_1),
\end{aligned}
$$

where $C > 0$ is an constant that changes by line but is independent of the dimension. From this expression it is easy to see that the error term decays at least like $1/\ln d'$ (recall that $d' \leq d$).

Finally, provided $F(d')$ decays as fast as than $\frac{1}{\ln(d')}$, the result will hold. This follows from Theorem 1 in [7].

4.4 Theorem 30 Proof Outline

We will use the following result (formulated as an analogue to lemma 32) , due to Kane and Nelson, that reduces the amount of randomness required to perform a JL transform.

Proposition 33. *(Theorem 13 and Remark 14 in [33]) Suppose* $\gamma > 0$*,* $x, y \in S^{d-1}$*. Then, there is a random matrix* $Z \in \mathbb{R}^{d\times m}$ *with* $m = \mathcal{O}(\gamma \ln^3(d))$ *and sampled with* $\mathcal{O}(\gamma \ln^2(d))$ *random bits such that with probability* $1 - \mathcal{O}(d^{-\gamma})$*,*

$$\left(1 - \frac{1}{\ln d}\right) \leq \|Zx\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right),$$
$$\left(1 - \frac{1}{\ln d}\right) \leq \|Zy\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right),$$
$$\left(1 - \frac{1}{\ln d}\right) \|x-y\|_2^2 \leq \|Z(x-y)\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right) \|x-y\|_2^2$$

Now we want to construct a hash scheme that uses a Gaussian rotation with which to compare our discretized scheme. Define

$$h'_D(x) = \underset{u=\{\pm e_i\}}{\operatorname{argmin}} \left\| \frac{\mathcal{G}'Zx}{\|\mathcal{G}'Zx\|_2} - u \right\|_2, \tag{4.14}$$

where $\mathcal{G}' \in \mathbb{R}^{m\times m}$ is a standard i.i.d. Gaussian matrix. The following elementary lemma gives us a suitable replacement for each Gaussian in the matrix $\mathcal{G}'$.

Lemma 34. *Suppose* $g \sim \mathcal{N}(0, 1)$*. Then, there is a symmetric, discrete random variable* X *taking* 2^b *values such that for any* $x \in \mathbb{R}$*,*

$$\Pr[g \leq x] = \Pr[X \leq x] + \mathcal{O}(2^{-b}) \tag{4.15}$$

The discretized scheme can now be constructed by

$$h_D(x) = \operatorname*{argmin}_{u=\{\pm e_i\}} \left\| \frac{\widehat{\mathcal{G}}Zx}{\|\widehat{\mathcal{G}}Zx\|_2} - u \right\|_2, \tag{4.16}$$

where the entries of $\widehat{\mathcal{G}} \in \mathbb{R}^{d' \times m}$ are i.i.d. copies of the random variable X in Lemma 34. Note that each discrete random variable has b bits of randomness, so the hashing scheme has minimial randomness when $d' = m$, thus there are $m \times m \times b + \mathcal{O}(\gamma \ln^2(d)) = \mathcal{O}(\gamma^2 \ln^6(d) b + \gamma \ln^2(d))$ bits of randomness. As we will see, we can choose γ and b to be a power of $\ln(d)$ while still achieve the optimal asymptotic ρ. For this we have the following lemma.

Lemma 35. *Let $x, y \in \mathbb{R}^d$ be such that $\|x - y\|_2 = R$, $\widetilde{x} = Zx$, and let h, h' be as defined in (4.16) and (4.14) respectively with $m = \mathcal{O}(\ln^4(d))$, $b = \log_2(d)$ where $\widetilde{R} = \|\widetilde{x} - \widetilde{y}\|_2$. Then,*

$$\ln(\Pr[h_D(x) = h_D(y)]) = \ln(\Pr[h'_D(x) = h'_D(y)]) + \mathcal{O}_{\widetilde{R}}(1) \tag{4.17}$$

We defer the proof of lemma 35 to the sequel, but the idea is as follows. We can first write

$$\Pr[h'_D(x) = h'_D(y)] = 2d' \Pr[h'_D(x) = h'_D(y) = e_1].$$

Note that the set $\{h'_D(x) = h'_D(y) = e_1\} = \{(\mathcal{G}'\widetilde{x})_1 \geq |(\mathcal{G}'\widetilde{x})_2|, (\mathcal{G}'\widetilde{y})_1 \geq |(\mathcal{G}'\widetilde{y})_2|\}$, which is the Gaussian measure of a convex polytope, so we can write the above probability as the integral over m intervals of the m-dimensional Gaussian probability distribution. We can then use equation (4.15) to replace the Gaussian pdf with the discrete Gaussian pdf in each coordinate succesively, and (keeping track of parameters), the lemma follows.

We now run the same argument as in Theorem 29 by setting $\gamma = \ln d$, so combining lemma 35 and proposition 27 applied to $h'_D(x)$, we have that

$$\begin{aligned}
\rho &= \frac{\ln(\Pr[h_D(x) = h_D(y)])}{\ln(\Pr[h_D(cx) = h_D(cy)])} \\
&= \frac{\ln(\Pr[h'_D(x) = h'(y)]) + \mathcal{O}_{\widetilde{R}}(1)}{\ln(\Pr[h'_D(cx) = h'(cy)]) + \mathcal{O}_{\widetilde{R}}(1)} \\
&= \frac{\frac{R_+^2}{4-R_+^2}\ln(d') + \ln\ln(d') + C + \mathcal{O}_{\widetilde{R}}(1)}{\frac{c^2R_-^2}{4-c^2R_-^2}\ln(d') + \ln\ln(d') + C + \mathcal{O}_{\widetilde{R}}(1)} \\
&= \frac{\frac{R_+^2}{4-R_+^2}\ln(d') + \ln\ln(d') + C}{\frac{c^2R_-^2}{4-c^2R_-^2}\ln(d') + \ln\ln(d') + C} \\
&= \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2} + o(1), \text{ by lemma 31.}
\end{aligned}$$

Finally, by our choice of γ and b in the above lemma, we know that there are $\mathcal{O}(\ln^9(d))$ bits of randomness.

4.5 Proofs of Lemmas

4.5.1 Proof of Lemma 31

We know that for any $\epsilon > 0$ and d large enough, $m_d(b) \geq b - \epsilon$, so that

$$\begin{aligned}
\lim_{d\to\infty} \frac{g(d)}{m_d(b)f(d) + g(d)} &\leq \lim_{d\to\infty} \frac{g(d)}{(b-\epsilon)f(d) + g(d)} \\
&= \lim_{d\to\infty} \frac{1}{(b-\epsilon)\frac{f(d)}{g(d)} + 1} = 0,
\end{aligned}$$

and by positivity the inequality is an equality. This implies that

$$\lim_{d\to\infty} \frac{m_d(a)f(d)+g(d)}{m_d(b)f(d)+g(d)} = \lim_{d\to\infty} \frac{m_d(a)f(d)}{m_d(b)f(d)+g(d)}.$$

The same argument on the reciprocal shows that

$$\lim_{d\to\infty} \frac{m_d(a)f(d)}{m_d(b)f(d)+g(d)} = \lim_{d\to\infty} \frac{m_d(a)f(d)}{m_d(b)f(d)} = \frac{a}{b}$$

4.5.2 Proof of Lemma 32

Define the event

$$E_{v,\delta} := \{v \in \mathbb{R}^n : (1-\delta)\|v\|_2 \leq \|\widetilde{v}\|_2 \leq (1+\delta)\|v\|_2\}.$$

Combining Theorem 4.5 of [28] and Theorem 3.1 of [37], we know that for any $\eta \in (0,1)$, any $s \geq 40\ln(12/\eta)$, some $C_0 > 0$, and provided $m = \mathcal{O}(\delta^{-2}\ln^2(1/\delta)s\ln^2(s/\delta)\ln(d))$,

$$\Pr[E_{x,\delta} \cap E_{y,\delta} \cap E_{x-y,\delta}] \geq (1-\eta)(1-2^{-C_0\ln(d)\ln(s/\delta)})$$

Setting $\delta = 1/\ln(d)$, $\eta = d^{-\gamma}$, $s = 40C\ln(12d)$, we get

$$\Pr[E_{x,\delta} \cap E_{y,\delta} \cap E_{x-y,\delta}] \geq (1-d^{-\gamma})(1-2^{-C_0\ln(d)\ln(40\gamma\ln(12d)\ln(d))}),$$

and the lemma follows.

4.5.3 Proof of Lemma 35

Note that since the entries of $\widehat{\mathcal{G}}\widetilde{x}$ are symmetric and i.i.d., the probability of hashing to one value is equal for all hash values, so we get

$$\begin{aligned}
&\Pr[h_D(x) = h_D(y)] = 2d'\Pr[h_D(x) = h_D(y) = e_1] \\
&= 2d'\Pr[\cap_{j=2}^{d'} (\widehat{\mathcal{G}}\widetilde{x})_1 \geq |(\widehat{\mathcal{G}}\widetilde{x})_j|, (\widehat{\mathcal{G}}\widetilde{y})_1 \geq |(\widehat{\mathcal{G}}\widetilde{y})_j|] \\
&= 2d'\mathbb{E}_{(\widehat{\mathcal{G}}\widetilde{x})_1,(\widehat{\mathcal{G}}\widetilde{y})_1}(\Pr[(\widehat{\mathcal{G}}\widetilde{x})_1 \geq |(\widehat{\mathcal{G}}\widetilde{x})_2|, (\widehat{\mathcal{G}}\widetilde{y})_1 \geq |(\widehat{\mathcal{G}}\widetilde{y})_2|]^{d'-1}). \qquad (4.18)
\end{aligned}$$

Our goal is to bound the probability $\Pr[(\widehat{\mathcal{G}}\widetilde{x})_1 \geq |(\widehat{\mathcal{G}}\widetilde{x})_2|, (\widehat{\mathcal{G}}\widetilde{y})_1 \geq |(\widehat{\mathcal{G}}\widetilde{y})_2|]$ in terms of the probability $\Pr[(\mathcal{G}'\widetilde{x})_1 \geq |(\mathcal{G}'\widetilde{x})_2|, (\mathcal{G}'\widetilde{y})_1 \geq |(\mathcal{G}'\widetilde{y})_2|]$. Define $E_{\mathcal{G}'} = \{(\mathcal{G}'\widetilde{x})_1 \geq |(\mathcal{G}'\widetilde{x})_2|, (\mathcal{G}'\widetilde{y})_1 \geq |(\mathcal{G}'\widetilde{y})_2|\}$ and similarly for $\widehat{\mathcal{G}}$. Since $E_{\mathcal{G}'}$ is a convex polytope, we can write

$$\Pr[E_{\mathcal{G}'}] = \int_{I_1}\int_{I_2(x_1)} \cdots \int_{I_m(x_1,x_2,\ldots,x_{m-1})} \frac{1}{(2\pi)^m} e^{-(x_1^2+\ldots+x_m^2)/2} dx_m \ldots dx_1,$$

where each $I_j(x_1, \ldots, x_j)$ is a (possibly unbounded) interval. By construction of X,

$$\int_{I_j(x_1,\ldots,x_j)} \frac{1}{2\pi} e^{-x_{j+1}^2/2} dx_{j+1} = \int_{I_j(x_1,\ldots,x_j)} p_X(x_{j+1}) dx_{j+1} + \mathcal{O}(2^{-b})$$

where $p_X(x)$ is the pdf of X. This implies that

$$\begin{aligned}
&\Pr[E_{\mathcal{G}'}] \\
&= \int_{I_1} \cdots \int_{I_m(x_1,\ldots,x_{m-1})} \frac{1}{(2\pi)^{m-1}} e^{-(x_1^2+\ldots+x_{m-1}^2)/2} p_X(x_m) dx_m \ldots dx_1 + \mathcal{O}(2^{-b}) \\
&\ldots = \int_{I_1} \cdots \int_{I_m(x_1,\ldots,x_{m-1})} p_X(x_1) \ldots p_X(x_m) dx_m \ldots dx_1 + \mathcal{O}(m2^{-b}) \\
&= \Pr[E_{\widehat{\mathcal{G}}}] + \mathcal{O}(m2^{-b}).
\end{aligned}$$

Plugging this into (4.18), we get

$$\begin{aligned}
\Pr[h_D(x) = h_D(y)] &= 2d'\mathbb{E}_{(\widehat{\mathcal{G}}\widetilde{x})_1,(\widehat{\mathcal{G}}\widetilde{y})_1}(\Pr[E_{\mathcal{G}'}] + \mathcal{O}(m2^{-b})))^{d'-1} \\
&= 2d'\mathbb{E}_{(\widehat{\mathcal{G}}\widetilde{x})_1,(\widehat{\mathcal{G}}\widetilde{y})_1}\left[\sum_{k=1}^{d'-1}\binom{d'-1}{k}\Pr[E_{\mathcal{G}'}]^k(\mathcal{O}(m2^{-b}))^{d'-1-k}\right].
\end{aligned}$$

We now make the choice $m = C\ln^4(d)$, $b = \log_2(d)\ln(d)$, so that the above summation becomes

$$\begin{aligned}
&\sum_{k=1}^{d'-1}\binom{d'-1}{k}\Pr[E_{\mathcal{G}'}]^{d'-1-k}(C\ln^4(d)d^{-\ln(d)})^k \\
&\qquad\qquad = \sum_{k=1}^{d'-1}\binom{d'-1}{k}\Pr[E_{\mathcal{G}'}]^{d'-1-k}(C\ln^4(d)d^{-\ln(d)})^k
\end{aligned}$$

This first term in the summation is the main term $\Pr[E_{\mathcal{G}'}]^{d'-1}$ and the other terms can be bounded using Sterling's approximation as follows,

$$\binom{d'-1}{k}\Pr[E_{\mathcal{G}'}]^{d'-1-k}(C\ln^4(d)d^{-\ln(d)})^k \leq \left(\frac{d'e}{k}\right)^k (C\ln^4(d)d^{-\ln(d)})^k.$$

For $k \geq 1$ this is certainly bounded by $\mathcal{O}(d^{-\ln(d)+1})$, and we have

$$\begin{aligned}
&\sum_{k=1}^{d'-1}\binom{d'-1}{k}\Pr[E_{\mathcal{G}'}]^{d'-1-k}(C\ln^4(d)d^{-\ln(d)})^k \\
&\qquad\qquad = \Pr[E_{\mathcal{G}'}]^{d'-1} + \mathcal{O}(d^{-\ln(d)+2})
\end{aligned}$$

We note that the last asymptotic approximation is very rough but sufficient for our purposes. This means that

$$\Pr[h_D(x) = h_D(y)] = 2d'\mathbb{E}_{(\widehat{\mathcal{G}}\widetilde{x})_1,(\widehat{\mathcal{G}}\widetilde{y})_1}(\Pr[E_{\mathcal{G}'}]^{d'-1}) + \mathcal{O}(md^{-\ln(d)+2}). \qquad (4.19)$$

Using the same technique as above where we replace the Gaussian density function with $P_X(x)$, we have

$$\begin{aligned}\Pr[h'_D(x) = h'_D(y)] &= 2d'\mathbb{E}_{(\mathcal{G}'\widetilde{x})_1,(\mathcal{G}'\widetilde{y})_1}(\Pr[E_{\mathcal{G}'}]^{d'-1})\\ &= 2d'\mathbb{E}_{(\widehat{\mathcal{G}}\widetilde{x})_1,(\widehat{\mathcal{G}}\widetilde{y})_2}(\Pr[E_{\mathcal{G}'}] + \mathcal{O}(m2^{-b}))^{d'-1}\\ &= 2d'\mathbb{E}_{(\widehat{\mathcal{G}}\widetilde{x})_1,(\widehat{\mathcal{G}}\widetilde{y})_2}(\Pr[E_{\mathcal{G}'}]^{d'-1}) + \mathcal{O}(md^{-\ln(d)+2})\end{aligned}$$

Finally, plugging this into (4.19), we get

$$\begin{aligned}\Pr[h_D(x) = h_D(y)] &= \Pr[h'_D(x) = h'_D(y)] + \mathcal{O}(md^{-\ln(d)+2})\\ &= \Pr[h'_D(x) = h'_D(y)] + \mathcal{O}(d^{-\ln(d)+3}).\end{aligned}$$

Now, we know that by Theorem 27,

$$\ln(\Pr[h_D(x) = h_D(y)]) = -\frac{\widetilde{R}^2}{4-\widetilde{R}^2}\ln(d') + \mathcal{O}_{\widetilde{R}}(\ln(\ln d')),$$

so provided d is large enough that $\ln(d) - 2 > \frac{\widetilde{R}^2}{4-\widetilde{R}^2}$, the lemma follows.

4.6 LSH Numerics

To illustrate our theoretical results in the low dimensional case, we ran Monte Carlo simulations to compare the collision probabilities for regular cross-polytope LSH as well as the fast and discrete versions for various values of the original and lifted dimension. We refer to [7] for an in depth comparison of run times for cross-polytope LSH and other popular hashing schemes.

The experiments were run with $N = 20000$ trials. The discretized scheme used 10 bits of randomness for each entry. The fast, discrete, and regular cross-polytope LSH schemes exhibit similar collision probabilities for small distances, with fast/discrete cross-polytope having marginally higher collision probabilities for larger distances. It is clear that as the lifted dimension decreases, the fast and discrete versions have higher collision probabilities at further distances, which decreases the sensitivity of those schemes.

The following figures illustrate the rate of convergence to the optimal collision probability as $d \to \infty$, as well as various lines that illustrate the optimal rate of convergence $C/\ln(d)$, where C varies for illustrative purposes. The experiments were run with varying distances and clearly show the same rate of convergence for the collision probability between the standard and fast cross-polytope schemes. We note that at low dimensions, the schemes behave even more similarly because the embedded dimension is much closer to the original dimension in this case.

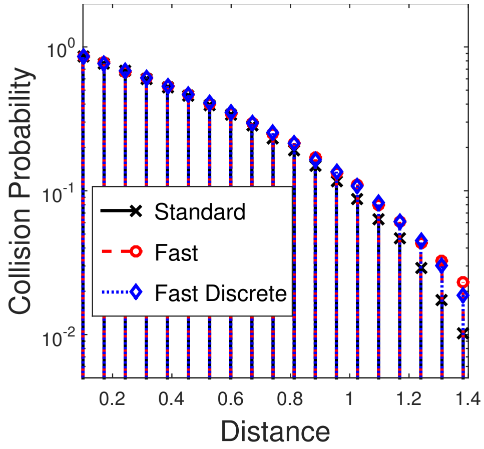

Figure 4.1: LSH collision probabilities, $d = 128$, $d' = 128$

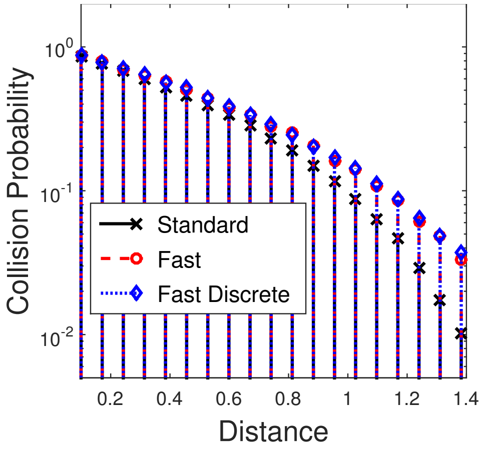

Figure 4.2: LSH collision probabilities, $d = 128$, $d' = 64$

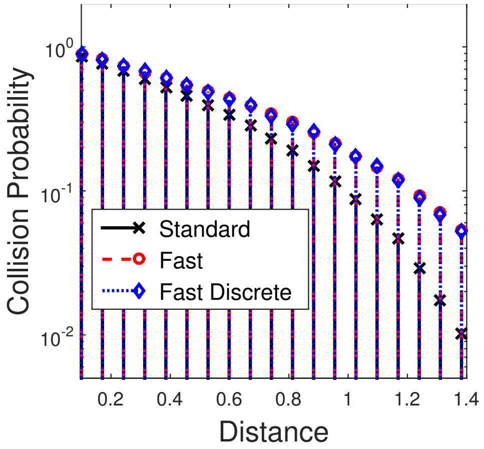

Figure 4.3: LSH collision probabilities, $d = 128$, $d' = 32$

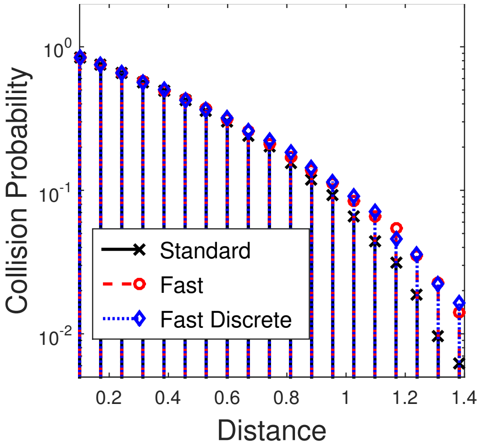

Figure 4.4: LSH collision probabilities, $d = 256$, $d' = 256$

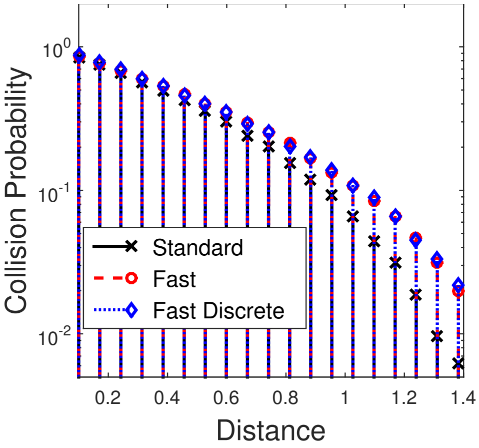

Figure 4.5: LSH collision probabilities, $d = 256$, $d' = 128$

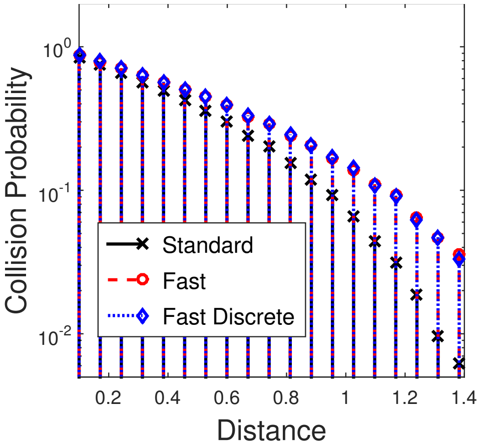

Figure 4.6: LSH collision probabilities, $d = 256$, $d' = 64$

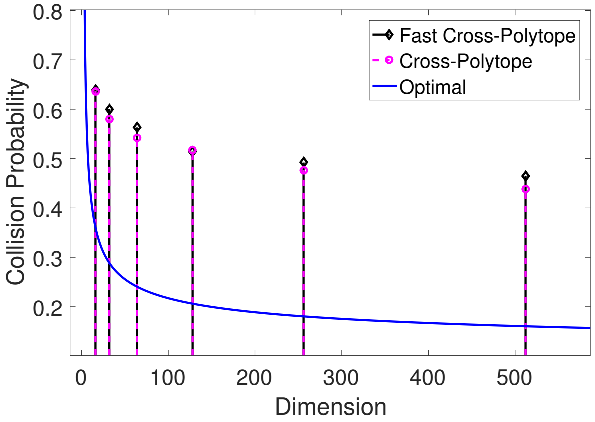

Figure 4.7: LSH collision probabilities by dimension, $R = 0.4$

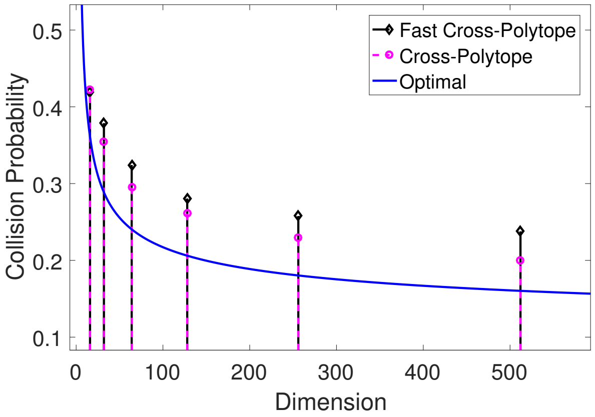

Figure 4.8: LSH collision probabilities by dimension, $R = 0.7$

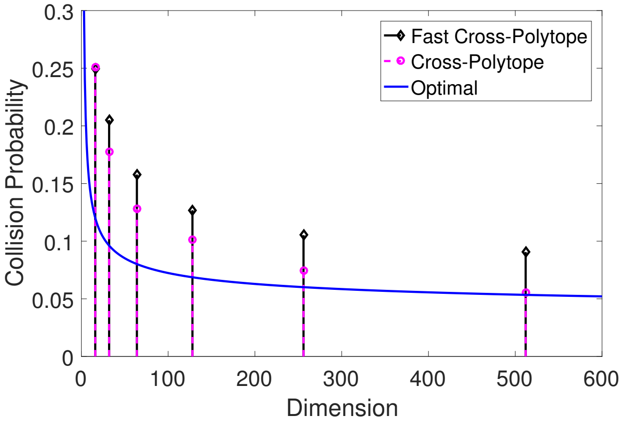

Figure 4.9: LSH collision probabilities by dimension, $R = 1$

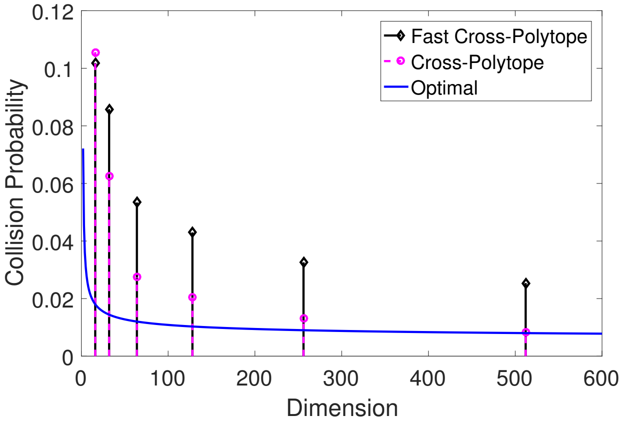

Figure 4.10: LSH collision probabilities by dimension, $R = 1.3$

Chapter 5

Distributional Robustness of Quantization Error

Recall our setting: we have a partition (a.k.a. quantization scheme) $\mathcal{Q} := \{(P_i, w_i)\}_{i=1}^{W} \subset \mathbb{R}^d \times \mathbb{R}^d$ adapted to a probability distribution $\mathbb{P}_1$ over $\mathbb{R}^d$. For this section, $\mathcal{Q}$ will be fixed. Here, $\{P_i\}_i$ is the Voronoi partition with centroids $\{w_i\}_i$ and "adapted" means that the quantization error,

$$E_{\mathbb{P}_1,\mathcal{Q}} := \int_{\mathbb{R}^d} \ell(x, \operatorname{argmin}_{\{w_i\}_i} \|x - w_i\|_2) d\mathbb{P}_1(x), \tag{5.1}$$

is small. ℓ is some loss function that we fix to be squared euclidean distance $\ell(x, y) = \|x - y\|_2^2$. Since $\mathcal{Q}$ is fixed, in the sequel we shorten the above notation to $E_{\mathbb{P}_1}$. If we assume the distribution $\mathbb{P}_1$ admits a probability density function $p(x) : \mathbb{R}^d \to \mathbb{R}$, then we can rewrite

$$E_{\mathbb{P}_1} = \sum_{i=1}^{W} \int_{P_i} p(x) \|x - w_i\|_2^2 dx. \tag{5.2}$$

We now receive samples $\{y_1, ..., y_n\} \sim \mathbb{P}_2$, and want to know the appropriate notion of distance between the distributions $\mathbb{P}_1$ and $\mathbb{P}_2$ such that the quantization error of $\mathbb{P}_2$, using the scheme $\{(P_i, w_i)\}_i$, is small. It turns out that there is a natural notion of distributional distance which guarantees this, the

Wasserstein distance (for some cost function $c : \mathbb{P}_1 \times \mathbb{P}_2 \to \mathbb{R}$),

$$W_c(\mathbb{P}_1, \mathbb{P}_2) := \inf_{M \in \Pi(\mathbb{P}_1, \mathbb{P}_2)} \mathbb{E}_M \left[c(\mathbb{P}_1, \mathbb{P}_2) \right]. \tag{5.3}$$

This distance computes the minimal expected cost between $\mathbb{P}_1$ and $\mathbb{P}_2$ among all join probability distributions $M \in \Pi(\mathbb{P}_1, \mathbb{P}_2)$ with marginals are $\mathbb{P}_1$ and $\mathbb{P}_2$. For this reason, it is often called the optimal transport distance where M is the transport map. With $c(x, y) = \|x - y\|_2^2$, the following proposition is immediate.

Proposition 36.

$$\sup_{\{\mathbb{P}_2 : W_d(\mathbb{P}_1, \mathbb{P}_2) \leq \rho\}} E_{\mathbb{P}_2} \leq E_{\mathbb{P}_1} + \rho. \tag{5.4}$$

Proof. Follows immediately from duality [65],
$W_d(\mathbb{P}, \mathbb{Q}) = \sup_{f : \mathrm{Lip}_1(f) \leq 1} \int f d\mathbb{P} - \int f d\mathbb{Q}$. □

Proposition 36 tells us Wasserstein distance is precisely what we need to achieve bounds on quantization error - so why doesn't the story end here? It turns out Wasserstein distance is insufficient for detecting local properties of our quantization scheme. For this purpose, we would like bounds of the form

$$\mathbb{P}_1(E)^{\alpha_0} C_0 \leq \mathbb{P}_2(E) \leq \mathbb{P}_1(E)^{\alpha_1} C_1, \tag{5.5}$$

simultaneously for all events E. The following proposition shows that Wasserstein distance is insufficient for getting scale invariant bounds.

Proposition 37. *For any $N > 0$ and some fixed constant $C > 0$, there are distributions $\mathbb{P}_1$ and $\mathbb{P}_2$, and an event E (all depending on $N > 0$), such that $W_d(\mathbb{P}_1, \mathbb{P}_2) \leq 1/N$ and $|\mathbb{P}_1(E) - \mathbb{P}_2(E)| \geq C$.*

In particular, this shows that no matter how small the Wasserstein distance is, the probability of some event under $\mathbb{P}_1$ can be a fixed amount from it's probability in $\mathbb{P}_2$.

Before we prove this counterexample which will be constructed using discrete distributions, we characterize a formulation of Wasserstein distance between two discrete distributions. Suppose that $\mathbb{P}_1$ is supported on $\{p_i\}_{i=1}^m$ and $\mathbb{P}_2$ is supported on $\{q_i\}_{i=1}^n$. Then, a probability measure on the product space $\mathbb{P}_1 \times \mathbb{P}_2$ with marginals $\mathbb{P}_1$ and $\mathbb{P}_2$ is precisely a set of indices $\{\lambda_{i,j}\}$ such that the marginal sums $\sum_{i=1}^m \lambda_{i,j} = q_j$ for all j and $\sum_{j=1}^n \lambda_{i,j} = p_i$ for all i. Defining the set

$$\mathcal{C} := \{\lambda \in \mathbb{R}^{m \times n} : \sum_{i=1}^m \lambda_{i,j} = q_j, \sum_{j=1}^n \lambda_{i,j} = p_i, \lambda_{i,j} \geq 0 \, \forall i, j\},$$

it follows that we can write the Wasserstein distance as the following linear program:

$$W_d(\mathbb{P}_1, \mathbb{P}_2) = \min_{\lambda \in \mathcal{C}} \sum_{i=1}^m \sum_{j=1}^n \lambda_{i,j} d(p_i, q_j). \tag{5.6}$$

Remark 38. *Although the above formulation is for finite discrete distributions, it extends naturally to discrete distributions with countably infinite support, where all the sums become infinite sums and the optimization is over an infinite dimensional space.*

Note that in the set $\mathcal{C}$, the condition that the marginals are $\mathbb{P}_1$ and $\mathbb{P}_2$ ensures that the product distribution λ is in fact a probability distribution. The above formulation has the advantage that is is computationally tractable using standard LP solvers, although solving a full LP in the product space is highly inefficient for computing distances. This allows us to formulate the simple counterexample given by Proposition 37.

Proof. (Proposition 37)
For simplicity assume $d = 1$ (the argument extends trivially to higher dimensions). Let $\mathbb{P}_1$ be the dirac distribution at 0, and let $\mathbb{P}_2$ be such that $\mathbb{P}_2(0) = 1 - C$, and $\mathbb{P}_2(i_N) = C$ for some i_n to be chosen later. Note first that, since $\mathbb{P}_1$ is supported at the origin, using (5.6), the marginal conditions force $\lambda_i = \mathbb{P}_2(i)$ for all $i > 0$, thus

$$\begin{aligned} W_d(\mathbb{P}_1, \mathbb{P}_2) &= \sum_{i=1}^{\infty} \mathbb{P}_2(i) i \\ &= C i_N. \end{aligned}$$

By construction, $\mathbb{P}_1(\{0\}) = \mathbb{P}_2(\{0\}) = C$, and choosing $i = \frac{1}{CN}$ we are done. □

Remark 39. *As the above counterexample suggests, Wasserstein distance will measure distance between probability distributions supported on disjoint low-dimensional manifolds, which necessarily precludes scale invariant bounds. For a continuous example, if $\mathbb{P}_1$ is the uniform distribution on $\{0\} \times [0, 1]$ and $\mathbb{P}_2$ is the uniform distribution on $\{\theta\} \times [0, 1]$, then $W_d(\mathbb{P}_1, \mathbb{P}_2) = \theta$, whereas for scale-invariant bounds this should necessarily be 0.*

5.1 Rényi Divergences

Comparing a quantization scheme fitted to distribution $\mathbb{P}_1$ to incoming samples $\mathbb{P}_2$ suggest using the Kullback-Leibner(KL) divergence, a standard metric for comparing probability distributions. Suppose that $\mathbb{P}_1$ is absolutely continuous with respect to $\mathbb{P}_2$ (so that the Radon-Nikodym derivative $\frac{d\mathbb{P}_1}{d\mathbb{P}_2}$ exists). Then, the KL-divergence from $\mathbb{P}_2$ to $\mathbb{P}_1$ is defined as

$$D_{KL}(\mathbb{P}_1 \| \mathbb{P}_2) := \mathbb{E}_{\mathbb{P}_1}\left[\ln \frac{d\mathbb{P}_1}{d\mathbb{P}_2}\right]. \tag{5.7}$$

For discrete distributions over the integers, this simplifies to $D_{KL}(\mathbb{P}_1 \| \mathbb{P}_2) = \sum_i \mathbb{P}_1(i) \ln \frac{\mathbb{P}_1(i)}{\mathbb{P}_2(i)}$.

It should be noted that the fact that the KL-divergence exists (from $\mathbb{P}_1$ to $\mathbb{P}_2$ and $\mathbb{P}_2$ to $\mathbb{P}_1$) implies that the distributions $\mathbb{P}_1$ and $\mathbb{P}_2$ have the same support. Although this condition is stringent, it is also necessary to achieve bounds of the form (5.5) over all events. However, even the KL-divergence is still insufficient.

Proposition 40.

Instead, we use a notion of distance called the α-Rényi divergence, for $\alpha > 1$,

$$R_\alpha(\mathbb{P}_1 \| \mathbb{P}_2) := \frac{1}{\alpha - 1} \ln\left(\mathbb{E}_{\mathbb{P}_2} \frac{d\mathbb{P}_1}{d\mathbb{P}_2}^{\alpha}\right). \tag{5.8}$$

Note that this converges to the KL-divergence as $\alpha \downarrow 1$. We can now prove scale-invariant bounds about our probability distributions.

Proposition 41. *Suppose $\mathbb{P}_1$ and $\mathbb{P}_2$ are probability distributions such that $\mathbb{P}_1$ is absolutely continuous w.r.t. $\mathbb{P}_2$ and $\mathbb{P}_2$ is absolutely continuous w.r.t. $\mathbb{P}_1$. Then, for all events E,*

$$\begin{aligned}&\mathbb{P}_2(E)^{(\alpha-1)/\alpha}\exp[-(\alpha-1)R_\alpha(\mathbb{P}_2\|\mathbb{P}_1)] \leq \mathbb{P}_1(E)\\ &\leq \mathbb{P}_2(E)^{\alpha/(\alpha-1)}\exp[(\alpha-1)R_\alpha(\mathbb{P}_1\|\mathbb{P}_2)].\end{aligned} \tag{5.9}$$

This immediately implies the following.

Corollary 42. *Suppose $\mathbb{P}_1$ and $\mathbb{P}_2$ are as in proposition 41. Then for any quantization scheme $\mathcal{Q}$,*

$$\exp[-(\alpha-1)R_\alpha(\mathbb{P}_2\|\mathbb{P}_1)]E_{\mathbb{P}_2} \leq E_{\mathbb{P}_1} \leq \exp[(\alpha-1)R_\alpha(\mathbb{P}_1\|\mathbb{P}_2)]E_{\mathbb{P}_2} \tag{5.10}$$

Proof. (Proposition 41 + Corollary 42)

The proposition follows from the following simple computation.

$$\begin{aligned}\mathbb{P}_1(E) &= \mathbb{E}_{\mathbb{P}_2}\left[1_E\frac{d\mathbb{P}_1}{d\mathbb{P}_2}\right] \quad \text{definition of Radon-Nikodym derivative}\\ &\leq \mathbb{P}_2(E)^{\alpha/(\alpha-1)}\left(\mathbb{E}_{\mathbb{P}_2}\frac{d\mathbb{P}_1}{d\mathbb{P}_2}\right)^\alpha \quad \text{Holder's Inequality}\\ &\leq \mathbb{P}_2(E)^{\alpha/(\alpha-1)}\mathbb{E}_{\mathbb{P}_2}\frac{d\mathbb{P}_1}{d\mathbb{P}_2}^\alpha \quad \text{Jensen's inequality}\\ &= \mathbb{P}_2(E)^{\alpha/(\alpha-1)}\exp[(\alpha-1)R_\alpha(\mathbb{P}_1\|\mathbb{P}_2)].\end{aligned}$$

Now, returning to the quantization error, in order to compare $E_{\mathbb{P}_2}$ and

$E_{\mathbb{P}_1}$ we fix *any* choice of centroids $\{w_i\}_i$,

$$
\begin{aligned}
&\sum_i \int_{P_i} \|x - w_i\|_2^2 d\mathbb{P}_1(x) = \int_{P_i} 2 \left(\int_0^{\|x-w_i\|_2} t dt \right) d\mathbb{P}_1(x) \\
&= \sum_i 2 \int_0^\infty t \left(\int_{P_i : t < \|x - w_i\|_2} d\mathbb{P}_1(x) \right) dt \quad \text{Fubini} \\
&= \sum_i 2 \int_0^\infty t\mathbb{P}_1(x \in P_i \cap \|x - w_i\|_2 > t)dt \\
&\leq \sum_i 2 \exp[(\alpha - 1)R_\alpha(\mathbb{P}_1\|\mathbb{P}_2) \int_0^\infty t\mathbb{P}_2(x \in P_k \cap \|x - w_i\|_2 > t)^{\alpha/(\alpha-1)}dt \\
&\leq \sum_i 2 \exp[(\alpha - 1)R_\alpha(\mathbb{P}_1\|\mathbb{P}_2)] \int_0^\infty t\mathbb{P}_2(x \in P_k \cap \|x - w_i\|_2 > t)dt \\
&= \exp[(\alpha - 1)R_\alpha(\mathbb{P}_1\|\mathbb{P}_2)] \sum_i \int_{P_i} \|x - w_i\|_2^2 d\mathbb{P}_2(x) \\
&= \exp[(\alpha - 1)R_\alpha(\mathbb{P}_1\|\mathbb{P}_2)] E_{\mathbb{P}_2}.
\end{aligned}
$$

The lower bound follows by symmetry. □

It follows that Rényi divergences measure both quantization error and scale-invariant probability bounds and are only slightly suboptimal to KL-divergence for the former (note that Corollary 42 approaches the bound for KL-divergence as $\alpha \downarrow 1$). The following table illustrates the similarities and differences between the different distributional distances we've presented so far.

Distance	Small Event Pr	Quantization Bounds
Wasserstein	None	$\sup_{\mathbb{P}_2 : W_d(\mathbb{P}_1, \mathbb{P}_2) \leq \rho} E_{\mathbb{P}_2} \leq E_{\mathbb{P}_1} + \rho$
KL Divergence	None	None
Rényi Divergence	(5.9)	(5.10)

5.2 Finite Sample Bounds

In order to make use of standard concentration inequalities, assume for this section that the distributions $\mathbb{P}_1$ and $\mathbb{P}_2$ have support bounded in a ball of radius $R > 0$. Fix a quantization scheme $\mathcal{Q} = \{(P_i, w_i)\}_{i=1}^W$ and after receiving N samples $q_i \sim \mathbb{P}_2$, define the empirical quantization error to be $\widehat{E}_{\mathbb{P}_2} := \sum_{i=1}^N \frac{1}{N} \|q_i - h(q_i)\|_2$, where $h(x) = \operatorname{argmin}_{w_i} \|x - w_i\|_2$. Note that the empirical quantization error has expectation equal to $E_{\mathbb{P}_2}$,

$$\mathbb{E}_{\mathbb{P}_2}\left[\widehat{E}_{\mathbb{P}_2}\right] = E_{\mathbb{P}_2}.$$

Therefore, using concentration inequalities we immediately have the following.

Proposition 43. *Suppose that $\mathbb{P}_1$ and $\mathbb{P}_2$ are mutually absolutely continuous, such that*

$R_\alpha(\mathbb{P}_1 \| \mathbb{P}_2), R_\alpha(\mathbb{P}_2 \| \mathbb{P}_1) \leq \delta$, for some $\alpha > 1$, $\delta > 0$. Suppose also that we have some fixed quantization scheme $\mathcal{Q}$ with error $E_{\mathbb{P}_1} < \epsilon$ on distribution $\mathbb{P}_1$ and $\epsilon > 0$ small, and we receive $\{q_i\}_{i=1}^N \sim \mathbb{P}_2$. Then,

$$\mathbb{P}_2(|\widehat{E_{\mathbb{P}_2}} - E_{\mathbb{P}_1}| \geq t + C_{\delta,\alpha}\epsilon) \leq 2\exp\left(\frac{-2Nt^2}{R^2}\right), \tag{5.11}$$

where $C_{\alpha,\delta} := \max\{|1 - \exp[(\alpha - 1)\delta]|, |1 - \exp[-(\alpha - 1)\delta]\}$.

Proof. Note that using the observation $\mathbb{E}_{\mathbb{P}_2}\left[\widehat{E}_{\mathbb{P}_2}\right] = E_{\mathbb{P}_2}$, we have by Hoeffding's inequality,

$$\mathbb{P}_2(|\widehat{E}_{\mathbb{P}_2} - E_{\mathbb{P}_2}| \geq t) \leq 2\exp\left(\frac{-2Nt^2}{R^2}\right).$$

Now, observe from proposition 42 that

$$\begin{aligned}|\widehat{E}_{\mathbb{P}_2} - E_{\mathbb{P}_1}| &\leq |\widehat{E}_{\mathbb{P}_2} - E_{\mathbb{P}_2}| + |E_{\mathbb{P}_2} - E_{\mathbb{P}_1}| \\ &\leq |\widehat{E}_{\mathbb{P}_2} - E_{\mathbb{P}_2}| + C_{\alpha,\delta} E_{\mathbb{P}_1},\end{aligned}$$

and the proposition follows. □

5.3 Future Work

Given our notion of Rényi divergence, which we showed is a good measure of how close two distributions are for the purpose of quantization, a natural question is how do we improve a quantization scheme $\mathcal{Q}$ adapted to $\mathbb{P}_1$, to work well for $\mathbb{P}_2$ given $R_\alpha(\mathbb{P}_1, \mathbb{P}_2) >> 0$ for all $\alpha \geq \alpha_0 > 0$ (note that $R_\alpha(\mathbb{P}_1, \mathbb{P}_2)$ is an increasing functions of α, for fixed $\mathbb{P}_1, \mathbb{P}_2$). We could also impose conditions on $\mathbb{P}_1$ and $\mathbb{P}_2$ so that we still achieve small probability bounds and quantization error but don't require such a strict distance metric.

Another consideration is that quantization error is not a universal indicator of quality for a quantization scheme. There are many other desirable properties:

- Uniform number of data points in each partition element.
- Ability to detect interesting (non-linear) partitions of the data.
- k-Nearest Neighbor recall.

The first condition roughly means that the probability of an point $p \sim \mathbb{P}_1$ lands in a given partition element P_i is roughly $1/W$ (recall W is the number of partition elements). This property is much more difficult to track even for a simple scheme like a Voronoi partition, and does not seem to admit a simple analysis.

The second condition is beyond the scope of Voronoi partitions, and as such we can longer exploit this strict structure. This class includes a large breadth of classification algorithms, and the quality is heavily dependent on the underlying (unknown) distribution of the data.

Finally, the third condition points to a particular application of quantization schemes, to hash nearby points to the same partition element (similar to LSH). This condition is closely related to small event probabilities, in particular near the boundaries of the partition elements.

Bibliography

[1] Nir Ailon and Bernard Chazelle. The fast Johnson-Lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39(1):302–322, May 2009.

[2] Nir Ailon and Edo Liberty. An almost optimal unrestricted fast johnson-lindenstrauss transform. *ACM Transactions on Algorithms (TALG)*, 9(3):21, 2013.

[3] Nir Ailon and Holger Rauhut. Fast and rip-optimal transforms. *Discrete & Computational Geometry*, 52(4):780–798, 2014.

[4] Noga Alon. Problems and results in extremal combinatorics. *Discrete Mathematics*, 273:31–53, 2003.

[5] Uri Alon, Naama Barkai, Daniel A Notterman, Kurt Gish, Suzanne Ybarra, Daniel Mack, and Arnold J Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750, 1999.

[6] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*,

FOCS '06, pages 459–468, Washington, DC, USA, 2006. IEEE Computer Society.

[7] Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. Practical and optimal lsh for angular distance. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 1225–1233, Cambridge, MA, USA, 2015. MIT Press.

[8] Alexandr Andoni, Piotr Indyk, Huy L Nguyen, and Ilya Razenshteyn. Beyond Locality-Sensitive Hashing. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1018–1028. SIAM, 2014.

[9] Alexandr Andoni and Ilya Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, STOC '15, pages 793–801, New York, NY, USA, 2015. ACM.

[10] Alexandr Andoni and Ilya Razenshteyn. Tight Lower Bounds for Data-Dependent Locality-Sensitive Hashing. *ArXiv e-prints*, July 2015.

[11] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[12] Anja Becker and Thijs Laarhoven. Efficient (ideal) lattice sieving using cross-polytope lsh. Cryptology ePrint Archive, Report 2015/823, 2015. `http://eprint.iacr.org/`.

[13] Alexandre Belloni, Victor Chernozhukov, and Lie Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.

[14] Peter Bickel, Ya'acov Ritov, and Alexandre B Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, pages 1705–1732, 2009.

[15] Emmanuel Candes and Mark A Davenport. How well can we estimate a sparse vector? *Applied and Computational Harmonic Analysis*, 34(2):317–323, 2013.

[16] Emmanuel Candes and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

[17] Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388. ACM, 2002.

[18] Sourav Chatterjee and Jafar Jafarov. Prediction error of cross-validated lasso. *arXiv preprint arXiv:1502.06291*, 2015.

[19] Anirban Dasgupta, Ravi Kumar, and Tamás Sarlós. Fast locality-sensitive hashing. In *Proceedings of the 17th ACM SIGKDD international confer-*

ence on Knowledge discovery and data mining, pages 1073–1081. ACM, 2011.

[20] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab Mirrokni. Locality sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual Symposium on Computational Geometry. New York*, pages 253–262, 2004.

[21] Lee Dicker. Variance estimation in high-dimensional linear models. *Biometrika*, 101(2):269–284, 2014.

[22] David L Donoho and Jain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3):425–455, 1994.

[23] Jianqing Fan, Shaojun Guo, and Ning Hao. Variance estimation using refitted cross-validation in ultrahigh dimensional regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(1):37–65, 2012.

[24] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[25] Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. In *Advances in Neural Information Processing Systems*, pages 3440–3448, 2016.

[26] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[27] Todd R Golub, Donna K Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P Mesirov, Hilary Coller, Mignon L Loh, James R Downing, Mark A Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537, 1999.

[28] Ishay Haviv and Oded Regev. The restricted isometry property of subsampled fourier matrices. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 288–297, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics.

[29] Darren Homrighausen and Daniel McDonald. The lasso, persistence, and cross-validation. In *International Conference on Machine Learning*, pages 1031–1039, 2013.

[30] Jan-Christian Hütter and Philippe Rigollet. Optimal rates for total variation denoising. In *Conference on Learning Theory*, pages 1115–1146, 2016.

[31] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirti-*

eth Annual ACM Symposium on Theory of Computing, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM.

[32] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26:189–206, 1984.

[33] Daniel M. Kane and Jelani Nelson. Sparser johnson-lindenstrauss transforms. *J. ACM*, 61(1):4:1–4:23, January 2014.

[34] Christopher Kennedy and Rachel Ward. Fast cross-polytope locality-sensitive hashing. *8th Innovations in Theoretical Computer Science*, 67(53):16, 2017.

[35] Christopher Kennedy and Rachel Ward. Greedy variance estimation for the lasso. *arXiv preprint arXiv:1803.10878*, 2018.

[36] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

[37] Felix Krahmer and Rachel Ward. New and improved Johnson-Lindenstrauss embeddings via the restricted isometry property. *SIAM Journal on Mathematical Analysis*, 43(3):1269–1281, 2011.

[38] Kasper Green Larsen and Jelani Nelson. The johnson-lindenstrauss lemma is optimal for linear dimensionality reduction. *arXiv preprint arXiv:1411.2404*, 2014.

[39] Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000.

[40] Jia Li and James Z Wang. Real-time computerized annotation of pictures. *IEEE transactions on pattern analysis and machine intelligence*, 30(6):985–1002, 2008.

[41] Yue Lin, Rong Jin, Deng Cai, Shuicheng Yan, and Xuelong Li. Compressed hashing. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 446–451, Washington, DC, USA, 2013. IEEE Computer Society.

[42] Ting Liu, Andrew W. Moore, Ke Yang, and Alexander G. Gray. An investigation of practical approximate nearest neighbor algorithms. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 825–832. MIT Press, 2005.

[43] Karim Lounici, Massimiliano Pontil, Sara Van De Geer, Alexandre B Tsybakov, et al. Oracle inequalities and optimal inference under group sparsity. *The Annals of Statistics*, 39(4):2164–2204, 2011.

[44] Enno Mammen, Sara van de Geer, et al. Locally adaptive regression splines. *The Annals of Statistics*, 25(1):387–413, 1997.

[45] Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics*, pages

246–270, 2009.

[46] Rajeev Motwani, Assaf Naor, and Rina Panigrahi. Lower bounds on Locality Sensitive Hashing. In *Proceedings of the Twenty-second Annual Symposium on Computational Geometry*, SCG '06, pages 154–157, New York, NY, USA, 2006. ACM.

[47] Deanna Needell and Rachel Ward. Near-optimal compressed sensing guarantees for total variation minimization. *IEEE transactions on image processing*, 22(10):3941–3949, 2013.

[48] Ryan O'Donnell, Yi Wu, and Yuan Zhou. Optimal lower bounds for Locality-Sensitive Hashing (except when Q is tiny). *ACM Trans. Comput. Theory*, 6(1):5:1–5:13, March 2014.

[49] Andrei Osipov. *A Randomized Approximate Nearest Neighbors Algorithm*. PhD thesis, Yale University, New Haven, CT, USA, 2011. AAI3467911.

[50] David Pollard. Quantization and the method of k-means. *IEEE Transactions on Information theory*, 28(2):199–205, 1982.

[51] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE transactions on information theory*, 57(10):6976–6994, 2011.

[52] Holger Rauhut. Compressive sensing and structured random matrices. *Theoretical foundations and numerical methods for sparse recovery*, 9:1–92, 2010.

[53] Stephen Reid, Robert Tibshirani, and Jerome Friedman. A study of error variance estimation in lasso regression. *Statistica Sinica*, pages 35–67, 2016.

[54] Alfréd Rényi et al. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961.

[55] Mark Rudelson and Roman Vershynin. On sparse reconstruction from fourier and gaussian measurements. *Communications on Pure and Applied Mathematics*, 61(8):1025–1045, 2008.

[56] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[57] Dinesh Singh, Phillip G Febbo, Kenneth Ross, Donald G Jackson, Judith Manola, Christine Ladd, Pablo Tamayo, Andrew A Renshaw, Anthony V D'Amico, Jerome P Richie, et al. Gene expression correlates of clinical prostate cancer behavior. *Cancer cell*, 1(2):203–209, 2002.

[58] Kengo Terasawa and Yuzuru Tanaka. Spherical LSH for approximate nearest neighbor search on unit hypersphere. In *Algorithms and Data Structures*, pages 27–38. Springer, 2007.

[59] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[60] Sara A Van de Geer. High-dimensional generalized linear models and the lasso. *The Annals of Statistics*, pages 614–645, 2008.

[61] Sara A Van De Geer, Peter Bühlmann, et al. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2009.

[62] Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[63] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

[64] Nicolas Verzelen et al. Minimax risks for sparse regressions: Ultra-high dimensional phenomenons. *Electronic Journal of Statistics*, 6:38–90, 2012.

[65] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[66] Martin J Wainwright. Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting. *IEEE Transactions on Information Theory*, 55(12):5728–5741, 2009.

[67] Yu-Xiang Wang, James Sharpnack, Alex Smola, and Ryan Tibshirani. Trend filtering on graphs. In *Artificial Intelligence and Statistics*, pages 1042–1050, 2015.

[68] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB '98, pages 194–205, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[69] Fei Ye and Cun-Hui Zhang. Rate minimaxity of the lasso and dantzig selector for the lq loss in lr balls. *Journal of Machine Learning Research*, 11(Dec):3519–3540, 2010.

[70] Jianbo Ye, Panruo Wu, James Z Wang, and Jia Li. Fast discrete distribution clustering using wasserstein barycenter with sparse support. *IEEE Transactions on Signal Processing*, 65(9):2317–2332, 2017.

[71] Cun-Hui Zhang and Jian Huang. The sparsity and bias of the lasso selection in high-dimensional linear regression. *The Annals of Statistics*, pages 1567–1594, 2008.

[72] Tong Zhang et al. Some sharp performance bounds for least squares regression with l1 regularization. *The Annals of Statistics*, 37(5A):2109–2144, 2009.

[73] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

.

Vita

Christopher Garrett Kennedy was born in Tucson, Arizona in 1991 and attended Catalina Foothills High School, where he graduated in 2009. He was interested in mathematics throughout high school and continued to pursue a degree in mathematics with a minor in computer science at Princeton University. Christopher graduated from Princeton in the spring of 2013, and immediately began attending graduate school at the University of Texas at Austin that fall.

Permanent address: 2314 Camino La Zorrela
Tucson, Arizona 85718

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.