BOUNDS FOR NEAREST NEIGHBOR ALGORITHMS AND EMBEDDINGS

by

Amirali Abdullah

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computing

School of Computing

The University of Utah

May 2016

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Amirali Abdullah**

has been approved by the following supervisory committee members:

Suresh Venkatasubramanian	, Chair	25th August, 2015 Date Approved
Alexandr Andoni	, Member	4th September, 2015 Date Approved
Feifei Li	, Member	31st August, 2015 Date Approved
Jeffrey Phillips	, Member	28th August, 2015 Date Approved
Robert Michael Kirby II	, Member	25th August, 2015 Date Approved

and by **Robert Michael Kirby II** , Chair of

the Department of **School of Computing**

and by **David B. Kieda** , Dean of The Graduate School.

ABSTRACT

In the last two decades, an increasingly large amount of data has become available. Massive collections of videos, astronomical observations, social networking posts, network routing information, mobile location history and so forth are examples of real world data requiring processing for applications ranging from classification to predictions. Computational resources grow at a far more constrained rate, and hence the need for efficient algorithms that scale well.

Over the past twenty high quality theoretical algorithms have been developed for two central problems: *nearest neighbor search* and *dimensionality reduction* over Euclidean distances in worst case distributions. These two tasks are interesting in their own right. Nearest neighbor corresponds to a database query lookup, while dimensionality reduction is a form of compression on massive data. Moreover, these are also subroutines in algorithms ranging from clustering to classification.

However, many highly relevant settings and distance measures have not received similar attention to that of worst case point sets in Euclidean space. The Bregman divergences include the information theoretic distances, such as entropy, of most relevance in many machine learning applications and yet prior to this dissertation lacked efficient dimensionality reductions, nearest neighbor algorithms, or even lower bounds on what could be possible. Furthermore, even in the Euclidean setting, theoretical algorithms do not leverage that almost all real world datasets have significant low-dimensional substructure.

In this dissertation, we explore different models and techniques for similarity search and dimensionality reduction. What upper bounds can be obtained for nearest neighbors for Bregman divergences? What upper bounds can be achieved for dimensionality reduction for information theoretic measures? Are these problems indeed intrinsically of harder computational complexity than in the Euclidean setting? Can we improve the state of the art nearest neighbor algorithms for real world datasets in Euclidean space? These are the questions we investigate in this dissertation, and that we shed some new insight on.

In the first part of our dissertation, we focus on Bregman divergences. We exhibit nearest neighbor algorithms, contingent on a distributional constraint μ on the datasets. We next show lower bounds suggesting that μ is in some sense inherent to the problem complexity. After this we explore dimensionality reduction techniques for the Jensen-Shannon and Hellinger distances, two popular information theoretic measures.

In the second part, we show that even for the more well-studied Euclidean case, worst case nearest neighbor algorithms can be improved upon sharply for real world datasets with spectral structure.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

After five years, it is my deepest pleasure to thank all those who helped this dream become a reality.

A beginning Ph.D. student meeting his advisor may feel as a journeyman mage would in Merlin's presence, pondering how to match the legend summoning celestial dragons down with his spells when all he can conjure up himself are weak puffs of smoke. Suresh Venkatasubramanian was to show me that research is not magic, but rather science buttressed by healthy doses of enthusiasm and caffeine. My words cannot adequately convey the value of his astute technical advice, his belief in my capabilities, and his seemingly supernatural grasp of just when he should be patient and when to push me harder in my work. But most of all, my deepest gratitude goes to Suresh for reminding me that first and foremost, research can be fun and inspirational, and that few pleasures compare to an afternoon of coffee, whiteboard scribbling and brainstorming.

I have been priveleged to work with many senior co-authors, all of whom have offered invaluable lessons, both in terms of technical insights as well as the oppportunity to observe exemplary work practices. Alexandr Andoni, Robert Krauthgamer, Ravindran Kannan, Jeff Phillips, Andrew Mcgregor, Ravi Kumar and Sergei Vassilvitskii were all wonderful collaborators that any graduate student would love to work with. I especially thank Alexandr Andoni for his willingness to mentor and advise me closely over many meetings in San Fransisco coffee shops and the offices of MSR Silicon Valley, and to give invaluable advice on best practices for a young researcher. Nabil Mustafa I am indebted to for introducing me to theoretical computer science, and the joys of mulling over a combinatorial or mathematical puzzle in the wee hours of morning.

Beyond my interaction with senior researchers, surviving through my Ph.D. was only possible due to a great deal of support from enthusiastic lab mates both in research and life. In this regard, I particularly want to mention Parasaran Raman for giving me a couch to crash on my first few weeks in the States, Samira Daruki for multiple collaborations and some lovely shared life experiences in Brazil and Berkeley, John Moeller for ever wry company and extensive writing help, Avishek Saha for providing me a senior's words of wisdom, and Mina Ghashami for wielding an inspirational work ethic.

On the personal front, this space cannot suffice to credit all who have been by my side in this process. Abdul Basit has been a constant companion and support since we both entered algorithm research six years ago back in Lahore, Pakistan. I have long lost count of the stories we have exchanged over graduate school frustrations, and idle philsophical musings. Ali Yost has always provided a shoulder to lean on, a penpal to correspond with, a home to celebrate Thanksgiving and Christmas in and a fellow bibliophile with whom to exchange books and ideas. As my comrades in being long standing Ph.D. students, Raheel Samuel and Abhijit Mondal always grounded my perspective that this Ph.D. journey will eventually reach its conclusion. Before I even started my research, Ammar Zafar believed in and encouraged me, as did my captain "O-cap" Bilal Iqbal and my friend Moonis Hashmi helped me make the transition between continents. To mention just a few of the multitude of friends who supported and encouraged me - Tehmina Pirzada, Sehrish Javaid, Mudassir Shabbir, Ankit Agarwal, Rahma Jawad, Hitesh Tolani and Darron Brown.

And finally the foundation and core of my life has always been my family. I wish to thank my dear Magda for placing my dream above all else in her life and little Ruckus for bringing mirth, kisses and a baby's joyous bawl to my home. I end on a special note of appreciation to my mother Gulshan for instilling in me a love of learning, along with my favorite niece Natasha, my aunt Almas, my sisters Saira and Fatima and my adoptive parents Yasmin and Iqbal for always being here for me.

CHAPTER 1

INTRODUCTION

The nearest neighbor problem is one of the most extensively studied problems in data analysis. The past 20 years has seen tremendous research into the problem of computing near neighbors efficiently as well as approximately in different kinds of metric spaces.

An important application of the nearest-neighbor problem is in querying content databases (images, text, and audio databases, for example). In these applications, the notion of similarity is based on a distance metric that arises from information-theoretic or other considerations. Popular examples include the Kullback-Leibler divergence (53), the Itakura-Saito distance (71) and the Mahalanobis distance (104). These distance measures are examples of a general class of divergences called the *Bregman divergences* (37), and this class has received much attention in the realm of machine learning, computer vision and other application domains.

Let us give a potential use case of nearest neighbor search. Suppose one had a pretty Siamese cat Paddy, that was beloved to all in the family, with soft paws and curved whiskers. After a happy life, the cat passes away peacefully at night, and the children wish to buy a similar looking feline. So of course, the parent searches for matches to pictures of a young Paddy from a database maintained by local pet shops – essentially a nearest neighbor search in the space of kitten images. More immediately to machine learning applications, consider the voting k nearest neighbor (k-NN) classification rule broadly used in machine learning (60). Given a large labelled dataset (say of images) and an appropriate distance measure, one may assign a label to any unknown query point q corresponding to the most frequent in the k-NN to q. Cover and Hart (52) show that as the number of neighbors k and point set size n approach infinity, in a manner such that $k/n \to 0$, this rule converges to the optimal Bayes error rate. The only expensive algorithmic primitive required here is the quick computation of the k-NN to any incoming data point.

As may have been implicit in the examples, a large component of such problems is the *representation* of the data (typically as points in $\mathbb{R}^d$) such that the distance measure of choice (say Euclidean distance) captures well the real world notion of similarity we care about. We do not consider the problems of learning or constructing such features over the course of this dissertation, but one major motivation for our work is that not all cases can be shoehorned into the Euclidean setting. One natural example is to consider a distance measure between probability distributions, say perhaps representing frequencies of keywords in documents. Here the distances which are most suitable are information theoretic measures such as the Hellinger, Jensen-Shannon or Kullback-Leibler, and it is precisely for high quality approximations in these settings for which we develop algorithms and embedding techniques. We continue now to define the basic problem and setting.

1.1 Problem statement and prior work

In the nearest neighbor (NN) problem, we are given a set P of n points in a d-dimensional space and the goal is building a data structure that reports a nearest neighbor to a query point q. Here clearly the space storage is at least $\Omega(nd)$ and one trivial approach is to simply perform a linear scan in $O(nd)$ time to return the nearest neighbor to q. In the Euclidean setting, another approach is to partition space so that each cell corresonds to the "nearest neighbor" region for a point. Once this partitioning is done, q may be assigned to the correct cell, for example by means of a binary tree. The space requirement of such a partitioning unfortunately may be very high, on the order of $O(n^{\lceil \frac{d}{2} \rceil})$. (See the paper by Clarkson on nearest neighbor using Voronoi diagrams (46), or the survey by Aurenhammer (24).)

In most applications, an *approximate* nearest neighbor (ANN) suffices. This is a point within a $(1+\varepsilon)$ multiplicative factor of the distance to the exact nearest neighbor, where ε is a parameter controlling the approximation quality. See for example Figure 1.1. The structure of the ANN problem varies greatly over the choice of distance measure. Besides the ubiquitous Euclidean distance (ℓ_2^2), other measures of interest include the ℓ_p distances (58, 115), the Frechet distance (83), variants of the edit distance (15) and metrics with certain restrictions (47, 79, 95). A broad range of techniques have been devised for ANN in various spaces. We do not give a comprehensive survey here, but we will touch upon three powerful tools that recur in the literature and which we draw upon in our work.

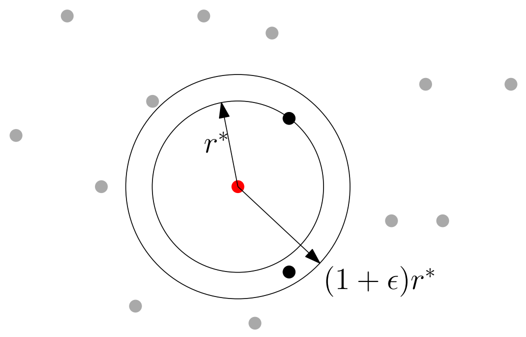


Figure 1.1. Illustration of approximate nearest neighbor

1.1.1 Recursive subdivision of space

In Euclidean spaces of fixed dimension, Arya *et al.* (23) gave the first algorithms for $(1+\varepsilon)$-ANN, with dependence exponential in dimension, logarithmic in the number of points n, ε specified at query time and with linear space. Their structure is a hierarchical subdivision of the domain by a balanced box-decomposition (BBD) tree, such that given a query point, the algorithm performs a restricted BFS for an approximate nearest neighbor. Key technical aspects of this work and subsequent research involves bounding the depth of the tree to be logarithmic in n, cutting off the search without necessarily expanding to the leaves, and limiting the number of cells explored at each level. See Figure 1.2 for an illustration of how the parameters control the algorithmic complexity.

Har-Peled and Mendel (79) showed this idea could be extended to low-dimensional metrics, whereas a number of works have presented hierachical trees that depend on some notion of low-dimensionality in the ambient space (32, 49, 90, 94, 95, 152). We refer to the book by Har-Peled (139) for an excellent exposition of low-dimensional algorithms for ANN in the Euclidean case. These ideas find utilization in our dissertation both for algorithms for low-dimensional ANN for the Bregman divergences, as well as for devising a new notion of low "spectral" dimension for ANN in the ℓ_2 setting.

1.1.2 Embedding, sketching and dimensionality reduction

A **sketch** for a set of points with respect to a property P is a function that maps the data to a small summary from which property P can be evaluated, albeit with some approximation error. Linear sketches are especially useful for estimating a derived property of a data stream in a fast and compact way. Sketches are by now well studied in Euclidean

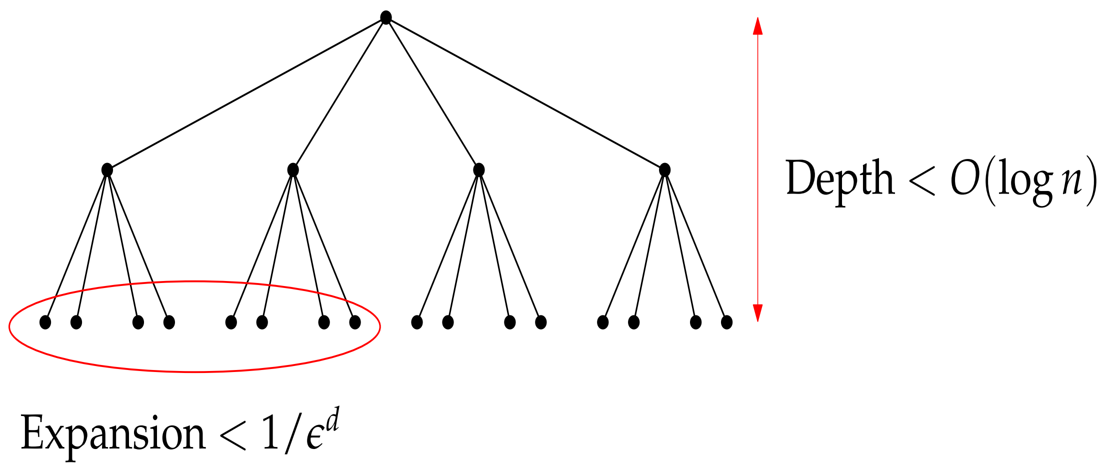


Figure 1.2. Quadtree parameters

and Hamming spaces (11, 84), as well as in graphs (7, 44) in which domain all streaming turnstile algorithms are known to be equivalent to a sketch (102).

Complementing sketching, **embedding** techniques are one to one mappings that transform a collection of points lying in one space X to another (presumably easier) space Y, while approximately preserving distances between points. In particular, a low distortion (D) embedding $f : X \to Y$ satisfies that for appropriate choice of scaling constant t we have that

$$\forall x, y \in X, d_Y(f(x_1), f(x_2)) \leq t d_X(x_1, x_2) \leq D d_y(f(x_1), f(x_2)).$$

Among well known results in a vast literature are embeddings of arbitrary metrics into tree metrics (29, 65), and average distortion embeddings (100, 133) as well as nearest neighbor preserving embeddings (85). Andoni, Krauthgamer and Razenshteyn (18) show that sketching and embedding into $\ell_{1-\varepsilon}$ are essentially equivalent for normed spaces.

Dimensionality reduction is a special kind of embedding which preserves the structure of the space, while reducing its dimension, i.e., mapping from a space X_d to X'_d for $d' \ll d$. These embedding techniques can be used in an almost "plug-and-play" fashion to speed up many algorithms in data analysis: for example for near neighbor search (and classification), clustering, and closest pair calculations. In particular, the Johnson-Lindenstrauss lemma (88) states that any set of n points in Euclidean space can be embedded into $O(\log n)$ dimensions with constant distortion, and has found wide application in applications from clustering (36, 48), to classification (67) and nearest neighbor (9, 14) search. The Johnson-

Lindenstrauss lemma is based on random projection, and Chapter 5 studies whether for certain point sets and models in ℓ_2, a "data-aware" projection may provide superior bounds.

1.1.3 Other notes and results

For spaces that are truly high-dimensional (i.e, the point set does not lie on a low-dimensional manifold or other corresponding notion of small intrinsic dimension), there is a whole line of work based on *Locality Sensitive Hashing*, both in the data dependent and independent sense. Indyk and Motwani first give a hashing scheme for high-dimensional data in the Hamming cube (84) based on a reduction to the problem of point location in equal balls, and selecting nonzero bits of binary vectors. This was improved in later work for ℓ_2^2 by Indyk and Andoni (14) to an optimal hashing scheme for data oblivious methods. However by allowing the hash functions to depend on the dataset, Andoni, Indyk, Nguyen and Razenshteyn (16) give a further improved algorithm, with Andoni and Razenshteyn (19) obtaining near optimal results. With great oversimplification, their scheme may be described as recursively clustering a point set and then noting that improved (oblivious) hashing schemes exist on the low-diameter families of points so induced. We provide here a summary of known results, Table 1.1, for ℓ_2 to compare against as the aim for our later bounds in this dissertation both for the Bregman divergences, and the Euclidean case where the data has low intrinsic dimension.

1.2 Bregman divergences

Bregman divergences are important distance measures that are used extensively in data-driven applications such as computer vision, text mining, and speech processing, and are a key focus of interest in machine learning. They were first introduced by Bregman (37) and are the unique divergences that satisfy certain axiom systems for distance measures (55), as well as being key players in the theory of information geometry (12).

For example, as discussed in (81), the KL-divergence (also known as relative entropy) between probability distributions, has found a multitude of applications such as determining how confusable two Hidden Markov Models are (141), finding a best match between histogram image models (70) or assigning important classes of geospatial distributions on Instagram images (147). The Jensen-Shannon divergence, which is linked to the concept of information gain, has found applications in coding theory (114), theoretical physics (108), or for computing distances between graphs (26). More generally, the important learning

Table 1.1. Results for Euclidean case[a]

Approximation	Space	Query time
Exact	$O(nd)$	$O(nd)$
Exact	$O(n^{\lceil \frac{d}{2} \rceil})$	$O(\log n)$
$1+\varepsilon$	$O(nd)$	$O\left(\frac{1}{\varepsilon^{O(d)}} \log n\right)$
c	$O(dn + n^{1+\rho})$	$O(dn^{\rho})$

[a] $\rho = 1/(2c^2-1)$

problems of boosting and logistic regression can be cast as an optimization of Bregman distances (51) and unify different mixture-model density estimation problems (27).

We now formally define the Bregman divergences. Let $\phi : M \subset \mathbb{R}^d \to \mathbb{R}$ be a *strictly convex* function that is differentiable in the relative interior of M. Strict convexity implies that the second derivative is never 0 and will be a convenient technical assumption. The *Bregman divergence* D_ϕ is defined as

$$D_\phi(x,y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x-y \rangle. \tag{1.1}$$

More geometrically, the Bregman divergence $D_\phi(x,y)$ can be viewed as the difference of $f(x)$ from that estimated by taking the linearization of f at y. In general, D_ϕ is asymmetric. We note too that *symmetrized* Bregman divergence can be defined by averaging

$$D_{s\phi}(x,y) = \frac{1}{2}(D_\phi(x,y) + D_\phi(y,x)) = \frac{1}{2}\langle x-y, \nabla\phi(x) - \nabla\phi(y) \rangle. \tag{1.2}$$

An important subclass of Bregman divergences are the *decomposable* Bregman divergences. Suppose ϕ has domain $M = \prod_{i=1}^{d} M_i$ and can be written as $\phi(x) = \sum_{i=1}^{d} \phi_i(x_i)$, where $\phi_i : M_i \subset \mathbb{R} \to \mathbb{R}$ is also strictly convex and differentiable in relint(S_i). Then $D_\phi(x,y) = \sum_{i=1}^{d} D_{\phi_i}(x_i, y_i)$ is a *decomposable* Bregman divergence. The majority of commonly used Bregman divergences are decomposable; (40, Chapter 3) illustrates some of the commonly used ones, including the Euclidean distance, the KL-divergence, and the Itakura-Saito distance.

A first study of the algorithmic geometry of Bregman divergences was performed by Boissonnat, Nielsen, and Nock (34). They observed since Bregman divergences retain the same combinatorial structures as ℓ_2, many exact algorithms from the Euclidean domain carry over naturally with the same bounds. For example, they showed that exact near neighbors can be computed in $O(n^{\lfloor \frac{d}{2} \rfloor})$ via a Voronoi diagram. Nielsen and Nock also observed that the smallest enclosing disk can be computed exactly in polynomial time (122). These parallels do *not* carry over to the approximate setting with the lack of a triangle inequality and symmetry rendering most tools for algorithm design useless. The algorithms that do exist attempt a work around via a structure constant μ. This constant is at least 1, and grows larger as the space becomes increasingly nonmetric. There are many algorithms for clustering whose resources are parametrized by μ: Mänthey and Roglin (106) compute approximate k-means with an extra μ^6 factor under a certain perturbation model. Ackermann and Blömer (3) exhibit a $O\left(\mu^2 \log k\right)$-approximate solution to k-means clustering via a k-means++-like procedure. The same authors give a $O(\mu)$ approximate k-median clustering for a certain class of well behaved input instances (2). McGregor and Chaudhuri (43) avoid dependence on μ in an approximate algorithm for k-means clustering under the KL-divergence, but at the cost of a $\log(n)$ factor in approximation. They also show k-means is NP hard to approximate within a constant factor if the centers are restricted to be from the point set and implicitly leverage the nonmetric nature of the space in their bound. There are numerous heuristic algorithms for computing with Bregman divergences approximately, including algorithms for the minimum enclosing ball (117) and near neighbor search (40, 159), but prior to the work in this dissertation, there has not been extensive study of what algorithmic guarantees can be obtained.

1.3 Thesis statement

This dissertation explores nearest neighbor algorithms and embeddings over certain more specialized settings. We will discuss upper bounds for approximate nearest neighbor search for general Bregman divergences. We will then present lower bounds that conversely show ANN search is hard in high dimensions due to the asymmetry of these spaces.

We will consider next a subclass of metric information distances which do not suffer from the "curse of asymmetry," including the popular Jensen-Shannon divergence, and demonstrate that they admit efficient dimensionality reduction on the simplex and succinct sketches. Our final result given here is devising spectral "data-aware" algorithms over high-dimensional data that contains a low-dimensional substructure, and showing that such an adaptive approach can provide exponentially better guarantees than worst-case analysis.

1.4 Organization of this dissertation

The remainder of this dissertation has five chapters, which we summarize below.

Chapter 2 presents nearest neighbor algorithms for Bregman divergences in low-dimensions. We propose parametrizing the degree of violation of triangle inequality, μ , and showing that we can bound the efficacy of ring-tree and quad-tree search thereby. We also leverage that Bregman divergences admit packings and grid-like decomposition in obtaining our bounds.

Following on from the results of the last chapter, Chapter 3 explores whether in some respects an algorithmic dependence on μ is essential. We show that indeed, in the high-dimensional case, μ does control the space requirement of any algorithm that uses a small number of queries.

In Chapter 4, we confine ourselves to a subclass of information divergences that are metrics and admit infinite-dimensional Hilbert space embeddings. We show that as opposed to more general Bregman divergences, these more structured spaces admit both dimensionality reduction from the simplex to simplex, as well as efficient sketching schemes.

In Chapter 5, we focus on ANN algorithms for data with some low dimensional spectral substructure. This corresponds with many heuristics used in practice, as well as the majority of real-world data sets. We show that theoretical bounds obtained in this setting far outperform hashing and randomized projection techniques used for worst-case data sets.

Finally, a short note in Chapter 6 suggests future directions.

CHAPTER 2

UPPER BOUND FOR BREGMAN ANN IN LOW DIMENSIONS

Our first algorithm resolves a query for a d-dimensional $(1+\varepsilon)$-ANN time $O\left(\left(\frac{\mu \log n}{\varepsilon}\right)^{O(d)}\right)$ and $O\left(n \log^{d-1} n\right)$ space and holds for generic μ-defective distance measures satisfying a Reverse Triangle Inequality (RTI). Our second algorithm is more specific in analysis to the Bregman divergences and uses a further structural parameter, the maximum ratio of second derivatives over each dimension of our allowed domain (c_0). This allows us to locate a $(1+\varepsilon)$-ANN in $O(\log n)$ time and $O(n)$ space, where there is a further $(c_0)^d$ factor in the big-Oh for the query time.

2.1 Definitions and overview of techniques

At a high level (139), low-dimensional Euclidean approximate near-neighbor search works as follows. The algorithm builds a quadtree-like data structure to search the space efficiently at query time. Cells reduce exponentially in size, and so a careful application of the triangle inequality and some packing bounds allows us to bound the number of cells explored in terms of the "spread" of the point set (the ratio of the maximum to minimum distance). Next, terms involving the spread are eliminated by finding an initial crude approximation to the nearest neighbor. Since the resulting depth to explore is bounded by the logarithm of the ratio of the cell sizes, any c-approximation of the nearest neighbor results in a depth of $O(\log(c/\varepsilon))$. A standard data structure that yields such a crude bound is the *ring-tree* (94).

Unfortunately, these methods (which work also for doubling metrics (32, 50, 94)) require two key properties: the existence of the triangle inequality, as well as packing bounds for fitting small-radius balls into large-radius balls. Bregman divergences in general are not symmetric and do not even satisfy a directed triangle inequality! We note in passing that such problems do not occur for the *exact* nearest neighbor problem in constant dimension; this problem reduces to point location in a Voronoi diagram, and Bregman Voronoi diagrams possess the same combinatorial structure as Euclidean Voronoi diagrams (34).

2.1.1 Reverse triangle inequality

The first observation we make is that while Bregman divergences do not satisfy a triangle inequality, they satisfy a weak *reverse triangle inequality*: along a line, the sum of lengths of two contiguous intervals is always *less* than the length of the union. This immediately yields a packing bound: intuitively, we cannot pack many disjoint intervals in a larger interval because their sum would be too large, violating the reverse triangle inequality.

2.1.2 μ-defectiveness

The second idea is to allow for a *relaxed* triangle inequality. We do so by defining a distance measure to be μ*-defective* w.r.t. a given domain if there exists a fixed $\mu \geq 1$ such that for all triples of points x, y, z, we have that $|D(x,y) - D(x,z)| \leq \mu D(y,z)$. This notion was employed by Farago *et al.* (66) for an algorithm optimizing average case complexity.

A different natural way to relax the triangle inequality would be to show there exists a fixed $\mu < 1$ such that for all triples (x,y,z), the inequality $D(x,y) + D(y,z) \geq \mu D(x,z)$. In fact, this is the notion of μ*-similarity* used by Ackermann *et al.* (3) to *cluster* data under a Bregman divergence. However, this version of a relaxed triangle inequality is too weak for the nearest-neighbor problem.

In fact, the relevant relaxation of the triangle inequality that we require is slightly different. Rearranging terms, we instead require that there exists a parameter $\mu \geq 1$ such that for all triples (x,y,z), $|D(x,y) - D(x,z)| \leq \mu D(y,z)$. We call such a distance μ*-defective*. It is fairly straightforward to see that a μ-defective distance measure is also $2/(\mu+1)$-similar, but the converse does not hold, as the example above shows.

Without loss of generality, assume that $D(x,y) \geq D(x,z) \geq D(y,z)$. Then $D(x,y) - D(x,z) \leq \mu D(y,z)$ and $D(x,y) - D(y,z) \leq \mu D(x,z)$, so $2D(x,y) \leq (\mu+1)(D(x,z) + D(y,z))$. Since $D(x,y)$ is the greatest of the three distances, this inequality is the strongest and implies the corresponding $2/(\mu+1)$-similarity inequalities for the other two distances.

Unfortunately, Bregman divergences do not satisfy μ-defectiveness for any size domain or value of μ! One of our technical contributions is demonstrating in Section 2.2 that surprisingly, the square root of Bregman divergences does satisfy this property with μ depending on the boundedness of the domain and choice of divergence.

2.1.3 A generic approximate near-neighbor algorithm

After establishing that Bregman divergences satisfy the reverse triangle inequality and μ-defectiveness (Section 2.2), we first show (Section 2.4) that *any* distance measure satisfying the reverse triangle inequality, μ-defectiveness, and some mild technical conditions admit a ring-tree-based construction to obtain a weak near neighbor. However, applying it to a quadtree construction creates a problem. The μ-defectiveness of a distance measure means that if we take a unit length interval and divide it into two parts, all we can expect is that each part has length between $1/2$ and $1/(\mu+1)$. This implies that while we may have to go down to level $\lceil \log_2 \ell \rceil$ to guarantee that all cells have side length $O(\ell)$, some cells might have side length as little as $\ell^{\log_2(\mu+1)}$, weakening packing bounds considerably.

We deal with this problem in two ways. For Bregman divergences, we can exploit geometric properties of the associated convex function ϕ (see Section 2.1.4) to ensure that cells at a fixed level have bounded size (Section 2.6); this is achieved by reexamining the second derivative ϕ''. For more general abstract distances that satisfy the reverse triangle inequality and μ-defectiveness, we instead construct a portion of the quad tree "on the fly" for each query (Section 2.5). While this is expensive, it still yields polylog(n) bounds for the overall query time in fixed dimensions. Both of these algorithms rely on packing/covering bounds that we prove in Section 2.3.

An important technical point is that for exposition and simplicity, we initially work with the *symmetrized* Bregman divergences (of the form $D_{s\phi}(x,y) = D_\phi(x \mid y) + D_\phi(y \mid x)$), and then extend these results to general Bregman divergences (Section 2.7). We note that the results for symmetrized Bregman divergences might be interesting in their own right, as they have also been used in applications (116, 118, 119, 120).

2.1.4 Definitions

In this chapter we study the approximate nearest neighbor problem for distance functions D: Given a point set P, a query point q, and an error parameter ε, find a point $\mathrm{nn}_q \in P$ such that $D(\mathrm{nn}_q, q) \leq (1+\varepsilon)\min_{p \in P} D(p,q)$. We start by defining general properties that we will require of our distance measures. In what follows, we will assume that the distance measure D is *reflexive*: $D(x,y) = 0$ if and only if $x = y$ and otherwise $D(x,y) > 0$.

Definition 1 (Monotonicity) *Let $M \subset \mathbb{R}$, $D : M \times M \to \mathbb{R}$ be a distance function. We say that D is* monotonic *if and only if for all $a < b < c$ we have that $D(a,b) \leq D(a,c)$ and $D(b,c) \leq D(a,c)$.*

For a general distance function $D : M \times M \to \mathbb{R}$, where $M \subset \mathbb{R}^d$, we say that D is monotonic if it is monotonic when restricted to any subset of M parallel to a coordinate axis.

Definition 2 (Reverse Triangle Inequality (RTI)) *Let M be a subset of $\mathbb{R}$. We say that a monotone distance measure $D : M \times M \to \mathbb{R}$ satisfies a* reverse triangle inequality *or RTI if for any three elements $a \leq b \leq c \in M$, $D(a,b) + D(b,c) \leq D(a,c)$.*

Definition 3 (μ-defectiveness) *Let D be a symmetric monotone distance measure satisfying the reverse triangle inequality. We say that D is* μ-defective *with respect to domain M if for all $a,b,q \in M$,*

$$|D(a,q) - D(b,q)| < \mu D(a,b). \tag{2.1}$$

For an asymmetric distance measure D, we define left and right sided μ-defectiveness respectively as

$$|D(q,a) - D(q,b)| < \mu D(a,b) \tag{2.2}$$

and

$$|D(a,q) - D(b,q)| < \mu D(b,a). \tag{2.3}$$

Note that by interchanging a and b and using the symmetry of the modulus sign, we can also rewrite left and right sided μ-defectiveness respectively as $|D(q,a) - D(q,b)| < \mu D(b,a)$ and $|D(a,q) - D(b,q)| < \mu D(a,b)$.

2.1.5 Two technical notes

The distance functions under consideration are typically defined over $\mathbb{R}^d$. We will assume in this chapter that the distance D is *decomposable*: roughly, that $D((x_1,\ldots,x_d),$

$(y_1,\dots,y_d))$ can be written as $g(\sum_i f(x_i,y_i))$, where g and f are monotone. This captures all the Bregman divergences that are typically used (with the exception of the Mahalanobis distance and matrix distances), as discussed in Chapter 1.

We will also need to compute the diameter of an axis parallel box of side-length ℓ. Our results hold as long as the diameter of such a box is $O(\ell d^{O(1)})$: note that this captures standard distances like those induced by norms, as well as decomposable Bregman divergences. In what follows, we will mostly make use of the *square root* of a Bregman divergence, for which the diameter of a box is $\ell(\mu+1)\sqrt{d}$ or $\ell\sqrt{d}$, and so without loss of generality we will use this in our bounds.

In this chapter we will hence limit ourselves to considering decomposable distance measures. We note that due to the primal-dual relationship of $D_\phi(a,b)$ and $D_{\phi^*}(b^*,a^*)$, for our results on the asymmetric Bregman divergence we need only consider right-sided nearest neighbors and left-sided results follow symmetrically.

2.1.6 Some notes on terminology and computation model

We note now that whenever we refer to "bisecting" an interval $[ab]$ under a distance measure D satisfying an RTI, we shall precisely mean finding x s.t. $D(a,x) = D(x,b)$. The RTI now implies that $D(a,x) = D(x,b) \leq \frac{1}{2}D(a,b)$ and that repeated bisection quickly reduces the length of subintervals. Computing such a bisecting point of an interval exactly, or even placing a point at a specified distance from a given point p is not trivial. However we argue in Section 2.8 that both tasks can be approximately done by numerical procedures without significantly affecting our asymptotic bounds. For the remainder of the chapter we shall take an idealized context and assume any such computations can be done to the desired accuracy quickly.

We also stipulate that the "diameter" of any subset of our domain $X \subset M$ under distance measure D shall be $\max_{x,y\in X} D(x,y)$. Where the choice of distance measure D may appear ambiguous from the context, we shall explicitly refer to the D-diameter.

2.2 Properties of Bregman divergences

Section 2.1 defined key properties that we desire of a distance function D. The Bregman divergences (or modifications thereof) satisfy the following properties, as can be shown by direct computation.

Lemma 2.2.1 *Any one-dimensional Bregman divergence is monotonic.*

Lemma 2.2.2 *Any one-dimensional Bregman divergence satisfies the reverse triangle inequality. Let $a \leq b \leq c$ be three points in the domain of D_ϕ. Then it holds that*

$$D_\phi(a,b) + D_\phi(b,c) \leq D_\phi(a,c) \tag{2.4}$$

and

$$D_\phi(c,b) + D_\phi(b,a) \leq D_\phi(c,a). \tag{2.5}$$

Proof. We prove the first case, the second follows almost identically.

$$\begin{aligned} D_\phi(a,b) + D_\phi(b,c) &= \phi(a) - \phi(b) - \phi'(b)(a-b) + \phi(b) - \phi(c) - \phi'(c)(b-c) \\ &= \phi(a) - \phi(c) - \phi'(b)(a-b) - \phi'(c)(b-c). \end{aligned}$$

But since $\phi''(x) \geq 0$ for all $x \in \mathbb{R}$, by convexity of ϕ we have that $\phi'(b) \leq \phi'(c)$. This allows us to make the substitution.

$$\begin{aligned} D_\phi(a,b) + D_\phi(b,c) &= \phi(a) - \phi(c) - \phi'(b)(a-b) - \phi'(c)(b-c) \\ &\leq \phi(a) - \phi(c) - \phi'(c)(a-b) - \phi'(c)(b-c) \\ &= \phi(a) - \phi(c) - \phi'(c)(a-c) = D_\phi(a,c). \end{aligned}$$

■

Note that this lemma can be extended similarly by induction to any series of n points between a and c. Further, using the relationship between $D_\phi(a,b)$ and the "dual" distance $D_{\phi^*}(b^*,a^*)$, we can show that the reverse triangle inequality holds going "left" as well: $D_\phi(c,b) + D_\phi(b,a) \leq D_\phi(c,a)$. These two separate reverse triangle inequalities together yield the result for $D_{s\phi}$. We also get a similar result for $\sqrt{D_{s\phi}}$ by algebraic manipulations.

Lemma 2.2.3 $\sqrt{D_{s\phi}}$ *satisfies the reverse triangle inequality.*

Proof. Fix $a \leq x \leq b$, and assume that the reverse triangle inequality does not hold:

$$\sqrt{D_{s\phi}(a,x)} + \sqrt{D_{s\phi}(x,b)} > \sqrt{D_{s\phi}(a,b)}$$
$$\sqrt{(x-a)(\phi'(x)-\phi'(a))} + \sqrt{(b-x)(\phi'(b)-\phi'(x))} > \sqrt{(b-a)(\phi'(b)-\phi'(a))}.$$

Squaring both sides, we get:

$$\begin{aligned}
&(x-a)(\phi'(x)-\phi'(a))+(b-x)(\phi'(b)-\phi'(x))\\
&\qquad +2\sqrt{(x-a)(b-x)(\phi'(x)-\phi'(a))(\phi'(b)-\phi'(x))} > (b-a)(\phi'(b)-\phi'(a))\\
&(b-x)(\phi'(x)-\phi'(a))+(x-a)(\phi'(b)-\phi'(x))\\
&\qquad -2\sqrt{(x-a)(b-x)(\phi'(x)-\phi'(a))(\phi'(b)-\phi'(x))} < 0\\
&\left(\sqrt{(b-x)(\phi'(x)-\phi'(a))}-\sqrt{(x-a)(\phi'(b)-\phi'(x))}\right)^2 < 0,
\end{aligned}$$

which is a contradiction, since the LHS is a perfect square. ■

While the Bregman divergences satisfy both monotonicity and the reverse triangle inequality, they are not μ-defective with respect to *any* domain! An easy example of this is ℓ_2^2, which is also a Bregman divergence. A surprising fact however is that $\sqrt{D_{s\phi}}$ and $\sqrt{D_\phi}$ do satisfy μ-defectiveness (with μ depending on the bounded size of our domain). While we were unable to show precise bounds for μ in terms of the domain, the values are small. For example, for the symmetrized KL-divergence on the simplex where each coordinate is bounded between 0.1 and 0.9, μ is 1.22. If each coordinate is between 0.01 and 0.99, then μ is 2.42. We discuss the empirical values of μ in greater detail in the Appendix. The proofs showing μ is bounded are somewhat tedious and not highly insightful, so we place those in the appendix as well for the interested reader.

Lemma 2.2.4 *Given any interval $I = [x_1 x_2]$ on the real line, there exists a finite μ such that $\sqrt{D_{s\phi}}$ is μ-defective with respect to I. We require all order derivatives of ϕ to be defined and bounded over the closure of I, and ϕ'' to be bounded away from zero.*

Proof. Refer to Appendix. ■

We note that the result for $\sqrt{D_\phi}$ is proven by establishing the following relationship between $D_\phi(a,b)$ and $D_\phi(b,a)$ over a bounded interval $I \subset \mathbb{R}$, and with some further computation.

Lemma 2.2.5 *Given a divergence D_ϕ and a bounded interval $I \subset \mathbb{R}$, $\sqrt{D_\phi(a,b)}/\sqrt{D_\phi(b,a)}$ is bounded by a parameter c_0 $\forall a,b \in I$ where c_0 depends on the choice of divergence and interval. We also require the derivatives of ϕ to be defined and bounded over the closure of I, and ϕ'' to be bounded away from zero.*

Proof. By continuity, compactness and the strict convexity of ϕ, we have that over a finite interval I $c_0 = \max_x \phi_i''(x)/\min_y \phi_i''(y)$ is bounded. Now by using the Lagrange form of $\sqrt{D_\phi(a,b)}$, we get that $\sqrt{D_\phi(a,b)}/\sqrt{D_\phi(b,a)} < \sqrt{c_0}$. ■

Lemma 2.2.6 *Given any interval $I = [x_1 x_2]$ on the real line, there exists a finite μ such that $\sqrt{D_\phi}$ is right-sided μ-defective with respect to I. We require all order derivatives of ϕ to be defined and bounded over the closure of I, and ϕ'' to be bounded away from zero.*

Proof. Refer to Appendix. ■

We extend our results to d dimensions naturally now by showing that if M is a domain such that $\sqrt{D_{s\phi}}$ and $\sqrt{D_\phi}$ are μ-defective with respect to the projection of M onto each coordinate axis, then $\sqrt{D_{s\phi}}$ and $\sqrt{D_\phi}$ are μ-defective with respect to all of M.

Lemma 2.2.7 *Consider three points, $a = (a_1,\ldots,a_i,\ldots,a_d)$, $b = (b_1,\ldots,b_i,\ldots,b_d)$, $q = (q_1,\ldots,q_i,\ldots,q_d)$ such that $|\sqrt{D_{s\phi}(a_i,q_i)} - \sqrt{D_{s\phi}(b_i,q_i)}| < \mu\sqrt{D_{s\phi}(a_i,b_i)}, \forall 1 \le i \le d$. Then*

$$\left|\sqrt{D_{s\phi}(a,q)} - \sqrt{D_{s\phi}(b,q)}\right| < \mu\sqrt{D_{s\phi}(a,b)}. \tag{2.6}$$

Similarly, if $|\sqrt{D_\phi(a_i,q_i)} - \sqrt{D_\phi(b_i,q_i)}| < \mu\sqrt{D_\phi(a_i,b_i)}, \forall 1 \le i \le d$. Then

$$\left|\sqrt{D_\phi(a,q)} - \sqrt{D_\phi(b,q)}\right| < \mu\sqrt{D_\phi(b,a)}. \tag{2.7}$$

Proof.

$$\begin{aligned}
\left|\sqrt{D_{s\phi}(a,q)} - \sqrt{D_{s\phi}(b,q)}\right| &< \mu\sqrt{D_{s\phi}(a,b)} \\
D_{s\phi}(a,q) + D_{s\phi}(b,q) - 2\sqrt{D_{s\phi}(a,q)D_{s\phi}(b,q)} &< \mu^2 D_{s\phi}(a,b) \\
\sum_{i=1}^{d}\Big(D_{s\phi}(a_i,q_i) + D_{s\phi}(b_i,q_i)\Big) - 2\sqrt{D_{s\phi}(a,q)D_{s\phi}(b,q)} &< \mu^2 \sum_{i=1}^{d} D_{s\phi}(a_i,b_i) \\
\sum_{i=1}^{d}\Big(D_{s\phi}(a_i,q_i) + D_{s\phi}(b_i,q_i) - \mu^2 D_{s\phi}(a_i,b_i)\Big) &< 2\sqrt{D_{s\phi}(a,q)D_{s\phi}(b,q)}.
\end{aligned}$$

The last inequality is what we need to prove for μ-defectiveness with respect to a,b,q. By assumption we already have μ-defectiveness w.r.t. each a_i,b_i,q_i, for every $1 \le i \le d$,

$$D_{s\phi}(a_i,q_i)+D_{s\phi}(b_i,q_i)-\mu^2 D_{s\phi}(a_i,b_i) < 2\sqrt{D_{s\phi}(a_i,q_i)D_{s\phi}(b_i,q_i)}$$

$$\sum_{i=1}^{d}\left(D_{s\phi}(a_i,q_i)+D_{s\phi}(b_i,q_i)-\mu^2 D_{s\phi}(a_i,b_i)\right) < 2\sum_{i=1}^{d}\sqrt{D_{s\phi}(a_i,q_i)D_{s\phi}(b_i,q_i)}.$$

So to complete our proof we need only show

$$\sum_{i=1}^{d}\sqrt{D_{s\phi}(a_i,q_i)}\sqrt{D_{s\phi}(b_i,q_i)} \le \sqrt{D_{s\phi}(a,q)}\sqrt{D_{s\phi}(b,q)}. \tag{2.8}$$

But notice the following:

$$\sqrt{D_{s\phi}(a,q)} = \left(\sum_{i=1}^{d} D_{s\phi}(a_i,q_i)\right)^{\frac{1}{2}} = \left(\sum_{i=1}^{d}\left(\sqrt{D_{s\phi}(a_i,q_i)}\right)^2\right)^{\frac{1}{2}}.$$

$$\sqrt{D_{s\phi}(b,q)} = \left(\sum_{i=1}^{d} D_{s\phi}(b_i,q_i)\right)^{\frac{1}{2}} = \left(\sum_{i=1}^{d}\left(\sqrt{D_{s\phi}(b_i,q_i)}\right)^2\right)^{\frac{1}{2}}.$$

So inequality 2.8 is simply a form of the Cauchy-Schwarz inequality, which states that for two vectors u and v in $\mathbb{R}^d$, that $|\langle u,v\rangle| \le \|u\|\|v\|$, or that

$$\left|\sum_{i=1}^{d} u_i v_i\right| \le \left(\sum_{i=1}^{d} u_i^2\right)^{\frac{1}{2}}\left(\sum_{i=1}^{d} v_i^2\right)^{\frac{1}{2}}.$$

The second part of the proposition can be derived by an essentially identical argument.

■

2.3 Packing and covering bounds

The aforementioned key properties (monotonicity, the reverse triangle inequality, decomposability, and μ-defectiveness) can be used to prove packing and covering bounds for a distance measure D. We now present some of these bounds.

2.3.1 Covering bounds in one dimension

Lemma 2.3.1 (Interval packing) *Consider a monotone distance measure D satisfying the reverse triangle inequality, an interval $[ab]$ such that $D(a,b)=s$ and a collection of disjoint intervals intersecting $[ab]$, where $I=\{[xx'] \mid [xx'], D(x,x')\ge \ell\}$. Then $|I| \le \frac{s}{\ell}+2$.*

Proof. Let I' be the intervals of I that are totally contained in $[ab]$. The combined length under D of all intervals in I' is at least $|I'|\ell$, but by the reverse triangle inequality their total length cannot exceed s, so $|I'| \leq \frac{s}{\ell}$. There can be only two members of I not in I', so $|I| \leq \frac{s}{\ell} + 2$. ■

A simple greedy approach yields a constructive version of this lemma.

Corollary 2.3.1 *Given any two points, $a \leq b$ on the line s.t. $D(a,b) = s$, we can construct a packing of $[ab]$ by $r \leq \frac{1}{\varepsilon}$ intervals $[x_i x_{i+1}]$, $1 \leq i \leq r$ s.t. $D(a,x_0) = D(x_i,x_{i+1}) = \varepsilon s$, $\forall i$ and $D(x_r,b) \leq \varepsilon s$. Here D is a monotone distance satisfying the reverse triangle inequality.*

We recall here that D_ϕ, $D_{s\phi}$ and $\sqrt{D_{s\phi}}$ satisfy the conditions of lemma 2.3.1 and corollary 2.3.1 as they satisfy an RTI and are decomposable. However, since $\sqrt{D_\phi}$ may not satisfy the reverse triangle inequality, we instead prove a weaker packing bound on $\sqrt{D_\phi}$ by using D_ϕ.

Lemma 2.3.2 (Weak interval packing) *Given distance measure $\sqrt{D_\phi}$ and an interval $[ab]$ such that $\sqrt{D_\phi}(a,b) = s$ and a collection of disjoint intervals intersecting $[ab]$ where $I = \{[xx'] \mid [xx'], \sqrt{D_\phi}(x,x') \geq \ell\}$. Then $|I| \leq \frac{s^2}{\ell^2} + 2$. Such a set of intervals can be explicitly constructed.*

Proof. We note that here $D_\phi(a,b) = s^2$, and $I = \{[xx'] \mid [xx'], D_\phi(x,x') \geq \ell^2\}$. The result then follows trivially from lemma 2.3.1, since D_ϕ satisfies the conditions of lemma 2.3.1. ■

2.3.2 Properties of cubes and their coverings

The one-dimensional bounds can be generalized to higher dimensions to provide packing bounds for balls and cubes (which we define below) with respect to a monotone, decomposable distance measure.

Definition 4 *Given a collection of d intervals a_i, b_i and distance measure D, s.t. $D(a_i,b_i) = s$ where $1 \leq i \leq d$, the* cube *in d dimensions is defined as $\prod_{i=i}^{d} [a_i b_i]$ and is said to have* side-length *s. We shall specify the choice of D by referring to the cube as either a D_ϕ-cube, $D_{s\phi}$-cube, $\sqrt{D_\phi}$-cube, $\sqrt{D_\phi}$-cube or a $\sqrt{D_{s\phi}}$-cube. Where we make an argument that holds for more than one of these types of cubes, we shall refer to simply a D-cube where the possible values of D will be specified. We follow the same convention for balls.*

We add that for a given distance measure D, a *box* H can be defined similarly to a cube, except that the side lengths need not necessarily be equal. In this case we let $H = \prod_{i=i}^{d}[a_i b_i]$ and let the ith side-length be $D(a_i, b_i)$. Again where the choice of distance measure D appears at all ambiguous we shall refer to the D side-length.

We pause here to note that for an asymmetric decomposable measure D in d dimensions, every D-box has an implied associated ordering on each of the d composing intervals. For a D-box defined as prod $\prod_{i=1}^{d}[a_i b_i]$ and bisected by a collection of x_i such that $D(a_i, x_i) = D(x_i, b_i)$, there will be 2^d subboxes produced such that their ith composing interval will be either $[a_i x_i]$ or $[x_i b_i]$. See Figure 2.1. In what can be viewed as a generalization of bisection to splitting each side of a D-cube into multiple subintervals, we show the following useful lemma that acts as a building block:

Lemma 2.3.3 *Given a d-dimensional D-cube B of side-length s under distance measure D, we can cover it with at most* $\frac{1}{\varepsilon^d}$ *D-cubes of side-length* exactly εs *under the same measure D, where D may be either* D_ϕ, $D_{s\phi}$ *and* $\sqrt{D_{s\phi}}$.

Proof. Note that $D_{s\phi}$, D_ϕ and $\sqrt{D_{s\phi}}$ satisfy all the conditions of corollary 2.3.1. Hence we can employ the packing of at most $\frac{1}{\varepsilon}$ points in each dimension spaced εs apart. We then take a product over all d dimensions, and the lemma now follows in a straightforward manner. ■

Weaker packing bounds for $\sqrt{D_\phi}$ as noted in lemma 2.3.2 yield us a weaker version of lemma 2.3.3.

Lemma 2.3.4 *Given a d-dimensional* $\sqrt{D_\phi}$*-cube B of side-length s, we can cover it with at most* $\frac{1}{\varepsilon^{2d}}$ $\sqrt{D_\phi}$*-cubes of side-length* exactly εs.

Proof. Identical to the proof of lemma 2.3.3 and using lemma 2.3.2 to obtain packing bounds. ■

We note that this subdivision of D-cubes corresponds to placing an equal number of points (the vertices of the cubes), and this is what we shall refer to more loosely as *gridding* in the remainder of our chapter. We shall employ this subdivision next to show results for covering of balls.

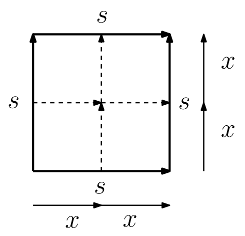


Figure 2.1. A cube of directed side length s subdivided into cubes of side length $x \leq \frac{s}{2}$

2.3.3 Covering with balls in higher dimensions

Covering a D-ball with a number of smaller D-balls is a key ingredient in our results. Our approach is to divide a D-ball into 2^d orthants, then to show each orthant can be covered by a certain number of smaller D-cubes, and then finally that each such D-cube can be covered by a D-ball of a certain radius.

We show now results for $D_{s\phi}$, D_{ϕ}, $\sqrt{D_{\phi}}$ and $\sqrt{D_{s\phi}}$. We present first the easier cases for the two symmetric measures, $D_{s\phi}$ and $\sqrt{D_{s\phi}}$.

Lemma 2.3.5 *A $D_{s\phi}$-cube in d dimensions of side-length s can be covered by a $D_{s\phi}$-ball of radius ds. Similarly, a $\sqrt{D_{s\phi}}$-cube in d dimensions of side-length s can be covered by a $\sqrt{D_{s\phi}}$-ball of radius $\sqrt{d}s$.*

Proof. Recall that a $D_{s\phi}$-cube is defined as $\prod_{i=1}^{d}[a_i b_i]$ s.t $D_{s\phi_i}(a_i, b_i) = s$ (where $D_{s\phi_i}(a_i, b_i)$ is induced by restricting $D_{s\phi}$ to the ith dimension). Let the vertex space of the $D_{s\phi}$-cube be $V = \prod_{i=1}^{d} v_i$, where $v_i \in \{a_i, b_i\}$. Now pick an arbitrary vertex $x \in V$, and consider the $D_{s\phi}$-ball B of radius ds with center v. By decomposability and monotonicity, for any $y \in V$, we have

$$D_{s\phi}(x,y) = \sum_{i=1}^{d} D_{s\phi_i}(x_i, y_i) \leq \sum_{i=1}^{d} D_{s\phi_i}(a_i, b_i)$$
$$= \sum_{i=1}^{d} s = ds.$$

Hence an $D_{s\phi}$-cube of side-length s can be covered by an $D_{s\phi}$-ball of radius ds. The second result follows by noting that an $\sqrt{D_{s\phi}}$-cube of side-length s is an $D_{s\phi}$-cube of side-length s^2. Hence this can be covered by an $D_{s\phi}$-ball of radius ds^2, which is simply an $\sqrt{D_{s\phi}}$ ball of radius $\sqrt{d}s$. ■

Lemma 2.3.6 *A D_ϕ-cube in d dimensions of side-length s can be covered by a D_ϕ-ball of radius ds. Similarly, a $\sqrt{D_\phi}$-cube in d dimensions of side-length s can be covered by a $\sqrt{D_{s\phi}}$-ball of radius $\sqrt{d}s$.*

Proof. Similar to lemma 2.3.5, we begin by recalling that a D_ϕ-cube is defined as $\prod_{i=1}^{d}[a_i b_i]$ s.t $D_{\phi_i}(a_i,b_i) = s$ (where $D_{\phi_i}(a_i,b_i)$ is induced by restricting D_ϕ to the ith dimension). We again let the vertex space of the D_ϕ-cube be $V = \prod_{i=1}^{d} v_i$, where $v_i \in \{a_i,b_i\}$. Now we have to be somewhat more careful in our choice of center for the D_ϕ-ball B of radius ds than we were in lemma 2.3.5. We choose the "lowest" point of the D_ϕ-cube, which is $x = \prod_{i=1}^{d} a_i$ (see Figure 2.2) and term this as a *canonical corner*.We note that our definition does not require that $a_i \le b_i$. Now for any other $y \in V$ we have

$$D_\phi(x,y) = \sum_{i=1}^{d} D_{\phi_i}(x_i,y_i) \le \sum_{i=1}^{d} D_{\phi_i}(a_i,b_i) = \sum_{i=1}^{d} s = ds.$$

The argument for $\sqrt{D_\phi}$ follows analogously to that for $\sqrt{D_{s\phi}}$ in lemma 2.3.5. ■

We will also find the following relation of the diameter of a $\sqrt{D_{s\phi}}$-cubes to the $\sqrt{D_\phi}$ side-length useful later in this chapter.

Lemma 2.3.7 *The diameter of an $\sqrt{D_{s\phi}}$-cube of side-length s is bounded by $\sqrt{d}s$.*

Proof. Consider any two points x and y in the $\sqrt{D_{s\phi}}$-cube of $\sqrt{D_{s\phi}}$-side-length s and defined as $\prod_{i=1}^{d}[a_i b_i]$. Note that since $x_i, y_i \in [a_i b_i]$ we have that $D_{s\phi}(x_i,y_i) \le s^2$. Hence $D_{s\phi}(x,y) \le ds^2$ and $\sqrt{D_{s\phi}}(x,y) \le \sqrt{d}s$. ■

Corollary 2.3.2 *For any $\sqrt{D_{s\phi}}$-box of maximum $\sqrt{D_{s\phi}}$-side length s, the diameter of the box is bounded by $\sqrt{d}s$.*

We now proceed to showing covering bounds for $\sqrt{D_{s\phi}}$ and $\sqrt{D_\phi}$ using the geometry we have developed thus far.

Lemma 2.3.8 *Consider a D-ball B of radius s and center c. Then in the case of $D = D_{s\phi}$, B can be covered with $\frac{2^d}{\varepsilon^d}$ $D_{s\phi}$-balls of radius $d\varepsilon s$. In the case of $D = \sqrt{D_{s\phi}}$, B can be covered with $\frac{2^d}{\varepsilon^d}$ $\sqrt{D_{s\phi}}$-balls of radius $\sqrt{d}\varepsilon s$.*

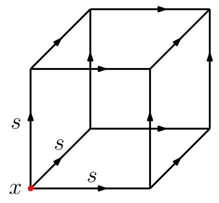


Figure 2.2. x is within ds distance under D_ϕ of every other point of the cube.

Proof. We divide the D-ball into 2^d orthants around the center c. Each orthant can be covered by a D-cube of size s. For both $D = D_{s\phi}$ and $D = \sqrt{D_{s\phi}}$, by lemma 2.3.3 each such D-cube can be broken down into $\frac{1}{\varepsilon^d}$ sub-D-cubes of side-length εs. By lemma 2.3.5, we can cover each such $D_{s\phi}$-cube by a $D_{s\phi}$-ball of radius $d\varepsilon s$ placed at any corner. Similarly, for $\sqrt{D_{s\phi}}$, we can cover each sub-$\sqrt{D_{s\phi}}$-cube by a $\sqrt{D_{s\phi}}$-ball of radius $\sqrt{d}\varepsilon s$ placed at any corner. Since there are $\frac{1}{\varepsilon^d}$ sub-D-cubes to each of the 2^d orthants whether $D = \sqrt{D_{s\phi}}$ or $D = D_{s\phi}$, respectively, the lemma now follows by covering each sub-D-cube with a D-ball of the required radius. ■

Lemma 2.3.9 *Consider a D-ball B of radius s and center c with respect to distance measure D. Then in the case of $D = D_\phi$, B can be covered with $\frac{2^d}{\varepsilon^d}$ D_ϕ-balls of radius $d\varepsilon s$. And for $D = \sqrt{D_\phi}$, B can be covered by $\frac{2^d}{\varepsilon^{2d}}$ $\sqrt{D_\phi}$-balls of radius $\sqrt{d}\varepsilon s$.*

Proof. We divide the D-ball into 2^d orthants around the center c. Each orthant can be covered by a D-cube of size s. We now consider each case separately. For D_ϕ, by lemma 2.3.3 each such D_ϕ-cube can be broken down into $\frac{1}{\varepsilon^d}$ D_ϕ-cubes of side-length εs. For $\sqrt{D_\phi}$, by lemma 2.3.4 we can break down each $\sqrt{D_\phi}$-cube into $\frac{1}{\varepsilon^{2d}}$ sub-$\sqrt{D_\phi}$-cubes of side-length εs.

By lemma 2.3.6, we can cover each such D_ϕ-cube by a D_ϕ-ball of radius $d\varepsilon s$ placed at a canonical corner. Similarly for $\sqrt{D_\phi}$, by lemma 2.3.6 we can cover each sub-$\sqrt{D_\phi}$-cube by a $\sqrt{D_\phi}$-ball of radius $\sqrt{d}\varepsilon s$ placed at a canonical corner. Since there are $\frac{1}{\varepsilon^{2d}}$ and $\frac{1}{\varepsilon^d}$ sub-D-cubes to each of the 2^d orthants for $D = \sqrt{D_\phi}$ and $D = D_\phi$, respectively, the lemma now follows by covering each sub-D-cube with a D-ball of the required radius. ■

2.4 Computing a rough approximation

To illustrate our techniques, we will focus on finding approximate nearest neighbors under $\sqrt{D_{s\phi}}$ over the next two sections. When we define our notation more generally, e.g., of a ring separator, we may use a more generic distance measure D.

Later we will show how our results can be extended to the asymmetric case with mild modifications and careful attention to directionality. We now describe how to compute a $O(\log n)$ rough approximate nearest-neighbor under $\sqrt{D_{s\phi}}$ on our point set P, which we will use in the next section to find the $(1+\varepsilon)$-approximate nearest neighbor. The technique we use is based on ring separators. Ring separators are a fairly old concept in geometry, notable appearances of which include the landmark paper by Indyk and Motwani (84). Our approach here is heavily influenced by Har-Peled and Mendel (78), and by Krauthgamer and Lee (94), and our presentation is along the template of the textbook by Har-Peled (139, Chapter 11).

We note here that the constant of $d^{d/2}$ which appears in our final bounds for storage and query time is specific to $\sqrt{D_{s\phi}}$. However, an argument on the same lines will yield a constant of $d^{O(d)}$ for any μ-defective, symmetric RTI-satisfying decomposable distance measure D such that the D-diameter of a cube of side-length 1 is bounded by $d^{O(1)}$.

Let $B(m,r)$ denote a D-ball of radius r centered at m, and let $B'(m,r)$ denote the complement (or exterior) of $B(m,r)$. A *ring* R is the difference of two concentric D-balls: $R = B(m,r_2) \setminus B(m,r_1), r_2 \geq r_1$. We will often refer to the larger D-ball $B(m,r_2)$ as B_{out} and the smaller D-ball as B_{in}. We use $P_{\text{out}}(R)$ to denote the set $P \cap B'_{\text{out}}$, and use $P_{\text{in}}(R)$ as $P \cap B_{\text{in}}$, where we may drop the reference to R when the context is obvious. A *t-ring separator* $R_{P,c}$ on a point set P is a ring such that $\frac{n}{c} < |P_{\text{in}}| < (1-\frac{1}{c})n$, $\frac{n}{c} < |P_{\text{out}}| < (1-\frac{1}{c})n$, $r_2 \geq (1+t)r_1$ and $B_{\text{out}} \setminus B_{\text{in}}$ is empty of points of P. A t-ring-tree is a binary tree obtained by repeated dispartition of our point set P using a t-ring separator. (We shall make the choice of distance measure D explicit whenever using a t-ring separator.)

Note that later on in this section, we will abuse this notation slightly by using ring-separators where the annulus is not actually empty, but we will bound the added space complexity and tree depth introduced. Finally, denote the minimum sized D-ball containing at least $\frac{n}{c}$ points of P by $B_{\text{opt},c}$; its radius is denoted by $r_{\text{opt},c}$.

We demonstrate that for any point set P a ring separator exists under $\sqrt{D_{s\phi}}$ and secondly, it can always be computed efficiently. Applying this "separator" recursively on our point structure yields a ring-tree structure for searching our point set. Before we proceed further, we need to establish some properties of disks under a μ-defective distance. lemma 2.4.1 is immediate from the definition of μ-defectiveness, lemma 2.4.2 is similar to one obtained by Har-Peled and Mazumdar (77) and the idea of repeating points in both children of a ring-separator derives from a result by Har-Peled and Mendel (78).

Lemma 2.4.1 *Let D be a μ-defective distance, and let $B(m,r)$ be a D-ball. Then for any two points $x,y \in B(m,r)$, $D(x,y) < (\mu+1)r$.*

Proof. Follows from the definition of μ-defectiveness.

$$D(x,y) - D(m,y) < \mu D(m,x)$$
$$D(x,y) < \mu r + D(m,y) \leq (\mu+1)r.$$

■

Corollary 2.4.1 *For any $\sqrt{D_{s\phi}}$-ball $B(m,r)$ and two points $x,y \in B(m,r)$, $\sqrt{D_{s\phi}}(x,y) < (\mu+1)r$.*

Proof. Since $\sqrt{D_{s\phi}}$ is μ-defective over a prespecified restricted domain. ■

Lemma 2.4.2 *Given a parameter $1 \leq c \leq n$, we can compute in $O(nc)$ expected time a $\mu+1$ approximation to the smallest radius $\sqrt{D_{s\phi}}$-ball containing $\frac{n}{c}$ points by the algorithm 1.*

Proof. As described by Har-Peled and Mazumdar (77) we let S be a random sample from P, generated by choosing every point of P with probability $\frac{c}{n}$. Next, compute for every $p \in S$, the smallest $\sqrt{D_{s\phi}}$-ball centered at p containing c points of P. By median selection, this can be done in $O(n)$ time and since $E(|S|) = c$, this gives us the expected running time of $O(nc)$. Now, let r' be the minimum radius computed. Note that by lemma 2.4.1, if $|S \cap B_{\text{opt},c}| > 0$ then we have that $r' \leq (\mu+1)r_{opt}$. But since $B_{\text{opt},c}$ contains $\frac{n}{c}$ points, we can upper bound the probability of failure as the probability that we do not select any of the $\frac{n}{c}$ points in B_{opt} in our sample. Hence,

$$Pr(|S \cap B_{\text{opt},c}| > 0) = 1 - \left(1 - \frac{c}{n}\right)^{\frac{n}{c}} \geq 1 - \frac{1}{e}.$$

Note that one can obtain a similar approximation deterministically by brute force search, but this would incur a prohibitive $O(n^2)$ running time. ■

```
$n \leftarrow |P|$
Choose $S$ by picking every $p \in P$ with probability $\frac{n}{c}$
$r \leftarrow \infty$
$B \leftarrow$ NULL
**for all** $s \in S$ **do**
    Compute smallest $\sqrt{D_{s\phi}}$-ball $B(s, r_S)$ with center $s$ that contains $c$ points of $P$.
    **if** $r_S < r$ **then**
        $B \leftarrow B(s, r_S)$
        $r \leftarrow r_S$
    **end if**
**end for**
**return** $B$
```

Algorithm 1: ApproxSmallestBall(P, c)

We can now use lemma 2.4.2 and the corresponding algorithm 2 to construct our ring-separator.

```
{Here $t > 1$, and the thickness of the separating ring is $O\left(\frac{1}{t} r_{\text{in}}\right)$}
$n \leftarrow |P|$
NODE IN $\leftarrow$ NULL
NODE OUT $\leftarrow$ NULL
$c \leftarrow 2(4(\mu+1)\sqrt{d})^d$
$B_1(m_1, r_1) \leftarrow$ ApproxSmallestBall$(P, c)$
$B_2(m_2, r_2) \leftarrow B(m_1, 2r_1)$
ANNULUS $\leftarrow B_2 \setminus B_1$
Divide ANNULUS into $t$ rings of equal thickness, such that RINGS$[i]$ is the $i$-th ring.
COUNT $\leftarrow \infty$
$r_{\text{in}} \leftarrow r_1$
**for all** $i = 1 \rightarrow t$ **do**
   **if** $|P \cap \text{RINGS}[i]| <$ COUNT **then**
      COUNT $\leftarrow |P \cap \text{RINGS}[i]|$
      $r_{\text{in}} \leftarrow r_1 + \left(\frac{i-1}{t}\right) r_1$
   **end if**
**end for**
**for all** $p \in P$ **do**
   **if** $\sqrt{D_{s\phi}(m_1, p)} \leq r_{\text{in}}$ **then**
      Add $p$ to IN
   **else if** $\sqrt{D_{s\phi}(m_1, p)} \geq r_{\text{in}} + \frac{r_1}{t}$ **then**
      Add $p$ to OUT
   **else**
      Add $p$ to IN and OUT
   **end if**
**end for**
COUNT-IN $\leftarrow$ number of points in IN
COUNT-OUT $\leftarrow$ number of points in OUT
**if** COUNT-IN $\geq \left(1 - \frac{1}{c}\right) n$ **or** COUNT-OUT $\geq \left(1 - \frac{1}{c}\right) n$ **then**
   **return** MakeRing $(P, t)$
   {This checks implictly that our earlier call to the randomized
   ApproxSmallestBall$(P, c)$ returned our desired approximation. If not, we try our
   procedure again as standard for Las Vegas algorithms.}
**else**
   **return** IN and OUT
**end if**
```

Algorithm 2: MakeRing(P, t)

Lemma 2.4.3 *For arbitrary t s.t $1 < t < n$ and $\sqrt{D_{s\phi}}$ in a μ-defective domain, we can construct a $\frac{1}{t}$-ring separator $R_{P,c}$ under $\sqrt{D_{s\phi}}$ in $O(n)$ expected time on a point set P by repeating points. See Algorithm 2.*

Proof. Using lemma 2.4.2, we compute a $\sqrt{D_{s\phi}}$-ball $S = B(m, r_1)$ (where $m \in P$) containing $\frac{n}{c}$ points such that $r_1 \leq (\mu+1)r_{\text{opt},c}$ where c is a parameter to be set. Consider the $\sqrt{D_{s\phi}}$-ball $\bar{S} = B(m, 2r_1)$. We shall argue that there must be $\frac{n}{c}$ points of P in the complement of $\bar{S}$, $\bar{S}'$, for careful choices of c. As described in lemma 2.3.8, $\bar{S}$ can be covered by 2^d hypercubes of side-length $2r_1$, the union of which we shall refer to as H. Set $L = (\mu+1)\sqrt{d}$. Imagine a partition of H into a grid, where each cell is of $\sqrt{D_{s\phi}}$-side-length $\frac{r_1}{L}$ and hence of diameter at most $\Delta(\frac{r_1}{L}, d) = \frac{r_1}{\mu+1} \leq r_{\text{opt},c}$ (by lemma 2.3.7). A $\sqrt{D_{s\phi}}$-ball of radius $r_{\text{opt},c}$ on any corner of a cell will contain the entire cell, and so it will contain at most $\frac{n}{c}$ points, by the definition of $r_{\text{opt},c}$.

By lemma 2.3.3 the grid on H has at most $2^d(2r_1/\frac{r_1}{L})^d = (4(\mu+1)\sqrt{d})^d$ cells. Set $c = 2(4(\mu+1)\sqrt{d})^d$. Then we have that $\bar{S} \subset H$ contains at most $\frac{n}{c}(4(\mu+1)\sqrt{d})^d = \frac{n}{2}$ points. Since the inner $\sqrt{D_{s\phi}}$-ball S contains at least $\frac{n}{c}$ points, and the outer $\sqrt{D_{s\phi}}$-ball $\bar{S}$ contains at most $\frac{n}{2}$ points, hence the annulus $\bar{S} \setminus S$ contains at most $\frac{n}{2} - \frac{n}{c}$ points. Now, divide $\bar{S} \setminus S$ into t rings of equal width, and by the pigeonhole principle at least one of these rings must contain at most $O(\frac{n}{t})$ points of P. Now let the inner $\sqrt{D_{s\phi}}$-ball corresponding to this ring be B_{in} and the outer $\sqrt{D_{s\phi}}$-ball be B_{out}. Let $P_{\text{in}} = P \cap B_{\text{in}}$, $P_{\text{out}} = P \cap B'_{\text{out}}$. Add any remaining points of P to *both* P_{in} and P_{out}(see Figure 2.3), i.e., consider that these points are duplicated and are in both sets.

Assign P_{in} and P_{out} to two nodes v_{in} and v_{out}, respectively. Even for $t = 1$, each node contains at most $\frac{n}{2} + (\frac{n}{2} - \frac{n}{c}) = (1 - \frac{1}{c})n$ points. Also, the thickness of the ring is bounded by $\frac{2r_1 - r_1}{t} / 2r_1 = \frac{1}{2t}$, i.e., it is a $O(\frac{1}{t})$ ring separator. Finally, we can check in $O(n)$ time if the randomized process of lemma 2.4.2 succeeded simply by verifying the number of points in the inner and outer ring is bounded by the values just computed. ■

Lemma 2.4.4 *Given any point set P under $\sqrt{D_{s\phi}}$ in a μ-defective domain, we can construct a $O(\frac{1}{\log n})$ ring-separator tree T of depth $O(d^{\frac{d}{2}}(\mu+1)^d \log n)$ by algorithm 3.*

Proof. Repeatedly partition P by lemma 2.4.2 into P^v_{in} and P^v_{out} where **v** is the parent

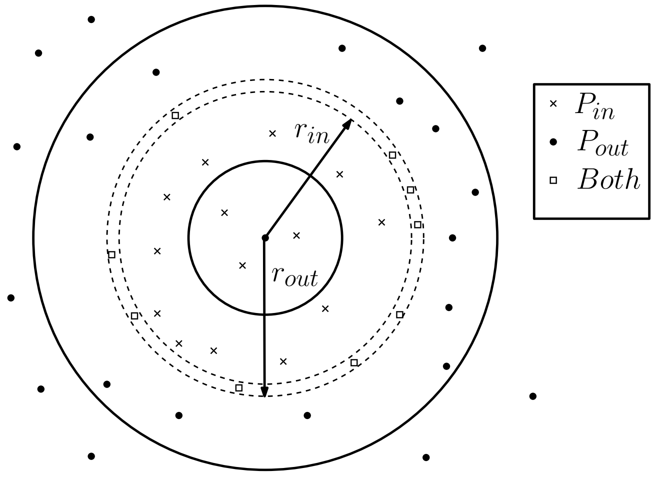


Figure 2.3. The points are split into P_{in} and P_{out} with some point duplication

node. Store only the single point $\text{rep}_v = m \in P$ in node $\mathbf{v}$, the center of the $\sqrt{D_{s\phi}}$-ball $B(m, r_1)$. We continue this partitioning until we have nodes with only a single point contained in them. Since each child contains at least $\frac{n}{c}$ points (by proof of lemma 2.4.3), each subset reduces by a factor of at least $1 - \frac{1}{c}$ at each step, and hence the depth of the tree is logarithmic.

We calculate the depth more exactly, noting that in lemma 2.4.3, $c = O(d^{\frac{d}{2}}(\mu+1)^d)$. Hence, the depth x can be bounded as

$$\begin{aligned} n(1-\frac{1}{c})^x &= 1 \\ (1-\frac{1}{c})^x &= \frac{1}{n} \\ x &= \frac{\ln\frac{1}{n}}{\ln(1-\frac{1}{c})} = \frac{-1}{\ln(1-\frac{1}{c})}\ln n \\ x &\leq c\ln n = O\left(d^{\frac{d}{2}}(\mu+1)^d \log n\right). \end{aligned}$$

■

Finally, we verify that the storage space required is not excessive.

Lemma 2.4.5 *To construct a $O(\frac{1}{\log n})$ ring-separator tree under $\sqrt{D_{s\phi}}$ in a μ-defective domain requires $O(n)$ storage and $O(d^{\frac{d}{2}}(\mu+1)^d n \log n)$ time.*

```
{Here $t < 1$ is the thickness of the ring w.r.t. radius of the inner ball.}
Add $P$ to ROOT
(IN, OUT) $\leftarrow$ MakeRing $\left(P, \frac{1}{t}\right)$
Set IN as a child of ROOT
Set OUT as a child of ROOT
MakeTree($P \cap$ IN , IN, $t$)
MakeTree($P \cap$ OUT , OUT, $t$)
```

Algorithm 3: MakeTree(P, NODE ROOT, t)

Proof. By lemma 2.4.4, the depth bounds still hold upon repeating points. For storage, we have to bound the total number of points in our data structure after repetition, let us say P_R. Since each node corresponds to a splitting of P_R, there may be only $O(P_R)$ nodes and total storage. We aim to show $|P_R| = O(|P|) = O(n)$. We begin by noting that in the proof of lemma 2.4.3, for a node containing x points, at most an additional $\frac{x}{\log n}$ may be duplicated in the two children.

To bound this over each level of our tree, we sum across each node to obtain that the number of points T_i in our structure at the i-th level, as

$$T_i = T_{i-1}\left(1 + \frac{1}{\log T_{i-1}}\right). \tag{2.9}$$

Note also by lemma 2.4.4, the tree depth is $O(\log n)$ or bounded by $k \log n$ where k is a constant. Hence we only need to bound the storage at the level $i = O(\log n)$. We solve the recurrence, noting that $T_0 = |P| = n$ (no points have been duplicated yet) and $T_i > n$ for all i and hence $T_i < T_{i-1}(1 + \frac{1}{\log n})$. Thus the recurrence works out to

$$T_i < n\left(1 + \frac{1}{\log n}\right)^{O(\log n)} < n\left(\left(1 + \frac{1}{\log n}\right)^{\log n}\right)^k < n(e^k),$$

where the main algebraic step is that $(1 + \frac{1}{x})^x < e$. This proves that the number of points, and hence our storage complexity is $O(n)$. Multiplying the depth by $O(n)$ for computing the smallest under $\sqrt{D_{s\phi}}$-ball across nodes on each level gives us the time complexity of $O(n \log n)$. We note that other tradeoffs are available for different values of approximation quality (t) and construction time / query time. ■

2.4.1 Algorithm and quality analysis

Let best_q be the best candidate for nearest neighbor to q found so far and $D_{\text{near}} = \sqrt{D_{s\phi}}(\text{best}_q, q)$. Let nn_q be the exact nearest neighbor to q from point set P and $D_{\text{exact}} =$

```
curr ← rep(ROOT)
best_q ← curr
D_near ← $\sqrt{D_{s\phi}}(q, \text{curr})$
**while** ROOT has children **do**
   curr ← rep(ROOT)
   $B(m, r_{\text{in}})$ is the inner ball associated with ROOT.
   **if** $\sqrt{D_{s\phi}}(q, \text{curr}) < D_{\text{near}}$ **then**
      best_q ← rep(ROOT)
      D_near ← $\sqrt{D_{s\phi}}(q, \text{best}_q)$
   **end if**
   **if** $\sqrt{D_{s\phi}}(q, \text{curr}) < (1 + \frac{t}{2})r_{\text{in}}$ **then**
      ROOT ← INNER CHILD
   **else**
      ROOT ← OUTER CHILD
   **end if**
**end while**
**return** best_q
```

Algorithm 4: FindRoughNN(P, q, t, NODE ROOT)

$\sqrt{D_{s\phi}}(\text{nn}_q, q)$ be the exact nearest neighbor distance. Finally, let **curr** be the tree node currently being examined by our algorithm, and rep_{curr} be a representative point $p \in P$ of **curr**. By convention r_v represents the radius of the *inner* $\sqrt{D_{s\phi}}$-ball associated with a node **v**, and within each node **v** we store $\text{rep}_v = m_v$, which is the center of $B_{\text{in}}(m_v, r_v)$. The node associated with the inner $\sqrt{D_{s\phi}}$-ball B_{in} is denoted by $\mathbf{v}_{\text{in}}$ and the node associated with B_{out} is denoted by $\mathbf{v}_{\text{out}}$.

Lemma 2.4.6 *Given a t-ring-tree T for a point set with respect to $\sqrt{D_{s\phi}}$ in a μ-defective domain, where $t \leq \frac{1}{\log n}$ and query point q we can find a $O(\mu + \frac{2\mu^2}{t})$ nearest neighbor to q in $O((\mu+1)^d d^{\frac{d}{2}} \log n)$ time.*

Proof. Our search algorithm is a binary tree search. Whenever we reach node **v**, if $D(\text{rep}_v, q) < D_{\text{near}}$ set $\text{best}_q = \text{rep}_v$ and $D_{\text{near}} = \sqrt{D_{s\phi}}(\text{rep}_v, q)$ as our current nearest neighbor and nearest neighbor distance, respectively. Our branching criterion is that if $\sqrt{D_{s\phi}}(\text{rep}_v, q) < (1 + \frac{t}{2})r_v$, we continue search in $\mathbf{v}_{\text{in}}$, else we continue the search in $\mathbf{v}_{\text{out}}$. Since the depth of the tree is $O(\log n)$ by lemma 2.4.4, this process will take $O(\log n)$ time.

Turning now to quality, let $\mathbf{w}$ be the first node such that $\mathrm{nn}_q \in \mathbf{w}_{\mathrm{in}}$ but we searched in $\mathbf{w}_{\mathrm{out}}$, or vice-versa. After examining rep_w, $D_{\mathrm{near}} \leq \sqrt{D_{s\phi}}(\mathrm{rep}_w, q)$ and D_{near} can only decrease at each step. An upper bound on $\sqrt{D_{s\phi}}(q, \mathrm{rep}_w)/\sqrt{D_{s\phi}}(q, \mathrm{nn}_q)$ yields a bound on the quality of the approximate nearest neighbor produced. In the first case, suppose $\mathrm{nn}_q \in \mathbf{w}_{\mathrm{in}}$, but we searched in $\mathbf{w}_{\mathrm{out}}$. Then $\sqrt{D_{s\phi}}(\mathrm{rep}_w, q) > \left(1+\frac{t}{2}\right) r_w$ and $\sqrt{D_{s\phi}}(\mathrm{rep}_w, \mathrm{nn}_q) < r_w$. Now μ-defectiveness implies that

$$\begin{aligned}
\mu\sqrt{D_{s\phi}}(q, \mathrm{nn}_q) &> \sqrt{D_{s\phi}}(\mathrm{rep}_w, q) - \sqrt{D_{s\phi}}(\mathrm{rep}_w, \mathrm{nn}_q)\\
\mu\sqrt{D_{s\phi}}(q, \mathrm{nn}_q) &> \left(1+\frac{t}{2}\right) r_w - r_w\\
\sqrt{D_{s\phi}}(q, \mathrm{nn}_q) &> \frac{t}{2\mu} r_w.
\end{aligned}$$

And for the upper bound on $\sqrt{D_{s\phi}}(\mathrm{rep}_w, q)/\sqrt{D_{s\phi}}(q, \mathrm{nn}_q)$, we again apply μ-defectiveness to conclude that $\sqrt{D_{s\phi}}(\mathrm{rep}_w, q) - \sqrt{D_{s\phi}}(q, \mathrm{nn}_q) < \mu\sqrt{D_{s\phi}}(\mathrm{nn}_q, \mathrm{rep}_w)$, which yields

$$\begin{aligned}
\frac{\sqrt{D_{s\phi}}(\mathrm{rep}_w, q)}{\sqrt{D_{s\phi}}(q, \mathrm{nn}_q)} &< 1 + \mu\frac{r_w}{\sqrt{D_{s\phi}}(q, \mathrm{nn}_q)}\\
&< 1 + \mu\frac{r_w}{\frac{t}{2\mu} r_w}\\
&= 1 + 2\frac{\mu^2}{t}.
\end{aligned}$$

We now consider the other case. Suppose $\mathrm{nn}_q \in \mathbf{w}_{\mathrm{out}}$ and we search in $\mathbf{w}_{\mathrm{in}}$ instead. By construction we must have $\sqrt{D_{s\phi}}(\mathrm{rep}_w, q) < \left(1+\frac{t}{2}\right) r_w$ and $\sqrt{D_{s\phi}}(\mathrm{rep}_w, \mathrm{nn}_q) > (1+t) r_w$. Again, μ-defectiveness yields $\sqrt{D_{s\phi}}(q, \mathrm{nn}_q) > \frac{t}{2\mu} r_w$. Now we can simply take the ratios of the two: $\frac{\sqrt{D_{s\phi}}(\mathrm{rep}_w, q)}{\sqrt{D_{s\phi}}(q, \mathrm{nn}_q)} < \frac{(1+\frac{t}{2}) r_w}{\frac{t}{2\mu} r_w} = \mu + \frac{2\mu}{t}$. Taking an upper bound of the approximation provided by each case, the ring-tree provides us a $\mu + 2\frac{\mu^2}{t}$ approximation. The space/running time bound follows from lemma 2.4.5, and noting that taking a thinner ring ($t \leq \frac{1}{\log n}$) in the proof there only decreases the depth of the tree due to lesser duplication of points. ■

Corollary 2.4.2 *Setting $t = \frac{1}{\log n}$, given a point set with respect to $\sqrt{D_{s\phi}}$ in a μ-defective domain we can find a $O(\mu + 2\mu^2 \log n)$ approximate nearest neighbor to a query point q in*

$O(d^{\frac{d}{2}}(\mu+1)^d\log(n))$ time, using a $O(\frac{1}{\log n})$ ring separator tree constructed in $O(d^{\frac{d}{2}}(\mu+1)^d n\log(n))$ expected time.

Proof. The query time is bounded by the depth of the tree, which is bounded in lemma 2.4.4. That we can construct a ring of our desired thickness at each step in reasonable expected time is guaranteed by 2.4.3. The space guarantee comes from lemma 2.4.5 and the quality of nearest neighbor obtained from our ring-tree analyzed by lemma 2.4.6. Note that we are slightly abusing notation in lemma 2.4.3, in that the separating ring obtained there and which we use is not empty of points of P as originally stipulated. However remember that if nn_q is in the ring, then nn_q repeats in *both* children and cannot fall off the search path. Hence we can "pretend" the ring is empty as in our analysis in lemma 2.4.6.

■

2.5 Computing a $1+\varepsilon$ approximation

We give our overall algorithm for obtaining a $1+\varepsilon$ nearest neighbor in $O\left(\frac{1}{\varepsilon^d}\log^{2d} n\right)$ query time under $\sqrt{D_{s\phi}}$. We note that although our bounds are for $\sqrt{D_{s\phi}}$, similar bounds follow in the same manner for any decomposable symmetric distance measure D, which satisfies an RTI and for which the ratio of diameter to side length of a cube is bounded by $O(d^{O(1)})$.

2.5.1 Preprocessing

We first construct an improved ring-tree R on our point set P in $O(n\log n)$ time as described in lemma 2.4.5, with ring thickness $O(\frac{1}{\log n})$. We then compute an efficient orthogonal range reporting data structure on P in $O(n\log^{d-1} n)$ time, such as that described in (5) by Afshani *et al.*We note the main result we need:

Lemma 2.5.1 *We can compute a data structure from P with $O(n\log^{d-1} n)$ storage (and same construction time), such that given an arbitrary axis parallel box H we can determine in $O(\log^d n)$ query time a point $p \in P\cap H$ if $|P\cap H| > 0$.*

2.5.2 Query handling

Given a query point q, we use R to obtain a point q_{rough} in $O(\log n)$ time such that $D_{\text{rough}} = \sqrt{D_{s\phi}}(q, q_{\text{rough}}) \leq (1+\mu^2\log n)\sqrt{D_{s\phi}}(q, \mathrm{nn}_q)$. Given q_{rough}, we can use lemma 2.3.8 to find a family F of 2^d $\sqrt{D_{s\phi}}$-cubes of side-length exactly D_{rough} such that they cover the $\sqrt{D_{s\phi}}$-ball $B(q, D_{\text{rough}})$. We use our range reporting structure to find a point $p \in P$ for all nonempty cubes in F in a total of $2^d \log^d n$ time. These points act as representatives of the $\sqrt{D_{s\phi}}$-cubes for what follows. Note that nn_q must necessarily be in one of these $\sqrt{D_{s\phi}}$-cubes, and hence there must be a $(1+\varepsilon)$-nearest neighbor $q_{\text{approx}} \in P$ in some $G \in F$. To locate this q_{approx}, we construct a quadtree (139, Chapter 11) (64) for repeated bisection and search on each $G \in F$.

Algorithm 5 describes the overall procedure. We call the collection of all cells produced during the procedure a *quadtree*. We borrow the presentation in Har-Peled's book (139) with the important qualifier that we construct our quadtree at runtime. The terminology here is as introduced earlier in Section 2.4.

```
Instantiate a queue $Q$ containing all cells from $F$ along with their representatives and
enqueue **root**.
Let $D_{\text{near}} = \sqrt{D_{s\phi}}(\text{rep}_{\text{root}}, q)$, $\text{best}_q = \text{rep}_{\textbf{root}}$
**repeat**
  Pull off the head of the queue and place it in **curr**.
  **if** $\sqrt{D_{s\phi}}(\text{rep}_{\text{curr}}, q) < \sqrt{D_{s\phi}}(\text{best}_q, q)$ **then**
    Let $\text{best}_q = \text{rep}_{\text{curr}}$, $D_{\text{near}} = \sqrt{D_{s\phi}}(\text{best}_q, q)$
    Bisect **curr** according to procedure of lemma 2.5.3; denote the result as $\{G_i\}$.
    **for all** $G_i$ **do**
      As described in 2.5.3, check if $G_i$ is nonempty by passing it to our range
      reporting structure, which will also return us some $p \in P$ if $G_i$ is not empty.
      Also check if $G_i$ may contain a point closer than $(1 - \frac{\varepsilon}{2})D_{\text{near}}$ to $q$. (This may
      be done in $O(d)$ time for each cell, given the coordinates of the corners.)
      **if** $G_i$ is nonempty AND has a close enough point to $q$ **then**
        Let $\text{rep}_{G_i} = p$
        Enqueue $G_i$
      **end if**
    **end for**
  **end if**
**until** $Q$ is empty
Return $\text{best}_q$
```

Algorithm 5: QueryApproxNN(P, **root**, q)

Lemma 2.5.2 *Algorithm 5 will always return a $(1+\varepsilon)$-approximate nearest neighbor.*

Proof. Let best_q be the point returned by the algorithm at the end of execution. By the method of the algorithm, for all points p for which the distance is directly evaluated, we have that $\sqrt{D_{s\phi}}(\text{best}_q,q) < \sqrt{D_{s\phi}}(p,q)$. The terminology here is as in Section 2.4. We look at points p which are *not* evaluated during the running of the algorithm, i.e., we did not expand their containing cells. But by the criterion of the algorithm for not expanding a cell, it must be that $\sqrt{D_{s\phi}}(\text{best}_q,q)(1-\frac{\varepsilon}{2}) < \sqrt{D_{s\phi}}(p,q)$. For $\varepsilon < 1$, this means that $(1+\varepsilon)\sqrt{D_{s\phi}}(p,q) > \sqrt{D_{s\phi}}(\text{best}_q,q)$ for any $p \in P$, including nn_q. So best_q is indeed a $1+\varepsilon$ approximate nearest neighbor. ■

We must analyze the time complexity of a single iteration of our algorithm, namely the complexity of a subdivision of a $\sqrt{D_{s\phi}}$-box G and determining which of the 2^d $\sqrt{D_{s\phi}}$-subcells of G are nonempty.

Lemma 2.5.3 *Let G be a $\sqrt{D_{s\phi}}$-box with maximum $\sqrt{D_{s\phi}}$-side-length s and G_i its subcells produced by bisecting along each side of G under $\sqrt{D_{s\phi}}$. For all nonempty $\sqrt{D_{s\phi}}$-subcubes G_i of G, we can find $p_i \in P \cap G_i$ in $O(2^d \log^d n)$ total time complexity, and the maximum $\sqrt{D_{s\phi}}$-side-length of any G_i is at most $\frac{s}{2}$.*

Proof. Note that G is defined as a product of d intervals. For each interval, we can find an approximate bisecting point under $\sqrt{D_{s\phi}}$ in $O(1)$ time and by the RTI each subinterval is of length at most $\frac{s}{2}$ under $\sqrt{D_{s\phi}}$. This leads to an $O(d)$ cost to find a bisection point for all intervals, which define $O(2^d)$ $\sqrt{D_{s\phi}}$-subboxes or children of G. We pass each $\sqrt{D_{s\phi}}$-subbox of G to our range reporting structure. By lemma 2.5.1, this takes $O(\log^d n)$ time to check emptiness or return a point $p_i \in P$ contained in the child, if nonempty. Since there are $O(2^d)$ nonempty children of G, this implies a cost of $2^d(\log^d n)$ time incurred. Checking each of the nonempty subboxes G_i to see if it may contain a point closer than $(1-\frac{\varepsilon}{2})D_{\text{near}}$ to q takes a further $O(d)$ time per cell or $O(d2^d)$ time. ■

We now bound the number of cells that will be added to our search queue. We do so indirectly, by placing a lower bound on the maximum $\sqrt{D_{s\phi}}$-side-length of all such cells.

Lemma 2.5.4 *Algorithm 5 will not add the children of node **C** to our search queue if the maximum side-length of **C** is less than* $\dfrac{\varepsilon\sqrt{D_{s\phi}}(q,nn_q)}{2\mu\sqrt{d}}$.

Proof. Let $\Delta(\mathbf{C})$ represent the $\sqrt{D_{s\phi}}$-diameter of cell $\mathbf{C}$. By construction, we can expand

C only if some subcell of **C** contains a point p such that $\sqrt{D_{s\phi}}(p,q) \leq (1-\frac{\varepsilon}{2})D_{\text{near}}$. Note that since **C** is examined, we have $D_{\text{near}} \leq \sqrt{D_{s\phi}}(\text{rep}_C,q)$. Now assuming we expand **C**, then we must have:

$$\mu\Delta(\mathbf{C}) > \sqrt{D_{s\phi}}(\text{rep}_C,q) - \sqrt{D_{s\phi}}(p,q) \geq D_{\text{near}} - (1-\frac{\varepsilon}{2})D_{\text{near}} = \frac{\varepsilon}{2}D_{\text{near}} \quad (2.10)$$

So $\varepsilon/(2\mu)D_{\text{near}} < \Delta(\mathbf{C})$. First note $\sqrt{D_{s\phi}}(\text{rep}_C,q) < D_{\text{near}}$. Also, by definition, $\sqrt{D_{s\phi}}(q,\text{nn}_q) < D_{\text{near}}$. And $\Delta(\mathbf{C}) < \sqrt{d}s$ by lemma 2.3.7 where s is the maximum side-length of **C**. Making the appropriate substitutions yields us our required bound. ∎

Given the bound on quadtree depth (lemma 2.5.4), and using the fact that at most 2^{xd} nodes are expanded at level x, we have:

Lemma 2.5.5 *Given a* $\sqrt{D_{s\phi}}$*-cube* G *of* $\sqrt{D_{s\phi}}$*-side-length* D_{rough}*, we can compute a* $(1+\varepsilon)$*-nearest neighbor to* q *in* $O\left(\frac{1}{\varepsilon^d}2^d\mu^d d^{\frac{d}{2}}\left(\frac{D_{rough}}{\sqrt{D_{s\phi}}(q,nn_q)}\right)^d \log^d n\right)$ *time.*

Proof. Consider a quadtree search from q on a $\sqrt{D_{s\phi}}$-cube G of $\sqrt{D_{s\phi}}$-side-length D_{rough}. By lemma 2.5.4, our algorithm will not expand cells with all $\sqrt{D_{s\phi}}$-side-lengths smaller than $\frac{\varepsilon\sqrt{D_{s\phi}}(q,\text{nn}_q)}{2\mu\sqrt{d}}$. But since the $\sqrt{D_{s\phi}}$-side-length reduces by at least half in each dimension upon each split, all $\sqrt{D_{s\phi}}$-side-lengths are less than this value after $x = \log\left(D_{\text{rough}}/\frac{\varepsilon\sqrt{D_{s\phi}}(q,\text{nn}_q)}{2\mu\sqrt{d}}\right)$ repeated bisections of our root cube.

Noting that $O(\log^d n)$ time is spent at each node by lemma 2.5.3, and that at the xth level the number of nodes expanded is 2^{xd}, we get a final time complexity bound of $O\left(\frac{1}{\varepsilon^d}2^d\mu^d d^{\frac{d}{2}}\left(\frac{D_{\text{rough}}}{\sqrt{D_{s\phi}}(q,\text{nn}_q)}\right)^d \log^d n\right)$. ∎

Substituting $D_{\text{rough}} = \mu^2 \log n\sqrt{D_{s\phi}}(q,\text{nn}_q)$ in lemma 2.5.5 gives us a bound of $O\left(2^d\frac{1}{\varepsilon^d}\mu^{3d}d^{\frac{d}{2}}\log^{2d} n\right)$. This time is per $\sqrt{D_{s\phi}}$-cube of F that covers $B(q,D_{\text{rough}})$. Noting that there are 2^d such $\sqrt{D_{s\phi}}$-cubes gives us a final time complexity of $O\left(2^{2d}\frac{1}{\varepsilon^d}\mu^{3d}d^{\frac{d}{2}}\log^{2d} n\right)$. For the space complexity of our run-time queue, observe that the number of nodes in our queue increases only if a node has more than one nonempty child, i.e., there is a split of our n points. Since our point set may only split n times, this gives us a bound of $O(n)$ on the space complexity of our queue.

2.6 Logarithmic bounds, with further assumptions

For a given $D_{s\phi}$, let $c_0 = \max_{i\in[1..d]} \sqrt{\frac{\max_x \phi_i''(x)}{\min_y \phi_i''(y)}}$ over our bounded subset of the domain (c_0 may be infinity over the unrestricted domain, or on a subset over whose closure ϕ'' tends to infinity or zero). c_0 is susceptible to the choice of bounded subset of the domain and in general grows as we expand our allowed subset. We conjecture that $c_0 = \Theta(\mu)$ although we cannot prove it. In particular, we show that if we assume a bounded c_0 (in addition to μ), we can obtain a $1+\varepsilon$ nearest neighbor in time $O(\log n + (\frac{1}{\varepsilon})^d)$ time for $\sqrt{D_{s\phi}}$. We do so by constructing a *Euclidean* quadtree T on our set in preproccessing and using c_0 and μ to express the bounds obtained in terms of $\sqrt{D_{s\phi}}$.

We will refer to the Euclidean distance l_2 as D_e and note first the following key relation between $\sqrt{D_{s\phi}}$ and D_e, where c_0 serves to relate the two measures by some constant factor. Nock *et al.* (121) use a comparable measure to c_0 as do Sra *et al.* (144), for similar purposes of establishing a constant factor approximation to the Euclidean distance.

Lemma 2.6.1 *Suppose we are given an interval $I = [x_1 x_2] \subset \mathbb{R}$ s.t. $x_1 < x_2$, $De(x_1,x_2) = r_e$, and $\sqrt{D_{s\phi}(x_1,x_2)} = r_\phi$. Suppose we divide I into m subintervals of equal length with endpoints $x_1 = a_0, a_1, \ldots a_{m-1}, a_m = x_2$, where $a_i < a_{i+1}$ and $De(a_i,a_{i+1}) = r_e/m$, $\forall i \in [0..m-1]$. Then $\frac{r_\phi}{c_0 m} \leq \sqrt{D_{s\phi}(a_i,a_{i+1})} \leq \frac{c_0 r_\phi}{m}$.*

Proof. We can relate $\sqrt{D_{s\phi}}$ to D_e via the Taylor expansion of $\sqrt{D_{s\phi}}$: $\sqrt{D_{s\phi}(a,b)} = \sqrt{\phi''(\bar{x})}De(a,b)$ for some $\bar{x} \in [ab]$. Combining this with c_0 yields

$$\frac{\min_i \sqrt{D_{s\phi}(a_i,a_{i+1})}}{\sqrt{D_{s\phi}(x_1,x_2)}} \geq \frac{De(a_i,a_{i+1})}{c_0 De(x_1,x_2)} = \frac{1}{c_0 m} \tag{2.11}$$

and

$$\frac{\max_i \sqrt{D_{s\phi}(a_i,a_{i+1})}}{\sqrt{D_{s\phi}(x_1,x_2)}} \leq c_0 \frac{De(a_i,a_{i+1})}{De(x_1,x_2)} = \frac{c_0}{m}. \tag{2.12}$$

■

Corollary 2.6.1 *If we recursively bisect an interval $I = [x_1 x_2] \subset \mathbb{R}$ s.t. $D_e(x_1,x_2) = r_e$ and $\sqrt{D_{s\phi}(x_1,x_2)} = r_\phi$ into 2^i equal subintervals (under D_e), then $\frac{r_\phi}{c_0 2^i} \leq \sqrt{D_{s\phi}(a_k,a_{k+1})} \leq \frac{c_0 r_\phi}{2^i}$ for any of the subintervals $[a_k a_{k+1}]$ so obtained. Hence after $\log\frac{c_0 r_\phi}{x}$ subdivisions, all intervals will be of length at most x under $\sqrt{D_{s\phi}}$. Also, given a cube of initial side-length r_ϕ, after $\log\frac{c_0 r_\phi}{x}$ repeated bisections (under D_e) the diameter will be at most $\sqrt{d}x$ under $\sqrt{D_{s\phi}}$.*

We find the smallest enclosing $\sqrt{D_{s\phi}}$-cube C that bounds our point set, and then construct our compressed Euclidean quadtree in preprocessing on this cube. Say C is of side-length s. Corollary 2.6.1 gives us that for cells formed at the i-th level of decomposition, the side-length under $\sqrt{D_{s\phi}}$ is between $\frac{s}{c_0 2^i}$ and $\frac{c_0 s}{2^i}$. Refer to these cells formed at the i-th level as L_i.

Lemma 2.6.2 *Given a $\sqrt{D_{s\phi}}$-ball B of radius r, let $i = \log\frac{s}{c_0 r}$. Then $|L_i \cap B| \leq O(2^d)$ and the side-length of each cell in L_i is between r and $c_0{}^2 r$ under $\sqrt{D_{s\phi}}$. We can also explicitly retrieve the quadtree cells corresponding to $|L_i \cap B|$ in $O(2^d \log n)$ time.*

Proof. Note that for cells in L_i, we have side-lengths under $\sqrt{D_{s\phi}}$ between $\frac{s}{c_0 2^i}$ and $\frac{c_0 s}{2^i}$ by Corollary 2.6.1. Substituting $i = \log\frac{s}{c_0 r}$, these cells have side-length between r and $c_0{}^2 r$ under $\sqrt{D_{s\phi}}$. By the reverse triangle inequality and lemma 2.3.1, we get our required bound for $|L_i \cap B|$. In preconstruction of our quadtree T we maintain for each dimension the corresponding interval quadtree T_k, $\forall k \in [1..d]$. Observe this incurs at most $O(n)$ storage, with d in the big-Oh. For retrieving the actual cells $|L_i \cap B|$, we first find the $O(1)$ intervals from level i in each T_k that may intersect B. Taking a product of these, we get $O(2^d)$ cells which are a superset of the canonical cells $L_i \subset T$. Each cell may be looked up in $O(\log n)$ time from the compressed quadtree (139) so our overall retrieval time is $O(2^d \log n)$. ■

Given query point q, we first obtain in $O(\log n)$ time with our ring-tree a rough $O(n)$ ANN q_{rough} under $\sqrt{D_{s\phi}}$ s.t. $D_{\text{rough}} = \sqrt{D_{s\phi}(q,q_{\text{rough}})} = \mu^2 n \sqrt{D_{s\phi}(q,\text{nn}_q)}$. Note that we can actually obtain a $O(\log n)$-ANN instead, using the results of Section 2.4.4. But a coarser approximation of $O(n)$-ANN suffices here for our bound. The tree depth (and implicitly the storage and running time) is still bounded by the $O(d^{\frac{d}{2}}(\mu+1)^d \log n)$

of lemma 2.4.4, since in using thinner rings we have less point duplication and the same proportional reduction in number of points in each node at each level.

Now lemma 2.6.2, we have $O(2^d)$ quadtree cells intersecting $B(q, \sqrt{D_{s\phi}(q, q_{\text{rough}})})$.

Let us call this collection of cells Q. We then carry out a quadtree search on each element of Q. Note that we expand only cells which may contain a point nearer to query point q than the current best candidate. We bound the depth of our search using μ-defectiveness similar to lemma 2.5.4:

Lemma 2.6.3 *We will not expand cells of* $\sqrt{D_{s\phi}}$*-diameter less than* $\frac{\varepsilon\sqrt{D_{s\phi}(q,nn_q)}}{2\mu}$ *or cells whose all side-lengths w.r.t.* $\sqrt{D_{s\phi}}$ *are less than* $\frac{\varepsilon\sqrt{D_{s\phi}(q,nn_q)}}{2\mu\sqrt{d}}$.

For what follows, refer to our *spread* as $\beta = \frac{D_{\text{rough}}}{\sqrt{D_{s\phi}(q,\text{nn}_q)}}$.

Lemma 2.6.4 *We will only expand our tree to a depth of* $k = \log(2{c_0}^3\mu\beta\sqrt{d}/\varepsilon)$.

Proof. Using lemma 2.6.3 and Corollary 2.6.1, each cell of Q will be expanded only to a depth of $k = \log\left(c_0{c_0}^2 D_{\text{rough}} / \frac{\varepsilon\sqrt{D_{s\phi}(q,\text{nn}_q)}}{2\mu\sqrt{d}}\right)$.

This gives us a depth of $\log(2{c_0}^3\mu\beta\sqrt{d}/\varepsilon)$. ■

Lemma 2.6.5 *The number of cells examined at the i-th level is*

$$n_i < 2^d\left(\mu^d d^{\frac{d}{2}} c_0^{4d} + (\frac{2^i c_0}{\beta})^d\right).$$

Proof. Recalling that the cells of Q start with side-length at most $c_0^2 D_{\text{rough}}$ under $\sqrt{D_{s\phi}}$, at the i-th level the $\sqrt{D_{s\phi}}$-diameter of cells is at most $\frac{c_0^3\sqrt{d}D_{\text{rough}}}{2^i}$, by Corollary 2.6.1. Hence by μ-defectiveness, there must be some point examined by our algorithm at $\sqrt{D_{s\phi}}$-distance at most $D_{\text{best}} = \sqrt{D_{s\phi}}(q, \text{nn}_q) + \frac{\mu c_0^3\sqrt{d}D_{\text{rough}}}{2^i}$. Note that our algorithm will only expand cells within this distance of q.

The $\sqrt{D_{s\phi}}$ side-length of a cell $\mathbf{C}$ at this level is at least $\Delta(\mathbf{C}) = \frac{D_{\text{rough}}}{c_0 2^i}$. Applying the packing bounds from lemma 2.3.3, and the fact that $(a+b)^d < 2^d(a^d + b^d)$, the number of cells expanded is at most

$$n_i = \left(\frac{D_{\text{best}}}{\Delta(\mathbf{C})}\right)^d < 2^d \left(\mu^d d^{\frac{d}{2}} c_0^{4d} + \left(\frac{c_0 2^i}{\beta}\right)^d\right).$$

■

Finally we add the n_i to get the total number of nodes explored,

$$\sum_i n_i = O\left(2^d \mu^d d^{\frac{d}{2}} c_0^{4d} \log(2c_0{}^3 \mu\beta\sqrt{d}/\varepsilon) + 2^{2d} c_0^{4d} \mu^d d^{\frac{d}{2}} / \varepsilon^d\right).$$

Recalling that $\beta = \frac{D_{\text{rough}}}{\sqrt{D_{s\phi}(q,\text{nn}_q)}} = \mu^2 n$, substituting back and ignoring lower order terms, the time complexity is

$$O\left(2^d \mu^d d^{\frac{d}{2}} c_0^{4d} \log n + 2^{2d} c_0^{4d} \mu^d d^{\frac{d}{2}} / \varepsilon^d\right).$$

Accounting for the 2^d cells in Q that we need to search, this adds a further 2^d multiplicative factor. This time complexity of this quadtree phase (number of cells explored) of our algorithm dominates the time complexity of the ring-tree search phase of our algorithm, and hence is our overall time complexity for finding a $(1+\varepsilon)$ ANN to q. For space and preconstruction time, we note that compressed Euclidean quadtrees can be built in $O(n\log n)$ time and require $O(n)$ space (139), which matches our bound for the ring-tree construction phase of our algorithm requiring $O(n\log n)$ time and $O(n)$ space.

2.7 The general case: Asymmetric divergences

Without loss of generality we will focus on the *right-sided* nearest neighbor: given a point set P, query point q and $\varepsilon \geq 0$, find $x \in P$ that approximates $\min_{p\in P} D(p,q)$ to within a factor of $(1+\varepsilon)$. Since a Bregman divergence is not in general μ-defective, we will consider instead $\sqrt{D_\phi}$: by monotonicity and with an appropriate choice of ε, the result will carry over to D_ϕ.

We list three issues that have to be resolved to complete the algorithm. Firstly, because of asymmetry, we cannot bound the diameter of a quadtree cell $\mathbf{C}$ of side-length s by $s\sqrt{d}$. However, as the proof of lemma 2.3.6 shows, we can choose a *canonical corner* of a cell such that a (directed) ball of radius $s\sqrt{d}$ centered at that corner covers the cell. By μ-defectiveness, we can now conclude (see lemma 2.7.7) that the diameter of $\mathbf{C}$ is at most

$(\mu+1)s\sqrt{d}$ (note that this incurs an extra factor of $\mu+1$ in all expressions). Secondly, since while $\sqrt{D_\phi}$ satisfies μ-defectiveness (unlike D_ϕ) the opposite is true for the reverse triangle inequality, which is satisfied by D_ϕ but not $\sqrt{D_\phi}$. This requires the use of a weaker packing bound based on lemma 2.3.2, introducing dependence in $1/\varepsilon^2$ instead of $1/\varepsilon$. And thirdly, the lack of symmetry means we have to be careful of the use of directionality when proving our bounds. Perhaps surprisingly, the major part of the arguments carry through simply by being consistent in the choice of directionality.

Note that for this section we are referring to $\sqrt{D_\phi}$. With some small adjustments, similar bounds can be obtained for more generic asymmetric, monotone, decomposable and μ-defective distance D measures satisfying packing bounds. The left-sided asymmetric nearest neighbor can be determined analogously.

Finally, given a bounded domain M, we have that $\sqrt{D_\phi}$ is left-sided μ-defective for some μ_L and right sided μ-defective for some μ_R (see lemma 2.2.6 for detailed proof). For what follows, let $\mu = \max(\mu_L, \mu_R)$ and describe M as simply μ-defective.

Most of the proofs here mirror their counterparts in Sections 2.4 and 2.5.

2.7.1 Asymmetric ring-trees

Since we focus on *right*-near-neighbors, all balls and ring separators referred to will use *left-balls* i.e., balls $B(m,r) = \{x \mid D(m,x) < r\}$. As in Section 2.4, we will design a ring-separator algorithm and use that to build a ring-separator tree.

Lemma 2.7.1 *Let D be a μ-defective distance, and let $B(m,r)$ be a left-ball with respect to D. Then for any two points $x,y \in B(m,r)$, $D(x,y) < (\mu+1)r$.*

Proof. Follows from the definition of right sided μ-defectiveness,

$$D(x,y) - D(m,y) < \mu D(m,x)$$

$$D(x,y) < \mu r + D(m,y) = (\mu+1)r.$$

■

Corollary 2.7.1 *For any $\sqrt{D_\phi}$-left-ball $B(m,r)$ and two points $x,y \in B(m,r)$, $\sqrt{D_\phi}(x,y) < (\mu+1)r$.*

Proof. Since $\sqrt{D_\phi}$ is μ-defective over a prespecified restricted domain. ■

As in lemma 2.4.2 we can construct (in $O(nc)$ expected time) a $(\mu+1)$-approximate $\sqrt{D_\phi}$-left-ball enclosing $\frac{n}{c}$ points. This in turn yields a ring-separator construction, and from it a ring-tree with an extra $(\mu+1)^d d^{\frac{d}{2}}$ factor in depth as compared to symmetric ring-trees, due to the weaker packing bounds for $\sqrt{D_\phi}$.

We note that the asymptotic bounds for ring-tree storage and construction time follow from purely combinatorial arguments and hence are unchanged for $\sqrt{D_\phi}$. Once we have the ring-tree, we can use it as before to identify a rough near-neighbor for a query q; once again, exploiting μ-defectiveness gives us the desired approximation guarantee for the result.

Lemma 2.7.2 *Given any parameter $1 \le c \le n$, we can compute in $O(nc)$ randomized time a $\sqrt{D_\phi}$-left-ball $B(m,r')$ such that $r' \le (\mu+1)r_{opt,c}$ and $B(m,r') \cap P \ge \frac{n}{c}$.*

Proof. Follows identically to the proof of lemma 2.4.2. ■

Lemma 2.7.3 *There exists a parameter c (which depends only on d and μ), such that for any d-dimensional point set P and any μ-defective $\sqrt{D_\phi}$, we can find a $O(\frac{1}{\log n})$ left-ring separator $R_{P,c}$ in $O(n)$ expected time.*

Proof. First, using our randomized construction, we compute a $\sqrt{D_\phi}$-left-ball $S = B(m,r_1)$ (where $m \in P$) containing $\frac{n}{c}$ points such that $r_1 \le (\mu+1)r_{\text{opt},c}$, where c is a parameter to be set. Consider the $\sqrt{D_\phi}$-left-ball $\bar{S} = B(m,2r_1)$. As described in lemma 2.3.9, $\bar{S}$ can be covered by 2^d $\sqrt{D_\phi}$-hypercubes of side-length $2r_1$, the union of which we shall refer to as H. Set $L = (\mu+1)\sqrt{d}$. Imagine a partition of H into a grid, where each cell is of side-length $\frac{r_1}{L}$. Each cell in this grid can be covered by a $\sqrt{D_\phi}$-ball of radius $\Delta(\frac{r_1}{L},d) = \frac{r_1}{\mu+1} \le r_{\text{opt},c}$ centered on it's lowest corner. This implies any cell will contain at most $\frac{n}{c}$ points, by the definition of $r_{\text{opt},c}$.

By lemma 2.3.4 the grid on H has at most $2^d(2r_1/\frac{r_1}{L})^{2d} = (4(\mu+1)\sqrt{d})^{2d}$ cells. Each cell may contain at most $\frac{n}{c}$ points. In particular, set $c = 2(4(\mu+1)\sqrt{d})^{2d}$. Then we have that H may contain at most $\frac{n}{c}(4(\mu+1)\sqrt{d})^{2d} = \frac{n}{2}$ points, or since $\bar{S} \subset H$, $\bar{S}$ contains at most $\frac{n}{2}$ points and $\bar{S}'$ contains at least $\frac{n}{2}$ points. The rest of the proof goes through as in lemma 2.4.3. ■

We proceed now to the construction of our ring-tree using the basic ring-separator structure of lemma 2.7.3.

Lemma 2.7.4 *Given any point set P, we can construct a $O(\frac{1}{\log n})$ left ring-separator tree T under $\sqrt{D_\phi}$ of depth $O(d^d(\mu+1)^{2d}\log n)$.*

Proof. Repeatedly partition P by lemma 2.7.3 into P^v_{in} and P^v_{out} where **v** is the parent node. Store only the single point $\text{rep}_v = m \in P$ in node **v**, the center of the ball $B(m, r_1)$. We continue this partitioning until we have nodes with only a single point contained in them.

Since each child contains at least $\frac{n}{c}$ points, each subset reduces by a factor of at least $1-\frac{1}{c}$ at each step, and hence the depth of the tree is logarithmic. We calculate the depth more exactly, noting that in lemma 2.7.3, $c = O(d^d(\mu+1)^{2d})$. Hence the depth x can be bounded as

$$\begin{aligned}
&n(1-\frac{1}{c})^x = 1\\
&(1-\frac{1}{c})^x = \frac{1}{n}\\
&x = \frac{\ln\frac{1}{n}}{\ln(1-\frac{1}{c})} = \frac{-1}{\ln(1-\frac{1}{c})}\ln n\\
&x \le c\ln n = O\left(d^d(\mu+1)^{2d}\log n\right).
\end{aligned}$$

■

Note that lemma 2.7.4 also serves to bound the query time of our data structure. We need only now bound the approximation quality. The derivation is similar to lemma 2.4.6, but with some care about directionality.

Lemma 2.7.5 *Given a t-ring-tree T for a point set with respect to a μ-defective $\sqrt{D_\phi}$, where $t \le \frac{1}{\log n}$, and query point q we can find a $O(\mu + \frac{2\mu^2}{t})$ nearest neighbor to query point q in $O((\mu+1)^{2d}d^d\log n)$ time.*

Proof. Our search algorithm is a binary tree search. Whenever we reach node **v**, if $\sqrt{D_\phi}(\text{rep}_v, q) < D_{\text{near}}$ set $\text{best}_q = \text{rep}_v$ and $D_{\text{near}} = \sqrt{D_\phi}(\text{rep}_v, q)$ as our current nearest neighbor and nearest neighbor distance, respectively. Our branching criterion is that if

$\sqrt{D_\phi}(\text{rep}_v, q) < (1+\frac{t}{2})r_v$, we continue search in $\mathbf{v}_{\text{in}}$, else we continue the search in $\mathbf{v}_{\text{out}}$. Since the depth of the tree is $O(\log n)$ by lemma 2.7.4, this process will take $O(\log n)$ time.

Let $\mathbf{w}$ be the first node such that $\text{nn}_q \in \mathbf{w}_{\text{in}}$ but we searched in $\mathbf{w}_{\text{out}}$, or vice-versa. The analysis goes by cases. In the first case as seen in Figure 2.4, suppose $\text{nn}_q \in \mathbf{w}_{\text{in}}$, but we searched in $\mathbf{w}_{\text{out}}$. Then

$$\sqrt{D_\phi}(\text{rep}_w, q) > \left(1+\frac{t}{2}\right) r_w$$
$$\sqrt{D_\phi}(\text{rep}_w, \text{nn}_q) < r_w.$$

Now left-sided μ-defectiveness implies a lower bound on the value of $\sqrt{D_\phi}(\text{nn}_q, q)$:

$$\mu\sqrt{D_\phi}(\text{nn}_q, q) > \sqrt{D_\phi}(\text{rep}_w, q) - \sqrt{D_\phi}(\text{rep}_w, \text{nn}_q)$$
$$\mu\sqrt{D_\phi}(\text{nn}_q, q) > \left(1+\frac{t}{2}\right) r_w - r_w$$
$$\sqrt{D_\phi}(\text{nn}_q, q) > \frac{t}{2\mu} r_w,$$

and for the upper bound on $\sqrt{D_\phi}(\text{rep}_w, q)/\sqrt{D_\phi}(\text{nn}_q, q)$. First by right-sided μ-defectiveness,

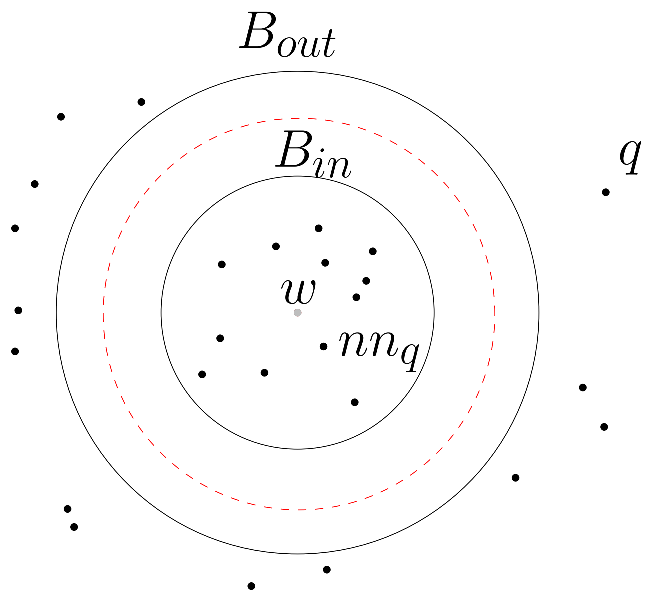


Figure 2.4. q is outside $(1+\frac{t}{2})r_{\text{in}}$ so we search $\mathbf{w}_{\text{out}}$, but $\text{nn}_q \in \mathbf{w}_{\text{in}}$

$$\begin{aligned}
\sqrt{D_\phi}(\mathrm{rep}_W,q) - \sqrt{D_\phi}(\mathrm{nn}_q,q) &< \mu\sqrt{D_\phi}(\mathrm{rep}_W,\mathrm{nn}_q)\\
\sqrt{D_\phi}(\mathrm{rep}_W,q) &< \sqrt{D_\phi}(\mathrm{nn}_q,q) + \mu r_W\\
\frac{\sqrt{D_\phi}(\mathrm{rep}_W,q)}{\sqrt{D_\phi}(\mathrm{nn}_q,q)} &< 1+\mu\frac{r_W}{\sqrt{D_\phi}(\mathrm{nn}_q,q)}\\
\frac{\sqrt{D_\phi}(\mathrm{rep}_W,q)}{\sqrt{D_\phi}(\mathrm{nn}_q,q)} &< 1+\mu\frac{r_W}{\frac{t}{2\mu}r_W}\\
\frac{\sqrt{D_\phi}(\mathrm{rep}_W,q)}{\sqrt{D_\phi}(\mathrm{nn}_q,q)} &< 1+\mu\frac{2\mu}{t} = 1+2\frac{\mu^2}{t}.
\end{aligned}$$

We now consider the other case. Suppose $\mathrm{nn}_q \in \mathbf{w}_{\mathrm{out}}$ and we search in $\mathbf{w}_{\mathrm{in}}$ instead. The analysis is almost identical. By construction we must have:

$$\begin{aligned}
\sqrt{D_\phi}(\mathrm{rep}_W,q) &< \left(1+\frac{t}{2}\right) r_W\\
\sqrt{D_\phi}(\mathrm{rep}_W,\mathrm{nn}_q) &> (1+t)r_W.
\end{aligned}$$

Again, left-sided μ-defectiveness yields:

$$\sqrt{D_\phi}(\mathrm{nn}_q,q) > \frac{t}{2\mu}r_W.$$

We can simply take the ratios of the two:

$$\frac{\sqrt{D_\phi}(\mathrm{rep}_W,q)}{\sqrt{D_\phi}(\mathrm{nn}_q,q)} < \frac{(1+\frac{t}{2})r_W}{\frac{t}{2\mu}r_W} = \mu + \frac{2\mu}{t}.$$

Taking an upper bound of the approximation quality provided by each case, we get that the ring separator provides us a $\mu + 2\frac{\mu^2}{t}$ rough approximation. Substitute $t \le \frac{1}{\log n}$ and the time bound follows from the bound of the depth of the tree in lemma 2.7.4. ■

Corollary 2.7.2 *We can find a $O(\mu+2\mu^2\log n)$ nearest neighbor to query point q in $O((\mu+1)^{2d}d^d\log n)$ time $\sqrt{D_\phi}$ using a $O(\frac{1}{\log n})$ ring-tree constructed in $O(d^d(\mu+1)^{2d}n\log(n))$ expected time.*

Proof. Set $t = \frac{1}{\log n}$, using lemma 2.7.4. The construction time for the ring-tree follows by combining lemmas 2.7.4 and 2.7.3. ■

2.7.2 Asymmetric quadtree decomposition

As in Section 2.5, we use the approximate near-neighbor returned by the ring-separator-tree query to progressively expand cells, using a subdivide-and-search procedure similar to Algorithm 5 albeit with $\sqrt{D_{s\phi}}$ replaced with $\sqrt{D_\phi}$. A key difference is the procedure used to bisect a cell.

Lemma 2.7.6 *Let G be a $\sqrt{D_\phi}$-box with maximum $\sqrt{D_\phi}$-side-length s and G_i its subcells produced by partitioning each side of G into two equal intervals under $\sqrt{D_\phi}$. For all nonempty subboxes G_i of G, we can find $p_i \in P \cap G_i$ in $O(2^d \log^d n)$ total time complexity, and the maximum $\sqrt{D_\phi}$-side-length of any G_i is at most $\frac{s}{\sqrt{2}}$.*

Proof. Note that G is defined as a product of d intervals. For each interval, we can find an approximate bisecting point under $\sqrt{D_\phi}$ in $O(1)$ time. Here the bisection point x of interval $[ab]$ is such that $\sqrt{D_\phi(a,x)} = \sqrt{D_\phi(x,b)}$. By resorting to the RTI for D_ϕ, we get that $D_\phi(a,x) + D_\phi(x,b) < s^2$ and hence $D_\phi(a,x) = D_\phi(x,b) < \frac{s^2}{2}$ which implies $\sqrt{D_\phi(a,x)} = \sqrt{D_\phi(x,b)} < \frac{s}{\sqrt{2}}$. The rest of our proof follows as in lemma 2.5.3. ■

We now bound the number of cells that will be added to our search queue. We do so indirectly, by placing a lower bound on the maximum $\sqrt{D_\phi}$-side-length of all such cells, and note that for the asymmetric case we get an additional factor of $\frac{1}{\mu+1}$.

Lemma 2.7.7 *The $\sqrt{D_\phi}$-diameter of an $\sqrt{D_\phi}$-cube C of $\sqrt{D_\phi}$-side-length s is bounded by $(\mu+1)\sqrt{d}s$.*

Proof. Since the cube may be covered by a $\sqrt{D_\phi}$-left-ball of radius $\sqrt{d}s$ placed at a suitably chosen corner (by lemma 2.3.6), lemma 2.7.1 bounding the diameter of such a ball gives the required bound on the diameter of the cube. ■

Lemma 2.7.8 *Algorithm 5 (with $\sqrt{D_{s\phi}}$ replaced by $\sqrt{D_\phi}$) will not add the children of node **C** to our search queue if the maximum $\sqrt{D_\phi}$-side-length of **C** is less than $\frac{\varepsilon D(nn_q,q)}{2\mu(\mu+1)\sqrt{d}}$.*

Proof. Let $\Delta(\mathbf{C})$ represent the diameter or maximum distance between any two points of cell $\mathbf{C}$.

By construction, we can expand $\mathbf{C}$ only if some subcell of $\mathbf{C}$ contains a point p such that $\sqrt{D_\phi}(p,q) \leq (1-\frac{\varepsilon}{2})D_{\text{near}}$. Note that since $\mathbf{C}$ is examined, we have $D_{\text{near}} \leq p(\text{rep}_C, q)$. Now assuming we expand $\mathbf{C}$, then we must have:

$$\begin{aligned} \sqrt{D_\phi}(\text{rep}_C,q) - \sqrt{D_\phi}(p,q) &< \mu\Delta(\mathbf{C}) \\ D_{\text{near}} - \left(1-\frac{\varepsilon}{2}\right)D_{\text{near}} &< \mu\Delta(\mathbf{C}) \\ \frac{\varepsilon}{2}D_{\text{near}} &< \mu\Delta(\mathbf{C}) \\ \frac{\varepsilon}{2\mu}D_{\text{near}} &< \Delta(\mathbf{C}). \end{aligned}$$

Note that we substitute $\sqrt{D_\phi}(\text{rep}_C,q) < D_{\text{near}}$ and that by the definition of D_{near} as our candidate nearest neighbor distance, $\sqrt{D_\phi}(\text{nn}_q,q) < D_{\text{near}}$. Our main modification from the symmetric case is that here $\Delta(\mathbf{C}) < (\mu+1)\sqrt{d}s$ by lemma 2.7.7, where s is the maximum side-length of $\mathbf{C}$, as opposed to $\sqrt{d}s$. ■

The main difference between this lemma and lemma 2.5.4 is the extra factor of $\mu+1$ that we incur (as discussed) because of asymmetry. We only need do a little more work to obtain our final bounds:

Lemma 2.7.9 *Given a $\sqrt{D_\phi}$-cube G of $\sqrt{D_\phi}$-side-length D_{rough}, and letting $x = \frac{1}{\varepsilon^d}2^d\mu^d(\mu+1)^d d^{\frac{d}{2}}\left(\frac{D_{rough}}{\sqrt{D_\phi}(nn_q,q)}\right)^d$ we can compute a $(1+\varepsilon)$- right sided nearest neighbor to q in G in $O(x^2\log^d n)$ time.*

Proof. Consider a quadtree search from q on a $\sqrt{D_\phi}$-cube G of $\sqrt{D_\phi}$-side-length D_{rough}. By lemma 2.7.8, our algorithm will not expand cells with all $\sqrt{D_\phi}$-side-lengths smaller than $\varepsilon\sqrt{D_\phi}(\text{nn}_q,q)/2\mu(\mu+1)\sqrt{d}$. But since the $\sqrt{D_\phi}$-side-length reduces by at least a factor of $\sqrt{2}$ in each dimension upon each split, all $\sqrt{D_\phi}$-side-lengths are less than this value after $k = \log_{\sqrt{2}}\left(2D_{\text{rough}}\mu(\mu+1)\sqrt{d}/\varepsilon\sqrt{D_\phi}(\text{nn}_q,q)\right)$ repeated bisections of our root cube. Observe now that $O(\log^d n)$ time is spent at each node by lemma 2.7.6, that at the k-th level the number of nodes expanded is 2^{kd}, and that $\log_{\sqrt{2}} n = (\log_2 n)^2$. We then get a final time complexity bound of

$$O\left((1/\varepsilon^{2d})2^{2d}\mu^{2d}(\mu+1)^{2d}d^d\left(D_{\text{rough}}/\sqrt{D_\phi}(\text{nn}_q,q)\right)^{2d}\log^d n\right).$$

■

Substituting $D_{\text{rough}} = \mu^2 \log(n)\sqrt{D_\phi}(\text{nn}_q, q)$ in lemma 2.7.9 gives us a bound of $O\left(2^{2d}\frac{1}{\varepsilon^{2d}}\mu^{6d}(\mu+1)^{2d}d^d \log^{3d} n\right)$. This time is per cube of F that covers right-ball $B(q, D_{\text{rough}})$. Noting that there are 2^d such cubes gives us a final time complexity of $O\left(2^{3d}\frac{1}{\varepsilon^{2d}}\mu^{6d}(\mu+1)^{2d}d^d \log^{3d} n\right)$. The space bound follows as in Section 2.5.

2.7.3 Logarithmic bounds for asymmetric Bregman divergences

We now extend our logarithmic bounds from Section 2.6 to asymmetric Bregman divergence $\sqrt{D_\phi}$. First note that the following lemma goes through by identical argument to lemma 2.6.1.

Lemma 2.7.10 *Given an interval $I = [x_1 x_2] \subset \mathbb{R}$ s.t. $x_1 < x_2$, $D_e(x_1, x_2) = r_e$ and $\sqrt{D_\phi}(x_1, x_2) = r_\phi$, suppose we divide I into m subintervals of equal length under D_e with endpoints $x_1 = a_0 < a_1 < \ldots < a_{m-1} < a_m = x_2$ where $D_e(a_i, a_{i+1}) = r_e/m$, for all $i \in [0 \ldots m-1]$. Then $\frac{r_\phi}{c_0 m} \le \sqrt{D_\phi}(a_i, a_{i+1}) \le \frac{c_0 r_\phi}{m}$.*

Corollary 2.7.3 *If we recursively bisect an interval $I = [x_1 x_2] \subset \mathbb{R}$ s.t. $D_e(x_1, x_2) = r_e$ and $\sqrt{D_\phi(x_1, x_2)} = r_\phi$ into 2^i equal subintervals (under D_e), then $\frac{r_\phi}{c_0 2^i} \le \sqrt{D_\phi(a_k, a_{k+1})} \le \frac{c_0 r_\phi}{2^i}$ for any of the subintervals $[a_k a_{k+1}]$ so obtained. Hence after $i = \lceil \log \frac{c_0 r_\phi}{x} \rceil$ subdivisions, all intervals will be of length at most x under $\sqrt{D_\phi}$.*

We now construct a compressed Euclidean quad tree as before, modifying the Section 2.6 analysis slightly to account for the weaker packing bounds for $\sqrt{D_\phi}$ and the extra $\mu + 1$ factor on the diameter of a cell.

Theorem 2.7.1 *Given an asymmetric decomposable Bregman divergence D_ϕ that is μ-defective over a domain with associated c_0 as in Section 2.6, we can compute a $(1+\varepsilon)$-approximate right-near-neighbor in time $O\left((\mu+1)^d d^{\frac{d}{2}} \log n + \left(\frac{2c_0{}^4(\mu+1)\mu^3\sqrt{d}}{\varepsilon}\right)^d\right)$.*

We note our first new lemma, a slightly modified packing bound due to $\sqrt{D_\phi}$ not having a direct RTI.

Lemma 2.7.11 *Given an interval $[x_1 x_2] \subset \mathbb{R}$ s.t. $\sqrt{D_\phi(x_1, x_2)} = r > 0$, and intervals with endpoints $a_0 < a_1 < \ldots < a_{m-1} < a_m$, s.t. for all $i \in [0 \ldots m-1]$, $\sqrt{D_\phi(a_i, a_{i+1})} \geq l$, at most $O(\frac{c_0 r}{l})$ such intervals intersect $[x_1 x_2]$.*

Proof. By the Lagrange form,

$$\frac{l}{r} < \frac{\sqrt{D_\phi(a_i, a_{i+1})}}{\sqrt{D_\phi(x1, x2)}} < c_0 \frac{D_e(a_i, a_{i+1})}{D_e(x_1, x_2)}, \tag{2.13}$$

or we can say that $\frac{D_e(a_i, a_{i+1})}{D_e(x1, x2)} > \frac{l}{rc_0}$. The RTI for D_e then gives us the required result. ■

Corollary 2.7.4 *Given a ball B of radius r under $\sqrt{D_\phi}$, there can be at most $c_0^d (\frac{r}{l})^d$ disjoint $\sqrt{D_\phi}$-cubes that can intersect B where each cube has side-length at least l under $\sqrt{D_\phi}$.*

As before, we find the smallest enclosing Bregman cube of side-length s that encloses our point set, and then construct a compressed Euclidean quadtree in preprocessing. Let L_i denote the cells at the i-th level.

Lemma 2.7.12 *Given a $\sqrt{D_\phi}$ right-ball B of radius r under $\sqrt{D_\phi}$, let $i = \log \frac{s}{c_0 r}$. Then $|L_i \cap B| \leq O({c_0}^d)$ and the side-lengths of each cell in L_i are between r and ${c_0}^2 r$ under $\sqrt{D_{s\phi}}$. We can also explicitly retrieve the quadtree cells corresponding to $|L_i \cap B|$ in $O({c_0}^d \log n)$ time.*

Proof. Note that for cells in L_i, we have $\sqrt{D_\phi}$-side-lengths between $\frac{s}{c_0 2^i}$ and $\frac{c_0 s}{2^i}$ by Corollary 2.7.3. Substituting $i = \log \frac{s}{c_0 r}$, these cells have side-length between r and ${c_0}^2 r$ under $\sqrt{D_{s\phi}}$. Now, we look in each dimension at the number of disjoint intervals of length at least r that can intersect B. By lemma 2.7.11, this is at most c_0. The rest of the proof follows as in lemma 2.6.2. ■

We first obtain in $O(\log n)$ time with our asymmetric ring-tree an $O(n)$ ANN q_{rough} to query point q, such that $D_{\text{rough}} = \sqrt{D_\phi}(q_{\text{rough}}, q) = O\left(\mu^2 n \sqrt{D_\phi}(\text{nn}_q, q)\right)$.
We then use lemma 2.7.12 to get $O({c_0}^d)$ cells of our quadtree that intersect right-ball $B\left(q, \sqrt{D_\phi(q_{\text{rough}}, q)}\right)$. Let us call this collection of cells as Q. We then carry out a

quadtree search on each element of Q. Note that we expand only cells which may contain a point nearer to query point q than the current best candidate. We bound the depth of our search using μ-defectiveness similar to lemma 2.6.4.

Lemma 2.7.13 *We need only expand cells of $\sqrt{D_\phi}$-diameter greater than* $\frac{\varepsilon\sqrt{D_\phi(nn_q,q)}}{2\mu}$

Proof. By μ-defectiveness, similar to lemma 2.5.4. ■

Corollary 2.7.5 *We will not expand cells where the length of each side is less than* $x = \frac{\varepsilon\sqrt{D_\phi(nn_q,q)}}{2\mu(\mu+1)\sqrt{d}}$ *under* $\sqrt{D_\phi}$.

Proof. Note that a quadtree cell **C** where every side-length is less than x can be covered by a ball of radius $\sqrt{d}x$ under $\sqrt{D_\phi}$ with appropriately chosen corner as center of ball, as explained in proof of lemma 2.3.6. Now by lemma 2.7.1, $\sqrt{D_\phi(a,b)} \leq (\mu+1)\sqrt{d}x$, $\forall a,b \in$ **C**. Substituting for x from lemma 2.7.13, the $\sqrt{D_\phi}$-diameter of **C** is at most $\frac{\varepsilon\sqrt{D_\phi}(\mathrm{nn}_q,q)}{2\mu}$. ■

Let the spread be $\beta = \frac{D_{\text{rough}}}{\sqrt{D_\phi}(\mathrm{nn}_q,q)} = O(\mu^2 n)$.

Lemma 2.7.14 *We will only expand our tree to a depth of* $k = \log(2c_0^3\mu(\mu+1)\beta\sqrt{d}/\varepsilon)$.

Proof. Note first that $D_{\text{rough}} = O\left(\beta\sqrt{D_\phi}(\mathrm{nn}_q,q)\right)$. Then by lemma 2.7.12, each of the cells of our corresponding quadtree is of $\sqrt{D_\phi}$-side-length at most $c_0^2 D_{\text{rough}}$. Using 2.7.5 to lower bound the minimum $\sqrt{D_\phi}$-side-length of any quadtree cell expanded, and 2.7.3 to bound number of bisections needed to guarantee all $\sqrt{D_\phi}$-side lengths are within this gives us out bound. ■

Lemma 2.7.15 $n_i < 2^d(\mu^d d^{\frac{d}{2}} c_0^{5d} + (\frac{c_0^2 2^i}{\beta})^d)$ *cells are expanded at the i-th level.*

Proof. Recalling that the cells of Q start with all $\sqrt{D_\phi}$-side-lengths at most $c_0^2 D_{\text{rough}}$, at the i-th level the side-length of a cell **C** is at most $\frac{c_0^3 D_{\text{rough}}}{2^i}$ under $\sqrt{D_\phi}$ by Corollary 2.7.3.

And using lemma 2.7.1, $\Delta\mathbf{C} < \sqrt{d}(\mu+1)\frac{c_0^3 D_{\text{rough}}}{2^i}$. Hence by μ-defectiveness there must be a point at distance at most $D_{\text{best}} = \sqrt{D_\phi(\text{nn}_q, q)} + \frac{\mu(\mu+1)c_0^3\sqrt{d}D_{\text{rough}}}{2^i}$.

The $\sqrt{D_\phi}$-side-length of a cell C at this level is at least $\frac{D_{\text{rough}}}{c_0 2^i}$, so the number of cells expanded is at most $n_i = c_0^d(\frac{D_{\text{best}}}{\Delta\mathbf{c}})^d = c_0^d(\mu(\mu+1)\sqrt{d}c_0^4 + \frac{c_0 2^i}{\beta})^d$, by Corollary 2.7.4. Using the fact that $(a+b)^d < 2^d(a^d+b^d)$, we get $n_i < 2^d\left(\mu^d(\mu+1)^d d^{\frac{d}{2}} c_0^{5d} + (\frac{c_0^2 2^i}{\beta})^d\right)$. ■

Simply summing up all i, the total number of nodes explored is

$$O(2^d\mu^d(\mu+1)^d c_0^{5d}\log(2c_0^3\mu\beta\sqrt{d}/\varepsilon) + 2^{2d}c_0^{5d}\mu^d(\mu+1)^d d^{\frac{d}{2}}/\varepsilon^d),$$

or

$$O\left(2^d\mu^d(\mu+1)^d c_0^{5d}\log n + 2^{2d}c_0^{5d}\mu^d(\mu+1)^d d^{\frac{d}{2}}/\varepsilon^d\right),$$

after substituting back for β and ignoring smaller terms. Recalling that there are c_0^d cells in Q adds a further c_0^d multiplicative factor. This time complexity of this quadtree phase (number of cells explored) of our algorithm dominates the time complexity of the ring-tree search phase of our algorithm, and hence is our overall time complexity for finding a $(1+\varepsilon)$ ANN to q. For space and preconstruction time, we note that compressed Euclidean quadtrees can be built in $O(n\log n)$ time and require $O(n)$ space (139), which matches our bound for the ring-tree construction phase of our algorithm requiring $O(n\log n)$ time and $O(n)$ space.

2.8 Numerical arguments for bisection

In our algorithms, we are required to *bisect* a given interval with respect to the distance measure D, as well as construct points that lie a fixed distance away from a given point. We note that in both these operations, we do not need exact answers: a constant factor approximation suffices to preserve all asymptotic bounds. In particular, our algorithms assume two procedures:

1. Given interval $[ab] \subset \mathbb{R}$, find $\bar{x} \in [ab]$ such that $(1-\alpha)\sqrt{D_{s\phi}(a,\bar{x})} < \sqrt{D_{s\phi}(\bar{x},b)} < (1+\alpha)\sqrt{D_{s\phi}(a,\bar{x})}$.

2. Given $q \in \mathbb{R}$ and distance r, find $\bar{x}$ s.t $|\sqrt{D_{s\phi}(q,\bar{x})} - r| < \alpha r$.

Cayton presents a similar bisection procedure (39) as ours for the second task above, although our analysis of the convergence time is more explicit in our parameters of μ and c_0. For a given $\sqrt{D_{s\phi}} : \mathbb{R} \to \mathbb{R}$ and precision parameter $0 < \alpha < 1$, we describe a procedure that yields an $0 < \alpha < 1$ approximation in $O(\log c_0 + \log \mu + \log \frac{1}{\alpha})$ steps for both problems, where c_0 implicitly depends on the domain of convex function ϕ:

$$c_0 = \sqrt{\max_{1 \le i \le d} \left(\max_x \phi_i''(x) / \min_y \phi_i''(y) \right)}. \tag{2.14}$$

Note that this implies linear convergence. While more involved numerical methods such as Newton's method may yield better results, our approximation algorithm serves as proof-of-concept that the numerical precision is not problematic.

A careful adjustment of our NN-analysis now gives a time complexity to compute a $(1+\varepsilon)$-ANN to query point q of $O\left(\left(\log \mu + \log c_0 + \log \frac{1}{\alpha} \right) 2^{2d} (1+\alpha)^d \frac{1}{\varepsilon^d} \mu^{3d} d^{\frac{d}{2}} \log^{2d} n \right)$.

We now describe some useful properties of $D_{s\phi}$.

Lemma 2.8.1 *Consider $\sqrt{D_{s\phi}} : \mathbb{R} \to \mathbb{R}$ such that $c_0 = \sqrt{\max_x \phi''(x) / \min_y \phi''(y)}$. Then for any two intervals $[x_1 x_2], [x_3 x_4] \subset \mathbb{R}$,*

$$\frac{1}{c_0} \frac{|x_1 - x_2|}{|x_3 - x_4|} < \frac{\sqrt{D_{s\phi}(x_1,x_2)}}{\sqrt{D_{s\phi}(x_3,x_4)}} < c_0 \frac{|x_1 - x_2|}{|x_3 - x_4|}. \tag{2.15}$$

Proof. *The lemma follows by the definition of c_0 and by direct computation from the Lagrange form of $\sqrt{D_{s\phi}(a,b)}$, i.e., $\sqrt{D_{s\phi}(a,b)} = \sqrt{\phi''(\bar{x}_{ab})} |b-a|$, for some $\bar{x}_{ab} \in [ab]$.* ■

Lemma 2.8.2 *Given a point $q \in \mathbb{R}$, distance $r \in \mathbb{R}$, precision parameter $0 < \alpha < 1$ and a μ-defective $\sqrt{D_{s\phi}} : \mathbb{R} \to \mathbb{R}$, we can locate a point x_i such that $|\sqrt{D_{s\phi}(q,x_i)} - r| < \alpha r$ in $O(\log \frac{1}{\alpha} + \log \mu + \log c_0)$ time.*

Proof. Let x be the point such that $\sqrt{D_{s\phi}(q,x)} = r$. We outline an iterative process, Algorithm 6, with i-th iterate x_i that converges to x.

```
Let $x_0 > q$ be such that $\frac{\sqrt{\phi''(q)}}{c_0}(x_0 - q) = r$
Let step $= (x_0 - q)/2$
**repeat**
  **if** $\sqrt{D_{s\phi}(q,x_i)} < r$ **then**
    $x_{i+1} = x_i +$ step
  **else**
    $x_{i+1} = x_i -$ step
  **end if**
  step $=$ step$/2$
**until** $|\sqrt{D_{s\phi}(q,x_i)} - r| \leq \alpha r$
Return $\bar{x} = x_i$
```

Algorithm 6: QueryApproxDist(q,r,c_0,α)

First note that $\frac{\sqrt{\phi''(q)}}{c_0} \leq \sqrt{\min_y \phi''(y)}$ and $\frac{\sqrt{\phi''(q)}}{c_0} \geq \frac{\max_z \sqrt{\phi''(z)}}{c_0^2}$. It immediately follows that $r \leq \sqrt{D_{s\phi}(q,x_0)} \leq c_0^2 r$.

By construction, $|x_i - x| \leq |x_0 - q|/2^i$. Hence by lemma 2.8.1, $\sqrt{D_{s\phi}(x_i,x)} < \frac{c_0^3 r}{2^i}$. We now use μ-defectiveness to upper bound our error $|\sqrt{D_{s\phi}(q,x_i)} - \sqrt{D_{s\phi}(q,x)}|$ at the i-th iteration:

$$\left|\sqrt{D_{s\phi}(q,x_i)} - \sqrt{D_{s\phi}(q,x)}\right| < \frac{\mu c_0^3 r}{2^i}, \tag{2.16}$$

choosing i such that $(\mu c_0^3)/2^i \leq \alpha$ implies that $i \leq \log \frac{1}{\alpha} + \log \mu + 3 \log c_0$. ■

An almost identical procedure can locate an approximate bisection point of interval $[ab]$ in $O(\log \mu + \log c_0 + \log \frac{1}{\alpha})$ time, and similar techniques can be applied for $\sqrt{D_\phi}$. We omit the details here.

CHAPTER 3

A DIRECTED ISOPERIMETRIC INEQUALITY WITH APPLICATION TO BREGMAN NEAR NEIGHBOR LOWER BOUNDS

Bregman divergences retain many of the *combinatorial* properties of ℓ_2 and so *exact* geometric algorithms based on space decomposition (Voronoi diagrams, convex hulls and so on) can be used to compute the corresponding Bregman counterparts (34). But the divergences are asymmetric[1] and violate triangle inequality, and so break most approximation algorithms for distance problems (clustering, near neighbor search and the like) that make heavy use of these properties.

This "degree of violation" can be quantified as a scalar parameter μ that depends only on the functional form of the divergence (not the size of input or its dimension). There are many ways μ is defined in the literature (1, 4, 107), and these are all loosely related to the view of μ as a measure of *asymmetry*: given a Bregman divergence D over a domain Δ, define μ as $\max_{x,y\in\Delta} \frac{D(x,y)}{D(y,x)}$.

To the best of our knowledge, μ appears as a term in theoretical guarantees for *all* constant factor approximation algorithms for geometric problems in these spaces. Indeed in Chapter 2, our upper bounds for a $(1+\varepsilon)$-ANN for Bregman divergences in low dimensions incorporated an exponential dependence on μ. This is highly unsatisfactory because μ can grow without bound independent of the data size or dimensionality. It is therefore natural to ask the question:

Is this dependence on μ intrinsic? Or are there clever algorithms that can circumvent the effect of asymmetry for such problems?

[1] In fact the squared Euclidean distance is the *unique* symmetric Bregman divergence.

In this chapter we provide the first evidence that this dependence is indeed intrinsic under a broad range of the parameters n and d (namely $d \gg \log n$). We focus on the fundamental problem of ANN search, which has been studied extensively for Bregman divergences.

3.1 Main results

We show the following under the cell probe model for *uniform* Bregman divergences (loosely speaking, distances composed as a sum of d identical measures):

Theorem 3.1.1 *For a uniform Bregman divergence D with measure of asymmetry μ in each dimension, let $L = \min\left(\frac{d}{\log n}, \mu\right)$. Any nonadaptive data structure which in r probes can return even a c' approximation to the nearest neighbor under D with constant probability (over the choice of query) requires $\Omega(dn^{1+\Omega(L/c'r)})$ space.*

In particular, this lower bound applies to methods based on locality-sensitive hashing and to several popularly used divergences such as the Kullback-Leibler or Itakura-Saito distances. Note that in comparison to the space lower bound of $\Omega(dn^{1+\Omega(1/cr)})$ for Euclidean (or ℓ_1) ANN (128) which is subquadratic and near linear for sufficiently high c, the space lower bound here is polynomial in n with an exponent of $\Omega(\mu)$ for constant factor approximations and (as we show later) strengthens upto $\mu = \Theta(d/\log n)$.[2] This indicates the increased hardness introduced by asymmetry.

3.1.1 ANN and Partial Match

There is one aspect of our work that may be of independent interest. Separately from our main result, we can show a direct reduction from geometric problems on the Hamming cube to the equivalent problems for Bregman divergences. In Section 3.11.2 we find a very interesting "interpolation" of lower bounds parametrized by μ: a constant factor approximation for Bregman ANN with $\mu = O(1)$ implies a constant factor approximation for ANN under ℓ_1, and a similar approximation for Bregman ANN with $\mu = \Omega(d)$ implies a constant factor approximation for Partial Match, which is a notoriously hard problem. Intriguingly while lower bounds for Partial Match are in general higher than those of ANN,

[2]Note that since there are 2^d points on the cube, we must have that $d > \log n$ just to fit all the point set in the cube.

at the intermediate point $\mu = \Theta(\frac{d}{\log n})$ in the interpolation we already obtain lower bounds that are as strong as those known for Partial Match (with the qualifier that our analysis restricts to nonadaptive algorithms).

One interpretation of this is that μ captures the intuition that Partial Match is an "asymmetric" version of ANN. It would also be interesting if this directed perspective allows us to obtain improved lower bounds for Partial Match itself by a reduction in the opposite direction. Indeed in the strictly linear space regime, the lower bound of $\Omega(d)$ queries for our asymmetric ANN is stronger than those known for Partial Match ($\Omega(d/\log d)$ for adaptive algorithms by (131)).

3.2 Overview of our approach

Our approach makes use of the Fourier-analytic approach to proving lower bounds for (randomized) near-neighbor data structures that has been utilized in a number of prior works (112, 128, 129). This approach generally works as follows: one thinks of the purported data structure as a partition of the Hamming cube, and in particular as a function defined on the Hamming cube. Then one shows that any such function is "expansive" with respect to small perturbations: in effect, that points scatter all over the cube. As a consequence, probing any particular cell of a data structure does not yield enough useful information because of the scattering, and one has to make many probes to be sure. The key technical result is showing that the function is expansive, and this is done using Fourier-analytic machinery, and hypercontractivity of the noise operator in particular (124). One also needs to construct a "gap instance" where the gap between nearest neighbor and second nearest neighbor is large.

While black-box reductions from ℓ_1-ANN can yield weak lower bounds for Bregman divergences (see Section 3.11.1), we need a much stronger argument to get a μ-sensitive bound. Specifically, we need the following components:

- **A gap instance**: We create an instance that separates a near neighbor at distance εd from a second nearest neighbor at distance μd. To do so, we define a Bregman hypercube and associated asymmetric noise operator (with different probabilities of changing 0 to 1 and 1 to 0) and observe our gap is far stronger than the natural symmetric analog - $\Omega\left(\frac{\mu}{\varepsilon}\right)$ vs $\Omega\left(\frac{1}{\varepsilon}\right)$.

- **Directed hypercontractivity**: The Fourier-analytic machinery breaks down for our noise operator because of lack of symmetry. Indeed, a simple example shows that a natural directed analog of the Bonami-Beckner (BB) inequality cannot be true. Instead, we prove a directed BB inequality in Section 3.6 that is true "on average", or on a subset of the hypercube, which will be sufficient for our lower bound. We prove this by relating the norm of the directed noise operator to related norms on biased (but symmetric) measure spaces, allowing us to make use of BB-type inequalities in these spaces.

- **A scatter lemma**: Showing that points "scatter" is relatively easy in symmetric spaces: in the directed setting, the argument can be made in a similar way but requires a nontrivial analysis of associated collision rates and inner products which we carry out in Section 3.8.

- **An information-theoretic argument**: We borrow the argument used by (128). Essentially, the scatter lemma shows a small sampling of the cells of a successful data structure must resolve many query points and thus will have high information content. This allows us to lower bound the space required by such a structure in Section 3.10 and obtain Theorem 3.1.1.

3.3 Related work

As discussed earlier in this dissertation, while exact algorithms for Euclidean case can often be applied to the Bregman divergences almost unchanged, the parallels to do *not* carry over to the approximate setting with the lack of a triangle inequality and symmetry rendering most tools for algorithm design useless. The algorithms that do exist in the literature generally attempt a work around via a structure constant μ. This constant is at least 1, and grows larger as the space becomes increasingly nonmetric. For ANN search, in Chapter 2 we gave an algorithm that is efficient in constant dimensions. Our algorithm yielded a $1+\varepsilon$ approximate nearest neighbor with an additional dependence on $\mu^{O(d)}$ besides standard dependence on factors of $\frac{1}{\varepsilon^{O(d)}}$ and $\log n$. Indeed, the results in this chapter emerged as a consequence of attempting to extend our results to higher dimensions.

Lower bounds for near neighbor search in metric spaces have been studied extensively. Borodin, Ostrovsky and Rabani (35) show a lower bound that any randomized cell probe algorithm for the exact match problem that must probe at least $\Omega(\log d)$ cells. Barkol and Rabani improve this bound to $\Omega(\frac{d}{\log n})$ cells (28). Liu (103) proves a lower bound of $d^{1-o(1)}$ on the query time of a deterministic approximate nearest neighbor algorithm in the cell probe model, whereas Chakrabarti and Regev give a lower bound of $\Omega\left(\frac{\log\log d}{\log\log\log d}\right)$ for the randomized case (41).

Our work is in the spirit of the program initiated by Motwani, Naor and Panigrahy (112), who analyze a random walk in the Hamming cube to lower bound the LSH quality parameter ρ as $\frac{1}{2c}$ (c is the separation between near and far points). O'Donnell, Wu and Zhou (125) later tighten this to $\frac{1}{c}$. Panigrahy, Talwar and Wieder (128) use the Boolean noise operator to simulate perturbations on the Hamming cube, and use hypercontractivity to show that these Hamming balls touch many cells of a data structure and obtain space-query trade off cell probe lower bounds. They then extend these to broader classes of metric spaces with certain isoperimetric properties of vertex and edge expansion (129). This is not a comprehensive survey; (129) gives a good overview of several of the known lower bounds. All of the above approaches use Fourier analysis on Boolean functions over the hypercube. This is a vast literature that we will not survey here: the reader is pointed to Ryan O'Donnell's lecture notes (124). In particular, we make use of results by Keller (91) and Ahlberg *et al.* (6) on the analysis of the noise operator in biased spaces.

3.4 The Bregman cube and redefining μ

We recall from Chapter 1 the definition of *decomposable* Bregman divergences. Suppose the inducing convex function ϕ has domain $M = \prod_{i=1}^{d} M_i$ and can be written as $\phi(x) = \sum_{i=1}^{d} \phi_i(x_i)$, where $\phi_i : M_i \subset \mathbb{R} \to \mathbb{R}$ is also strictly convex and differentiable in relint(M_i). Then

$$D_\phi(x,y) = \sum_{i=1}^{d} D_{\phi_i}(x_i, y_i)$$

is a *decomposable* Bregman divergence. We introduce here the special case of a *uniform* Bregman divergence which is a decomposable D_ϕ where all the $\phi_i, 1 \leq i \leq d$ are identical. In this case, we simply refer to each ϕ_i as $\phi_{\mathbb{R}} : \mathbb{R} \to \mathbb{R}$ and we have that $\phi(x) = \sum_{i=1}^{d} \phi_{\mathbb{R}}(x_i)$. Note that most commonly used Bregman divergences are uniform, including the Kullback-Leibler, Itakura-Saito, Exponential distance and Bit entropy. In what follows we will limit ourselves to uniform Bregman divergences.

3.4.1 Quantifying asymmetry

It is clear from the definition of D_ϕ that in general $D_\phi(x,y) \neq D_\phi(y,x)$. In what follows, we redefine the *measure of asymmetry* as $\mu = \max_{x,y \in M} \frac{D_\phi(x,y)}{D_\phi(y,x)}$.

By construction, $\mu \geq 1$. But it is not arbitrary; rather, it is a function of the generating convex function ϕ and the domain over which it is defined. To see this, recall that the Bregman divergence $D_\phi(x,y)$ can be viewed as the error (evaluated at x) incurred in replacing ϕ by its first-order approximation $\tilde{\phi}(x) = \phi(y) + \langle \nabla\phi(y), x-y \rangle$. By the Lagrange mean-value theorem, this error can be written as the quadratic form $D_\phi(x,y) = \langle x-y, \nabla^2\phi(c)(x-y) \rangle$ where $\nabla^2\phi$ is the Hessian associated with ϕ, and $c = c(x,y)$ is some point on the line connecting x and y. Note that this point c will general be different to the point c' that achieves equality when measuring $D_\phi(y,x)$.

Thus $\frac{D_\phi(x,y)}{D_\phi(y,x)}$ is bounded by the ratio of the maximum to minimum eigenvalue that the Hessian $\nabla^2\phi$ realizes over the domain M. In particular, μ need not depend on the number of points n or the dimension d.

Most prior work on algorithms with Bregman divergences focus on violations of the triangle inequality, rather than symmetry, including our own definition of μ in Chapter 2. However, the different variants of μ defined there all relate in similar fashion to the ratio of eigenvalues of the Hessian of ϕ, and can be shown to be loosely equivalent to each other; in the sense that if the measure of asymmetry μ grows without bound, so do these measures.

3.4.2 The Bregman cube

We introduce a new structure, the *Bregman cube* $B_\phi = \{0,1\}^d$ along with asymmetric distance measure D. This is combinatorially equivalent to a regular Hamming cube, but where distances of 1 and μ are associated with flipping a bit from 1 to 0 and 0 to 1, respectively. More precisely, given $D : \{0,1\}^d \times \{0,1\}^d \to \mathbb{R}$ and asymmetry parameter μ, we stipulate:

$$D(x,y) = \mu|\{i : y_i > x_i\}| + |\{j : x_j > y_j\}|, \forall x,y \in \{0,1\}^d. \tag{3.1}$$

We note now how B_ϕ and the associated measure D can be induced from a uniform Bregman divergence D_ϕ on $\mathbb{R}^d$. Let the asymmetry parameter μ of $D_{\phi_\mathbb{R}}$ be realized by points a, $b \in \mathbb{R}$. W.l.o.g. (due to scaling) assume $D_{\phi_\mathbb{R}}(b,a) = 1$ and $D_{\phi_\mathbb{R}}(a,b) = \mu$. Then distances on B_ϕ with parameter μ correspond exactly to those on $\{a,b\}^d \subset \mathbb{R}^d$ under D_ϕ. See Figure 3.1.

We use standard notation to define a "c-approximate nearest neighbor" (c-ANN) for query point q and point set P under D. Namely, let $p' \in P$ be a "c-approximate nearest neighbor" to q if $D(q,p') \leq c \min_{p \in P} D(q,p)$. We also fix q to be the first argument in the asymmetric distance D to maintain consistency.

3.5 Preliminaries of Fourier analysis

3.5.1 Basis and Fourier coefficients

Let the *p-biased measure* $\kappa_p = (p\delta_{\{1\}} + (1-p)\delta_{\{0\}})^{\otimes d}$ be the product measure defined over the hypercube $\{0,1\}^d$. Note that for $p = 1/2$ this is the uniform measure over binary strings of length d. All expectations and norms are implicitly defined according to the choice of measure κ_p as follows:

For any function $f : \{0,1\}^d \to \mathbb{R}$, let

1. $E_p[f] = \sum_{x \in \{0,1\}^d} \kappa_p(x) f(x)$.
2. $\|f\|_{j,p} = \left(\sum_{x \in \{0,1\}^d} \kappa_p(x) f(x)^j\right)^{1/j}$.

It is well known that there is a natural Fourier basis for the space F_p of all functions $f : \{0,1\}^d \to \mathbb{R}$ with respect to κ_p (see for example, (6, 91, 124)). For each $x \in \{0,1\}^d$ and $i \in [d]$ let

$$\chi_i^p = \begin{cases} \sqrt{\frac{p}{1-p}} & x_i = 0 \\ -\sqrt{\frac{1-p}{p}} & x_i = 1 \end{cases}.$$

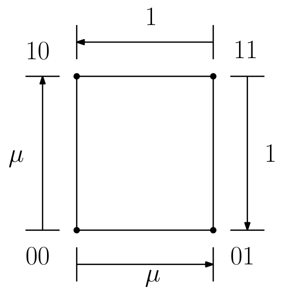


Figure 3.1. Asymmetric distances.

The set of χ_i^p corresponds to a bit wise parity basis which we can extend to arbitrary $S \subset [n]$ as $\chi_S^p(x) = \prod_{i \in S} \chi_i^p(x)$. The resulting χ_S^p form an orthonormal basis of F_p. That is, we can define the Fourier coefficient corresponding to a $S \subset [n]$ as:

$$\hat{f}^{(p)}(S) = \sum_{x \in \{0,1\}^d} \kappa_p(x) f(x) \chi_S^p(x). \tag{3.2}$$

And hence obtain that

$$f = \sum_{S \subseteq [n]} \hat{f}^{(p)}(S) \chi_S^p. \tag{3.3}$$

The orthonormality of the χ_S^p immediately yields the Parseval identity

$$\|f\|_{2,p}^2 = E_p[f^2] = \sum_{S \subseteq [n]} \left(\hat{f}^{(p)}(S) \right)^2. \tag{3.4}$$

See (124) for a full discussion of the parity basis. We note that wherever we drop the superscript and simply write $\hat{f}(S)$ or χ_S, we intend $\hat{f}^{(1/2)}(S)$ and $\chi_S^{\frac{1}{2}}$, respectively.

3.5.2 Noise operator and hypercontractivity

For $x \in \{0,1\}^d$, let y be the random variable obtained by flipping each bit of x with probability p. The *noise operator* $T_\delta f$ for a function f is defined as the expectation of f over y of $T_\delta f(x) = E_y[f(y)]$:

$$T_\delta f(x) = E_y[f(y)].$$

In the case of the uniform measure $\kappa_{\frac{1}{2}}$, $T_\delta f$ can be written as

$$T_\delta f = \sum_S (1-2\delta)^{|S|} \hat{f}^{(1/2)}(S) \chi_S.$$

More generally, given a function f and choice of measure κ_p we define the operator

$$\tau_\delta f = \sum_S \delta^{|S|} \hat{f}^p(S) \chi_S. \tag{3.5}$$

And we note that $T_\delta = \tau_{1-2\delta}$ for the uniform measure.

Theorem 3.5.1 (Hypercontractivity(124))

$$\|\tau_\delta f\|_{2,\frac{1}{2}} \leq \|f\|_{1+\delta^2,\frac{1}{2}}. \tag{3.6}$$

The above result holds for the uniform measure space ($p = 1/2$) but can also be considered in general p-biased measure spaces. The hypercontractivity problem in this context is to find $C(p,\delta)$ such that $\|\tau_\delta f\|_{2,p} \le \|f\|_{1+C(p,\delta),p}$. Partial results were obtained by Talagrand (146), Friedgut (68) and Kindler (93), whereas stronger bounds were obtained by Diaconis and Saloff-Coste (62) and the optimal known value of $C(p,\delta)$ was obtained by Oleskiewicz (126).

We prefer the following formulation of a bound on $C(p,\delta)$ by Keller (91) due to convenience in some algebraic cancellations:

Theorem 3.5.2 *Let $\bar{p} = min(p, 1-p)$, where $0 \le p \le 1$. Then for any $\delta \ge 0$, s.t. $\delta^2\sqrt{\frac{\bar{p}\lfloor\log\frac{1}{\bar{p}}\rfloor}{1-\bar{p}}} \le 1$, we have* [3]

$$\|\tau_\delta f\|_{2,p} \le \|f\|_{1+\delta^2(1-\bar{p})/(\bar{p}\lfloor\log 1/\bar{p}\rfloor),p}. \tag{3.7}$$

Observe that if we set $p = 1/2$ the general expression described in Theorem 3.5.2 reduces to the special case of Theorem 3.5.1. We also note that the constants in our main result of Theorem 3.1.1 may improve slightly if we use, for instance, the optimal value of the hypercontractivity parameter from Oleskiewicz (126); however the asymptotics are unaffected.

3.6 Isoperimetry in the directed hypercube

3.6.1 The asymmetric noise operator

For any point $x \in \{0,1\}^d$, let $\nu_{p_1,p_2}(x)$ be the distribution obtained by independently flipping each 0 bit of x to 1 with probability p_1 and each 1 bit of x to 0 with probability p_2.

Definition 5 (Asymmetric noise operator) *The asymmetric noise operator R_{p_1,p_2} is an operator defined on functions over $\{0,1\}^d$ and is defined as*

$$[R_{p_1,p_2} f](x) = E_{y\sim\nu_{p_1,p_2}(x)}[f(y)].$$

We note that Benjamini, Kalai and Schramm (30) study a version of this asymmetric noise in the context of percolation crossings, and that the formulation of $R_{p,0}$ by Ahlberg *et al.*(6) as a Fourier operator is highly useful in our analysis. We observe first that if we set

[3]In the remainder of the chapter we drop the floor arguments, which do not affect any of the asymptotics of our result.

$p = p_1 = p_2$, then $R_{p_1,p_2} = T_p$. There is in fact a stronger relationship between the two operators.

Theorem 3.6.1 *If* $p_1 \geq p_2$, $p_1 \leq 1 - p_2$ *then* $R_{p_1,p_2} = T_{p_2} R_{\frac{p_1-p_2}{1-2p_2},0}$.

Proof. The overall transition probabilities from a 0 to a 1 and vice versa must match on both sides of the equation. Therefore if we set $R_{p_1,p_2} = T_{p'} R_{p'',0}$, then the following two equations must hold true:

$$p_2 = p'.$$
$$p_1 = (p'')(1-p') + (1-p'')p'.$$

Solving this system yields us $p' = p_2$ and $p'' = \frac{p_1-p_2}{1-2p_2}$. ■

Our goal is to prove hypercontractivity for R_{p_1,p_2}. By the decomposition given in Theorem 3.6.1 and known hypercontractivity bounds for T_p, it will suffice to study how $R_{p,0}$ affects the Fourier coefficients of f. This turns out to be intimately related to the p-biased measure κ_p. We will combine this with standard hypercontractivity results for T_p to obtain the desired bound. (Bounds for $R_{0,p}$ follow by easy analogy and are also presented, although not needed for our main results.) Since we are looking at asymptotic bounds, it will be best to think of both p_1 and p_2 as smaller than a fixed constant, say $\frac{1}{100}$.

3.6.2 Hypercontractivity of $R_{p,0}$

Suppose we are given a function $f : \{0,1\}^d \to \mathbb{R}$.

Lemma 4.2 (6) shows that the Fourier coefficients of the *asymmetric* perturbation of f in a *uniform* space are related to the Fourier coefficients of f in a *biased* space.

Theorem 3.6.2 ((6))

$$\widehat{R_{0,p}f}^{(1/2)}(S) = \left(\sqrt{\frac{1-p}{1+p}}\right)^{|S|} \hat{f}^{\left(\frac{1-p}{2}\right)}(S).$$
$$\widehat{R_{p,0}f}^{(1/2)}(S) = \left(\sqrt{\frac{1-p}{1+p}}\right)^{|S|} \hat{f}^{\left(\frac{1+p}{2}\right)}(S).$$

Using this, we obtain the following result relating the asymmetric operator $R_{p,0}$ and $R_{0,p}$ to the symmetric operator τ_δ in a biased space.

Theorem 3.6.3

$$||R_{0,p}f||_{2,\frac{1}{2}} = ||\tau_{\sqrt{\frac{1-p}{1+p}}} f||_{2,\frac{1-p}{2}}. \tag{3.8}$$

$$||R_{p,0}f||_{2,\frac{1}{2}} = ||\tau_{\sqrt{\frac{1-p}{1+p}}} f||_{2,\frac{1+p}{2}}. \tag{3.9}$$

Proof. The proof follows by combining Parseval's identity with the definition of τ_δ in Equation (3.5). ■

Theorem 3.6.3 does *not* directly imply hypercontractivity for $R_{p,0}$ under the uniform measure. Instead, it relates the l_2 norm of $R_{p,0}f$ to the norm of f in a biased measure space. Indeed there can be adversarial choices of f where the norm increases under perturbation by $R_{p,0}$.

We give an example. Consider the function $f : \{0,1\} \to \mathbb{R}$. Let $f(0) = 0$ and $f(1) = 1$. Then $R_{p,0}f(0) = p$ and $R_{p,0}f(1) = 1$. In particular $||R_{p,0}f||^2_{2,1/2} = \frac{p^2+1}{2} > ||f||_{2,1/2} = \frac{1}{2}$ which indicates no hypercontractivity.

We address this issue in two parts. Firstly, we use the biased Bonami-Beckner inequality (Theorem 3.5.2) to relate the right-hand side of Eq. (3.9) to the norm of f.

Theorem 3.6.4

$$||\tau_{\sqrt{\frac{1-p}{1+p}}} f||_{2,\frac{1-p}{2}} \leq ||f||_{1+\frac{1}{1-\log(1-p)},\frac{1-p}{2}}. \tag{3.10}$$

$$||\tau_{\sqrt{\frac{1-p}{1+p}}} f||_{2,\frac{1+p}{2}} \leq ||f||_{1+\frac{1}{1-\log(1-p)},\frac{1+p}{2}}. \tag{3.11}$$

Proof. We recall first the statement of Theorem 3.6.3:

$$||R_{p,0}f||_{2,\frac{1}{2}} = ||\tau_{\sqrt{\frac{1-p}{1+p}}} f||_{2,\frac{1+p}{2}}.$$

We combine this with the biased hypercontractivity claim of Theorem 3.5.2 which states:

$$||\tau_\delta f||_{2,\bar{p}} \leq ||f||_{1+\frac{1-\bar{p}}{\bar{p}\lfloor\log\frac{1}{\bar{p}}\rfloor}\delta^2,\bar{p}}.$$

We note the $\bar{p}$ there is the smaller of the measures of 0 or 1 in the product space. Hence we plug in $\bar{p} = \frac{1-p}{2}$ and $\delta = \sqrt{\frac{1-p}{1+p}}$ to obtain:

$$\begin{aligned}
&1+\delta^2 \frac{1-\bar{p}}{\bar{p}\log\frac{1}{\bar{p}}} \\
=&1+\left(\sqrt{\frac{1-p}{1+p}}\right)^2 \left(1-\frac{1-p}{2}\right)/\left(\frac{1-p}{2}\log\left(1/\frac{1-p}{2}\right)\right) \\
=&1+\left(\frac{1-p}{1+p}\right)\left(\frac{1+p}{1-p}\right)/\frac{1}{\log\frac{2}{1-p}} \\
=&1+\frac{1}{1-\log(1-p)}.
\end{aligned}$$

The second result claimed in the theorem statement follows almost identically. ■

The second and final part of the argument is to relate the norms of f in the unbiased and biased spaces. Recall that our ultimate aim is to bound $\|R_{p,0}f\|_{2,\frac{1}{2}}$ by $\|f\|_{1+\frac{1}{1-\log(1-p)},\frac{1}{2}}$. Let us limit $f : \{0,1\}^d \to \{0,1\}$ to take its support from the lower half of the Hamming cube, which we stipulate as $L = \{x : \sum_i x_i \leq \frac{1}{2}d\}$. We can define the upper half of the Hamming cube L analogously. Whenever we refer to a function f_U, this will be understood to have support only on the upper half of the Hamming cube, whereas f_L will have support only on the lower half.

Theorem 3.6.5 *For any parameters* $\delta > 1$ *and* $\frac{1}{2} \leq p \leq 1$, *we have:*

$$\|f_L\|_{\delta,p}^{\delta} \leq \|f_L\|_{\delta,\frac{1}{2}}^{\delta}. \tag{3.12}$$

And for $0 \leq p \leq \frac{1}{2}$, *we have:*

$$\|f_U\|_{\delta,p}^{\delta} \leq \|f_U\|_{\delta,\frac{1}{2}}^{\delta}. \tag{3.13}$$

Proof. The first inequality follows because points in the lower half of the hypercube have larger measure under the uniform distribution than under the p-biased distribution for $p > \frac{1}{2}$. The second claimed inequality follows by symmetry. ■

By Theorems 3.6.3, 3.6.4 and 3.6.5 we finally obtain that:

Theorem 3.6.6

$$\|R_{0,p}f_U\|_{2,\frac{1}{2}} \leq \|f_U\|_{1+\frac{1}{1-\log(1-p)},\frac{1}{2}}. \tag{3.14}$$

$$\|R_{p,0}f_L\|_{2,\frac{1}{2}} \leq \|f_L\|_{1+\frac{1}{1-\log(1-p)},\frac{1}{2}}. \tag{3.15}$$

We will find the following asymptotic form of our result useful, and indeed this is our main tool employed in Section 3.8.

Corollary 3.6.1

$$\|R_{0,p}f_U\|_{2,\frac{1}{2}} \leq \|f_U\|_{2-p\log e+O(p^2),\frac{1}{2}}. \tag{3.16}$$

$$\|R_{p,0}f_L\|_{2,\frac{1}{2}} \leq \|f_L\|_{2-p\log e+O(p^2),\frac{1}{2}}. \tag{3.17}$$

Proof. By employing the Taylor expansions of $\log(1-x)$ and $1/(1-x)$. ■

We can now generalize to the case of R_{p_1,p_2}. Let $p_1 \geq p_2$, and define f_L as before. We do not use the following theorem in the remainder of this chapter, but we include it for the interested reader who seeks a complete statement of the hypercontractivity result.

Theorem 3.6.7 *For $p_1 \geq p_2$, and both $p_1, p_2 \leq \frac{1}{4}$, we have that:*

$$\|R_{p_1,p_2}f_L\|_{2,\frac{1}{2}} \leq \|f_L\|_{1+(1-2p_2)^2/\left(1-\log\left(\frac{1-p_1-p_2}{1-2p_2}\right)\right),\frac{1}{2}}. \tag{3.18}$$

Proof. First note by Theorem 3.6.1, we have $R_{p_1,p_2} = T_{p_2}R_{\frac{p_1-p_2}{1-2p_2},0}$. Now let $p = \frac{p_1-p_2}{1-2p_2}$. Recalling Theorem 3.6.2 and that $T_{p_2} = \tau_{1-2p_2}$, we obtain

$$\|T_{p_2}R_{p,0}f_L\|_{2,\frac{1}{2}} = \|\tau_{(1-2p_2)}\tau_{\sqrt{\frac{1-p}{1+p}}}f_L\|_{2,\frac{1+p}{2}}. \tag{3.19}$$

Now using the fact that $\tau_a\tau_b = \tau_{ab}$, and by a similar calculation as in Theorem 3.6.4 for hypercontractivity in a biased measure space, we obtain

$$\|\tau_{(1-2p_2)\sqrt{\frac{1-p}{1+p}}}f_L\|_{2,\frac{1+p}{2}} \leq \|f_L\|_{1+\frac{(1-2p_2)^2}{1-\log(1-p)},\frac{1+p}{2}}. \tag{3.20}$$

We can combine equation 3.20 with Theorem 3.6.5 to get:

$$\|T_{p_2}R_{p,0}f_L\|_{2,\frac{1}{2}} \leq \|f_L\|_{1+\frac{(1-2p_2)^2}{1-\log(1-p)},\frac{1}{2}}. \tag{3.21}$$

Now substituting back the value of $p = \frac{p_1-p_2}{1-2p_2}$ into equation 3.21 we obtain the claimed result. ■

3.7 Hard input distributions for the Bregman cube

We now describe the construction of a hard input distribution for the Bregman cube. The key properties of this distribution will be that a query point will (in expectation) either have a near neighbor within distance $O(\varepsilon d)$, or will not have any neighbor closer than $\Omega(\mu d)$. Note that in contrast, the corresponding gap distribution for the Hamming cube via the *symmetric* noise operator has a gap of $O(\varepsilon d)$ for the nearest neighbor versus $\Omega(d)$ for the second nearest neighbor. Finally, for the purposes of our result and this paper we will assume $\mu \geq \frac{1}{\varepsilon}$.

3.7.1 Generating our input and query on the cube

Define a random perturbation $\nu : \{0,1\}^d \to \{0,1\}^d$ as a random binary string $\nu_{p_1,p_2}(x)$ obtained by flipping any 0 bit in x to 1 with probability p_1 and a 1 bit to 0 with probability p_2. In what follows, assume that $p_1 = \varepsilon < \frac{1}{100}$ and $p_2 = \frac{\varepsilon}{\mu}$.

Uniformly at random pick n elements $S = \{s_1, s_2, \dots s_n\}$ from the set $L = \{x \in \{0,1\}^d$ s.t $\sum(x) \leq \frac{d}{2}\}$, which is the lower half of the Bregman cube. We first perturb S to obtain $P = \nu_{\varepsilon/\mu,\varepsilon}(S)$. We then perturb S in the opposite direction, to obtain $Q = \nu_{\varepsilon,\varepsilon/\mu}(S)$. We now assign P to be our data set and choose our query point q uniformly at random from Q.

Theorem 3.7.1 *Let q be the perturbation of $s \in S$, and $p \in P$ be the corresponding point of P. Then for $\mu = O\left(\frac{\varepsilon d}{\log n}\right)$ and with high probability (at least $1 - 1/poly(n)$):*

1. *$\forall p' \neq p$, $p' \in P$, $D(q,p') = \Omega(\mu d)$.*
2. *$D(q,p) = \Theta(\varepsilon d)$.*
3. *$\forall p' \neq p$, $p' \in P$, $\frac{D(q,p')}{D(q,p)} = \Omega(\mu/\varepsilon)$.*

Proof. We focus on the distance induced by a single bit, and multiply by d to get the overall distance (follows from each bit being chosen identically and independently).

For the first claim, we show that $D(q_i, p'_i) = \Omega(\mu)$ with high probability. To aid our argument, we define the Hamming weight of a point as $H(x) = \sum_i(x_i)$. Now Chernoff concentration bounds give us that if $C = \{x \in U | 0.5 \leq H(x) \leq 0.55\}$, then $|C|/|U| \geq 1 - e^{-\Omega(d)}$. Therefore each bit of randomly chosen $p' \in U$ is 0 with probability at least $0.5 - 1/\text{poly n}$. We obtain similarly that q_i is 1 with probability at least $0.5 - 1/\text{poly}(n)$ for ε smaller than a suitable choice of constant.

Since $D(0,1) = \mu$ and q is independent of p' for $p' \neq p$, we can argue now that $E[D(q,p')] = \Omega(\mu d)$. A standard Chernoff analysis shows that $D(q,p') = \Omega(\mu d)$ holds true for all $p' \neq p$ with high probability.

We consider now the second claim. Refer to the j-th bit of s , q and p as s^j, q^j and p^j, respectively, and recall again that $D(0,1) = \mu$ and $D(1,0) = 1$. Consider first the case where $s^j = 0$. Then,

$$\begin{aligned} E[D(q^j,p^j)|s^j = 0] =& \Pr[q^j = 1|s^j = 0]\Pr[p^j = 0|s^j = 0]D(1,0) + \\ & \Pr[q^j = 0|s^j = 0]\Pr[p^j = 1|s^j = 0]D(0,1) \\ =& \varepsilon\left(1-\frac{\varepsilon}{\mu}\right)D(1,0) + (1-\varepsilon)\left(\frac{\varepsilon}{\mu}\right)D(0,1) \\ =& \varepsilon\left(1-\frac{\varepsilon}{\mu}\right) + (1-\varepsilon)\left(\frac{\varepsilon}{\mu}\right)(\mu) \\ =& 2\varepsilon - \varepsilon^2 - \frac{\varepsilon^2}{\mu} = \Theta(\varepsilon). \end{aligned}$$

Similarly when $s^j = 1$, we have

$$\begin{aligned} E[D(q^j,p^j)|s^j = 1] =& \Pr[q^j = 1|s^j = 1]\Pr[p^j = 0|s^j = 1]D(1,0) + \\ & \Pr[q^j = 0|s^j = 1]\Pr[p^j = 1|s^j = 1]D(0,1) \\ =& \left(1-\frac{\varepsilon}{\mu}\right)(\varepsilon)D(1,0) + \frac{\varepsilon}{\mu}(1-\varepsilon)D(0,1) \\ =& \left(1-\frac{\varepsilon}{\mu}\right)(\varepsilon) + \frac{\varepsilon}{\mu}(1-\varepsilon)\mu \\ =& 2\varepsilon - \varepsilon^2 - \frac{\varepsilon^2}{\mu} = \Theta(\varepsilon). \end{aligned}$$

We show now these distances concentrate around the expectation. Recall the classic Chernoff bounds: given a collection of independent 0-1 random variables X_i and $X = \sum_i X_i$ such that u_X is the mean of X, then $Pr[|X - u_X| \geq \sigma u_X] \leq e^{\frac{-\sigma^2 u_X}{2}} + e^{\frac{-\sigma^2 u_X}{3}}$. Note that here that we can represent our distances in the form $Y = \sum_{i=1}^{d} Y_i$ and $Z = \sum_{i=1}^{d} Z_i$ where i is an index over the number of bits d and the probability of success is $\frac{\varepsilon}{\mu}(1-\varepsilon)$ and $\varepsilon\left(1-\frac{\varepsilon}{\mu}\right)$, respectively. (Or asymptotically ε and $\frac{\varepsilon}{\mu}$, respectively.) For Y and Z to concentrate around u_Y and u_Z, respectively, for all n points and suitable choice of constant σ, we clearly require $u_Y = \Omega(\log n)$ and $u_Z = \Omega(\log n)$. This requires $\frac{\varepsilon}{\mu}d = \Omega(\log n)$.

And finally, the third claim can be seen to follow directly from the first two. ■

We note that it can be shown even for arbitrarily large μ and some constant ε, that a "gap" of $\Omega(\frac{d}{\log n})$ can be achieved by setting $\mu' = d/\log n$ and applying perturbations $P = \nu_{\varepsilon/\mu',\varepsilon}(S)$ and $Q = \nu_{\varepsilon,\varepsilon/\mu'}(S)$, respectively.

3.8 Shattering a query

We are now ready to assemble the parts that make up the proof of Theorem 3.1.1. In this section, we show that a point "shatters": namely, that if we perturb a point by a little, then it is likely to go to many different hash buckets. In the next section, we will show that this implies an information-theoretic lower bound on the number of queries needed to recover the original point that generates the query.

We prove the shattering bound in two steps. In lemma 3.8.1 we show that if we fix any sufficiently small subset of the cube, then the set of points that are likely to fall into this subset under perturbation is small. Then in lemma 3.8.2, we use this lemma to conclude that for *any* partition of the space into sufficiently small sets (think of each set as the entries mapped to a specific hash table entry), any perturbed query will be sent to many of these sets (or equivalently, no entry contains more than a small fraction of the "ball" around the query).

As mentioned in Section 3.2, the structure of this section mirrors the argument presented by Panigrahy *et al.*(128). The difficulty is that we can no longer directly work with the (symmetric) operator T_ε, and so the analysis becomes more intricate.

We consider sets of points restricted to $L = \{x : \sum_i x_i \leq \frac{d}{2}\}$. Let A be a light cell, where we stipulate a light cell to be such that $|A| \leq a2^d$ for some small $0 < a < 1$. Define $\gamma_{y,\varepsilon,\frac{\varepsilon}{\mu}}(A) = \Pr[\nu_{\varepsilon,\frac{\varepsilon}{\mu}}(y) \in A]$. Let $B \subseteq L$ be the set of points for which a perturbation is likely to fall in A, i.e. $B = \{y \in L \mid \gamma_{y,\varepsilon,\frac{\varepsilon}{\mu}}(A) \geq a^{c_0\varepsilon}\}$ for some $0 < c_0 < 1$ to be chosen later.

We shall show that $|B| \leq 2^d a^{1+c_1\varepsilon}$, where once again $0 < c_1 < 1$ can be set later. To this purpose, we will use Taylor approximations to simplify the algebra to asymptotic behavior.

Lemma 3.8.1 *Let $A \subseteq \{0,1\}^d$ with $|A| \leq a.2^d$. Let $\varepsilon \in (0,1)$, $\mu \geq \frac{1}{\varepsilon}$ and $B = \{y \in L \mid \gamma_{y,\varepsilon,\frac{\varepsilon}{\mu}}(A) \geq a^{c_0\varepsilon}\}$. Then for suitable choices of constants c_0 and c_1 less than 1, and for sufficiently small ε,*

$$|B| < 2^d a^{1+c_1\varepsilon}. \tag{3.22}$$

Proof. Suppose on the contrary that $|B| > 2^d a^{1+c_1\varepsilon}$. By definition, for every $y \in B$, $Pr[v_{\varepsilon,\frac{\varepsilon}{\mu}}(y) \in A] \geq a^{c_0\varepsilon}$. Let Q_B denote the random variable obtained by picking an element from B uniformly at random and then applying $v_{\varepsilon,\frac{\varepsilon}{\mu}}$.

Now,

$$\begin{aligned}
Pr[Q_B \in A] &= \frac{2^d}{|B|}\langle R_{\varepsilon,\frac{\varepsilon}{\mu}} 1_B, 1_A\rangle && \text{(By definition of } R_{\varepsilon,\frac{\varepsilon}{\mu}}.\text{)}\\
&= \frac{2^d}{|B|}\langle T_{\frac{\varepsilon}{\mu}} R_{\left(\varepsilon-\frac{\varepsilon}{\mu}\right)/\left(1-2\frac{\varepsilon}{\mu}\right),0} 1_B, 1_A\rangle && \text{(By Theorem 3.6.1 .)}\\
&= \frac{2^d}{|B|}\langle \tau_{1-2\frac{\varepsilon}{\mu}} R_{\left(\varepsilon-\frac{\varepsilon}{\mu}\right)/\left(1-2\frac{\varepsilon}{\mu}\right),0} 1_B, 1_A\rangle. && \text{(By the definition of } \tau.\text{)}
\end{aligned}$$

For convenience, let $\delta_1 = \sqrt{\left(1 - \frac{\varepsilon-\frac{\varepsilon}{\mu}}{1-2\frac{\varepsilon}{\mu}}\right) \Big/ \left(1 + \frac{\varepsilon-\frac{\varepsilon}{\mu}}{1-2\frac{\varepsilon}{\mu}}\right)}$, $\delta_2 = 1 - 2\frac{\varepsilon}{\mu}$ and $p = \left(\varepsilon - \frac{\varepsilon}{\mu}\right) / \left(1 - 2\frac{\varepsilon}{\mu}\right)$.

We abuse notation slightly, and introduce the function:

$$1_B^{\left(\frac{1+p}{2}\right)} = \sum_{S\subset\{0,1\}^d} \hat{1_B}^{\frac{1+p}{2}}(S)\chi_S^{\frac{1}{2}}, \tag{3.23}$$

i.e., the Fourier coefficients in the uniform measure are the coefficients of 1_B in measure $\frac{1+p}{2}$. By Theorem 3.6.2, we can transform $R_{\left(\varepsilon-\frac{\varepsilon}{\mu}\right)/\left(1-2\frac{\varepsilon}{\mu}\right),0} 1_B$ as $\tau_{\delta_1} 1_B^{(\frac{1+p}{2})}$:

$$\begin{aligned}
Pr[Q_B \in A] &= \frac{2^d}{|B|}\langle \tau_{\delta_2}\tau_{\delta_1} 1_B^{(\frac{1+p}{2})}, 1_A\rangle\\
&= \frac{2^d}{|B|}\langle \tau_{\delta_1\delta_2} 1_B^{(\frac{1+p}{2})}, 1_A\rangle && \text{(Since } \tau \text{ is multiplicative.)}\\
&= \frac{2^d}{|B|}\langle \tau_{(\delta_1\delta_2)^{3/4}} 1_B^{(\frac{1+p}{2})}, \tau_{(\delta_1\delta_2)^{1/4}} 1_A\rangle. && \text{(Since } \tau \text{ can be distributed}\\
& && \text{in a dot product.)}
\end{aligned}$$

We now proceed using Cauchy Schwarz to upper bound the dot product as a product of two norms, Parseval's to claim $\|\tau_{(\delta_1\delta_2)^{3/4}} 1_B^{p'}\|_{2,\frac{1}{2}} = \|\tau_{(\delta_1\delta_2)^{3/4}} 1_B\|_{2,p'}$ and then hypercontractivity to upper bound each of the norms in biased and uniform measure space respectively. Setting $p' = \frac{1+p}{2}$,

$$
\begin{aligned}
Pr[Q_B \in A] &\leq \frac{2^d}{|B|} \|\tau_{(\delta_1\delta_2)^{3/4}} 1_B^{p'}\|_{2,1/2} \|\tau_{(\delta_1\delta_2)^{1/4}} 1_A\|_{2,1/2} \\
&= \frac{2^d}{|B|} \|\tau_{(\delta_1\delta_2)^{3/4}} 1_B\|_{2,p'} \|\tau_{(\delta_1\delta_2)^{1/4}} 1_A\|_{2,1/2} \\
&\leq \frac{2^d}{|B|} \|1_B\|_{1+\frac{p'}{(1-p')\log(1/(1-p'))}(\delta_1\delta_2)^{1.5},p'} \|1_A\|_{1+\sqrt{\delta_1\delta_2},1/2}.
\end{aligned}
$$

And finally, since $B \subset L$, by 3.6.5 we have $\|1_B\|_{1+\frac{p'}{(1-p')\log(1/(1-p'))}(\delta_1\delta_2)^{1.5},p'}$ will only increase if we measure the norm in uniform space instead of the biased space with $p' > \frac{1}{2}$:

$$
Pr[Q_B \in A] \leq \frac{2^d}{|B|} \|1_B\|_{1+\frac{p'}{(1-p')\log(1/(1-p'))}(\delta_1\delta_2)^{1.5},1/2} \|1_A\|_{1+\sqrt{\delta_1\delta_2},1/2}. \tag{3.24}
$$

On a high level now, our approach is simply to show that the power in the norm on both expressions is $2-\Omega(\varepsilon)$. This would show that the collision or intersection size $Pr[Q_B \in A] \leq \frac{2^d}{|B|}$ is much smaller than the product of the set sizes. To simplify these expressions, we focus now on $\frac{p'}{(1-p')\log(1/(1-p'))}(\delta_1\delta_2)^{1.5}$ and $\sqrt{\delta_1\delta_2}$. For $\sqrt{\delta_1\delta_2}$, some straightforward substitution and the assumption that $\mu \geq \frac{1}{\varepsilon}$ shows:

$$
\begin{aligned}
\sqrt{\delta_1\delta_2} &= \sqrt{\left(1-2\frac{\varepsilon}{\mu}\right)\sqrt{\frac{1-\varepsilon-\frac{\varepsilon}{\mu}}{1+\varepsilon-3\frac{\varepsilon}{\mu}}}} \\
&= 1-\frac{\varepsilon}{2}+O(\varepsilon^2).
\end{aligned}
$$

We come now to $\frac{p'}{(1-p')\log(1/(1-p'))}(\delta_1\delta_2)^{1.5}$. Simply plugging in the values for p', δ_1 and δ_2 now shows:

$$\begin{aligned}\frac{p'}{(1-p')\log(1/(1-p'))}(\delta_1\delta_2)^{1.5} &= \left(\frac{1+\varepsilon-3\frac{\varepsilon}{\mu}}{1-\varepsilon-\frac{\varepsilon}{\mu}}\right)^{1/4}\frac{\left(1-2\frac{\varepsilon}{\mu}\right)^{1.5}}{\left(1-\log\left(1-\left(\varepsilon-\frac{\varepsilon}{\mu}\right)/\left(1-2\frac{\varepsilon}{\mu}\right)\right)\right)}\\ &< \left(1+2\varepsilon+O(\varepsilon^2)\right)^{1/4}\frac{1}{1-\log\left(1-\varepsilon+O(\varepsilon^2)\right)}\\ &\le \left(1+\frac{\varepsilon}{2}+O(\varepsilon^2)\right)\left(1-\varepsilon\log e+O(\varepsilon^2)\right)\\ &< \left(1-\frac{\varepsilon}{2}+O(\varepsilon^2)\right).\end{aligned}$$

Or for sufficiently small ε, that $\frac{p'}{(1-p')\log(1/(1-p'))}(\delta_1\delta_2)^{1.5} = 1-\Omega(\varepsilon)$. We can now see the asymptotic behavior of the norms in Equation 3.24:

$$Pr[Q_B \in A] \le \frac{2^d}{|B|}\|1_B\|_{2-\Omega(\varepsilon),1/2}\|1_A\|_{2-\Omega(\varepsilon),1/2}. \tag{3.25}$$

Hence there exists a constant k, such that:

$$Pr[Q_B \in A] \le \frac{2^d}{|B|}\|1_B\|_{2-k\varepsilon,1/2}\|1_A\|_{2-k\varepsilon,1/2}.$$

Now let $k\varepsilon = 2\varepsilon'$, and also set $c_0\varepsilon$ and $c_1\varepsilon$ to be $\frac{\varepsilon'}{6}$. By the assumptions of our lemma, we have $\Pr[Q_B \in A] \ge a^{\frac{\varepsilon'}{6}}$. Recalling that $\|1_B\|_{2-2\varepsilon',\frac{1}{2}} = |B|^{1/(2-2\varepsilon')}$ and $\|1_A\|_{2-2\varepsilon',\frac{1}{2}} = |A|^{1/(2-2\varepsilon')}$, we obtain

$$\begin{aligned}Pr[Q_B \in A] &\le \frac{2^d}{|B|}\|1_B\|_{2-2\varepsilon',1/2}\|1_A\|_{2-2\varepsilon',1/2}\\ &= \frac{2^d}{|B|}\left(\frac{|B|}{2^d}\right)^{1/(2-2\varepsilon')}\left(\frac{|A|}{2^d}\right)^{1/(2-2\varepsilon')}\\ &= \left(\frac{|A|}{2^d}\right)^{1/(2-2\varepsilon')}\left(\frac{|B|}{2^d}\right)^{1/(2-2\varepsilon')-1}.\end{aligned}$$

Now recalling that $|A| \le a2^d$ and claiming for contradiction that $|B| \ge 2^d a^{1+\varepsilon'/6}$, we obtain

$$Pr[Q_B \in A] \le a^{1/(2-2\varepsilon')}a^{(1+\varepsilon'/6)(\frac{1}{2-2\varepsilon'}-1)} \le a^{\varepsilon'/6}.$$

However this is impossible, since by our lemma assumptions $Pr[Q_B \in A] \ge a^{\varepsilon'/6}$. Hence we must have that if $\Pr[Q_B \in A] \ge a^{\varepsilon'/6}$ then $|B| \le 2^d a^{1+\varepsilon'/6}$. Noting that $\varepsilon' = k\varepsilon$ and

k is some constant, our lemma follows. ■

The following lemma is now an easy consequence.

Lemma 3.8.2 *Let $A_1,\dots,A_m$ be partitions of $\{0,1\}^d$ and let $LC = \{i : \|A_i\| \le 2^d/\sqrt{m}\}$ be the set of light cells. Then:*

$$\Pr_{y\in LC}[\max_{i\in LC} \gamma_{y,\varepsilon,\frac{\varepsilon}{\mu}}(A_i) \ge m^{-\frac{c_0\varepsilon}{2}}] < m^{-\frac{c_1\varepsilon}{2}}. \tag{3.26}$$

Proof. Let $a_i = \frac{|A|}{2^d}$ and note $\sum_i a_i = 1$. By lemma 3.8.1 $\Pr_{y\in LC}[\gamma_{y,\varepsilon,\frac{\varepsilon}{\mu}}(A_i) \ge a_i^{c_0\varepsilon}] \le a_i^{1+c_1\varepsilon}]$. And we also have by the bound on light cells that $a_i^{c_0\varepsilon} \le m^{-\frac{c_0\varepsilon}{2}}$. Then by a union bound, we have that the desired probability is:

$$\Pr_{y\in LC}[\max_{i\in LC} \gamma_{y,\varepsilon,\frac{\varepsilon}{\mu}}(A_i) \ge m^{-\frac{c_0\varepsilon}{2}}] \le \sum_i \Pr_{y\in LC}[\gamma_{y,\varepsilon,\frac{\varepsilon}{\mu}}(A_i) \ge a_i^{c_0\varepsilon}]$$
$$\le a_i^{1+c_1\varepsilon} \le \max_i a_i^{c_1\varepsilon} \sum_i a_i \le \max_i a_i^{c_1\varepsilon} \le (\sqrt{m})^{-c_1\varepsilon}.$$

■

3.9 Alternate construction

For the sake of completeness, we give here an alternate, simpler construction inspired by Panigrahy, Talwar and Weider (128) for lower bounds on partial match which yields a weaker shattering result. Let H_δ denote the set of points in $\{0,1\}^d$ with exactly δd 1's. Each entry in the dataset P is chosen uniformly and independently in H_δ. Once the data structure is built, we sample a query point q around a random dataset point p as follows: first we convert the 1's in p to 0's. Then of the remaining bits, we convert an additional $\left(\frac{d}{2} - \delta d\right)$ 0's to 1's at random so that the resulting query point q has exactly $d/2$ 0's and $d/2$ 1's.

Note here that $D(q,p)$ is $\mu\delta d + \frac{d}{2} - \delta d$, by the straightforward definition we have that $D(0,1) = \mu$ and $D(1,0) = 1$. Similarly, for $D(q,p')$ where $p' \neq p$ we have that $E[D(q,p')] = \Omega(\mu d)$. Setting δ to be $\frac{1}{\mu}$ we get that $D(q,p) = O(d)$, whereas $E[D(q,p')] = \Omega(\mu d)$. Using Chernoff bounds along the line of Section 3.7 we obtain that for $\mu = O(d/\log n)$ this bound in expectation is true with high probability for every $p' \neq p$ and so the ratio between the second and first nearest neighbor distance is $O(\mu)$.

The authors of (128) show that this distribution satisfies shattering properties, saving us the need to rederive their result. In the following, define v_p to be the distribution obtained from a query point q from our process, and let F be an arbitrary hash function from $\{0,1\}^d$ to our set of cells $[m]$. Let A_i denote the set of query points that are mapped to a cell i by F.

Lemma 3.9.1 *Let $A_1,\dots,A_m$ be partitions of $\{0,1\}^d$ and let $LC = \{i : \|A_i\| \le 2^d/\sqrt{m}\}$ be the set of light cells. There exists a constant c_0 such that:*

$$\Pr_p[\max_{i\in LC} v_p(A_i) \ge m^{-\frac{c_0}{2}}] < m^{-\frac{c_0}{2}}. \tag{3.27}$$

-

This is a weaker form of lemma 3.8.2 without dependence on ε and with a smaller first to second nearest neighbor gap of $\Omega(\mu)$ rather than $\Omega\left(\frac{\mu}{\varepsilon}\right)$. This can still obtain a cruder version of our overall result Theorem 3.10.1, which is asymptotically equivalent in terms of μ for constant ε.

3.10 From hypercontractivity to a lower bound

First we lay out the notation and preliminaries of our argument, including our model. An (m,r,w) nonadaptive algorithm is an algorithm in which given n input points $p_1,\dots,p_n$ in $\{0,1\}^d$ we prepare in preprocessing a table T which consists of m words, each w bits long. Given a query point q, the algorithm queries the table at most r times and let $l_1,l_2,\dots,l_r$ denote the set of indices looked up by the algorithm. For every $t \le r$, the location of the t-th probe, $l_t = l_t(q)$ depends only upon the query point q and no t upon the content that was read in the previous queries. In other words, the functions $l_1,l_2,\dots,l_r$ depend on q only. In this section, we show a time-space cell probe lower bound. We mostly use the machinery given in (128), but for the sake of explication and clarity we reproduce the argument and expand some of the steps.

The high level idea of (128), and also work by Larsen (97) and Wang and Yin (153) is "cell sampling" of a data structure T on input P. If T resolves a large number of queries which do not err in a few probes, then a small sample of the cells will resolve many queries with high probability. Now if such a sample of cells can be described in fewer bits than the information complexity of these queries, then there would be a contradiction. This lower bounds the size of T.

We prepare our dataset and query point as described in Section 3.7. We first pick a set of n elements $S = \{s_1, s_2, \dots s_n\}$ from the lower half of the Bregman cube. More precisely, we pick from the set of strings $U = \{x \in \{0,1\}^d \text{ s.t} \sum(x) \le \frac{d}{2}\}$. We first perturb S to obtain $P = \nu_{\frac{\varepsilon}{\mu}, \varepsilon}(S)$. We then pick i uniformly at random from $[1 \dots n]$ and set q to be $\nu_{\varepsilon, \frac{\varepsilon}{\mu}}(s_i)$. In what follows, we will let s_i denote the point of S which is perturbed to obtain q and p_i be the corresponding point in P. Theorem 3.7.1 guarantees that $D(q, p_i) = \Theta(\varepsilon d)$ whereas $D(q, p_j) = \Omega(\mu d)$ for $j \neq i$ with high probability. Hence recovering a $\frac{\mu}{\varepsilon}$ nearest neighbor to q from P is equivalent to recovering p_i exactly. The table is populated in preprocessing based on P as the ground set.

Our assumption on the correctness of the algorithm is that when the input is sampled in this way, then for each i with probability $\frac{1}{2}$ over the choice of s_i and p_i, with probability $\frac{2}{3}$ over the choice of q_i the algorithm can reconstruct p_i. We can fix the coin tosses of the algorithm and assume the algorithm is deterministic, and we assume the query algorithm is given access to not only P but also S.

The main result of this section is as follows.

Theorem 3.10.1 *A (m, r, w) nonadaptive algorithm to recover a $O(\frac{\mu}{\varepsilon})$ nearest neighbor to a query point q with constant probability has $mw \ge \Omega(\varepsilon d n^{1+\Omega(\frac{\varepsilon}{r})})$ as long as w is polynomial in $d, \log n$.*

Note that Theorem 3.10.1 yields Theorem 3.1.1 as a corollary, by setting $c' = \mu/\varepsilon$.

Proof. We recall some standard information theoretic notation. Let $H(A)$ be the entropy of a distribution A, and let $I(A, B)$ be the mutual information of two distributions, such that $I(A, B) = H(A) - H(A|B) = H(B) - H(B|A)$. We have the following well-known rules for simplifying expressions:

1. $I(X, Y) = I(Y, X)$.

2. $I(X; Y|Z) = H(X|Z) - H(X|Y, Z)$.

3. $I(X; Y) = 0$ if X and Y are independent random variables.

4. $I(Z; X, Y) = I(Z; X) + I(Z|X; Y)$. If X and Y are independent.

5. For n independent random variables X_1 through X_n, we have
$I(Y; X_{1:n}) = \sum_i I(Y|X_{1:i-1}; X_i)$.

Now let L be a set of k locations picked at random from the table, where k is a parameter to be fixed later, and $T[L] = \{T[i] : i \in L\}$ be the corresponding set of words.

Claim 3.10.1 $I(T[L]; p_i|S, L, q) = I(T[L]; p_i|S, L)$.

Proof. When S and L are fixed, p_i is independent of q, and $T[L]$ is determined by P. So we may simply drop q here. ■

For the remainder of our proof, for ease of notation we will implicitly assume S and L are known to the algorithm and p_i is conditioned on them.

Claim 3.10.2 $\sum_{i=1}^{n} I(T[L]; p_i) \leq H(T[L]) \leq wk$.

Proof. First note that $H(T[L]) \leq wk$ simply from the fact that there are only wk bits in $T[L]$, and $H(T[L]) \leq wk$. For $\sum_{i=1}^{n} I(T[L]; p_i)$, we note that $I(T[L]; p_i) \leq I(T[L]|p_1, p_2, \dots p_{i-1}; p_i)$. To note this, we see the direct comparison:

$$\begin{aligned}
& I(T[L]; p_i) - I(T[L]|p_1, p_2, \dots p_{i-1}; p_i) \\
=& H(p_i) - H(p_i|T[L]) - H(p_i) + H(p_i|T[L], p_1, p_2, \dots p_{i-1}) \\
=& H(p_i|T[L], p_1, p_2, \dots p_{i-1}) - H(p_i|T[L]) \\
\leq& 0.
\end{aligned}$$

Hence $\sum_{i=1}^{n} I(T[L]; p_i) \leq \sum_{i=1}^{n} I(T[L]|p_1, p_2, \dots p_{i-1}; p_i)$. Applying the chain rule, this latter quantity equals $I(T[L]; p_1, p_2, \dots p_n) \leq H(T[L])$. ■

Taking expectations on both sides w.r.t. L and S, we have:

$$\sum_{i=1}^{n} E_{L,S}[I(H(T[L]; p_i|S, L)] \leq wk. \tag{3.28}$$

Set $k = m/n^{\Omega(\frac{\varepsilon}{r})}$. Our goal therefore is to show that $E_{L,S}[I(T[L]; p_i|S, L)] \in \Omega(\varepsilon d)$, as this would immediately imply the theorem.

We will prove the slightly stronger result that $I(T[L]; p_i|S, L) = \Omega(\varepsilon d)$. Suppose that our algorithm can reconstruct p_i given $T[L]$ with constant probability α. We can lower bound $H(p_i|S, L)$ as follows. Note that it suffices to examine $H(p_i|s_i)$. In each 1-bit of s_i there is an induced entropy of $-\varepsilon \log \varepsilon - (1-\varepsilon)\log(1-\varepsilon) \geq -\varepsilon \log(1-\varepsilon) - (1-$

$\varepsilon)\log(1-\varepsilon) = -\log(1-\varepsilon) = \Omega(\varepsilon)$. Similarly, in each 0-bit of s_i, the entropy is $\frac{\varepsilon}{\mu}\log\frac{\varepsilon}{\mu} + (1-\frac{\varepsilon}{\mu})\log(1-\frac{\varepsilon}{\mu}) = \Omega(\frac{\varepsilon}{\mu})$. Note that $H(p_i|S)$ is at least $\Omega(\varepsilon d)$, just from the entropy of the 0-bits of s_i.

We now use the following simplification of Fano's inequality, which in slightly different form was described by Regev (135):

Lemma 3.10.1 *Let X be a random variable, and let $Y = g(X)$ where $g(\cdot)$ is a random process. Assume the existence of a procedure f that given $y = g(x)$ can reconstruct x with probability p. Then*

$$I(X;Y) \geq pH(X) - H(p).$$

Proof. The proof follows from the usual statement of Fano's inequality in terms of the conditional entropy $H(X|Y)$. ■

Consider now the mutual information $I(T[L];p_i|S,L)$. By Fano's inequality and the lower bound on $H(p_i|S,L)$, the desired lower bound on $I(T[L];p_i|S,L)$ will follow if we can present a procedure that with constant probability will reconstruct p_i from $T[L]$ given S and L. Note that i is fixed in this process.

Denote by $l_j(q)$ the location of the j-th query when the query point is q. We write $l_{[r]}$ to denote $l_1(q) \cup \ldots \cup l_r(q)$. We say a point q_j is *good* for p_j if $D(q_j, s_j)$ is at most εd and p_j can be reconstructed from q_j, s_j and $T[L_{[r]}(q_j)]$ (the set of table lookups on q) with constant probability. Let Q_i denote the set of points which are good for p_i. We make the following useful observations about Q_i.

Lemma 3.10.2

1. *With probability at least $\frac{1}{2}$ over the choice of s_i, we have* $\Pr[\nu_{\varepsilon,\frac{\varepsilon}{\mu}}(s_i) \in Q_i] \geq \frac{2}{3}$ *(where the latter probability is taken over the perturbation).*

2. $|Q_i| \geq n^{\varepsilon}$ *with probability at least $\frac{1}{2}$.*

Proof. The correctness of the algorithm and definition of Q_i implies the first claim, i.e., for a large fraction of the points in S, with constant probability the perturbed point generated $q_i = \nu_{\varepsilon,\frac{\varepsilon}{\mu}}(s_i)$ will be a *good* point for reconstructing p_i.

For the second claim observe that there are $\Omega(d)^{\Omega(\varepsilon d)}$ points q within at most εd distance of s_i. Since our algorithm reconstructs p_i with constant probability, the majority of these points must be good for p_i. Since $n \leq 2^d$, hence $n \leq d^d$ as well and we must have that $|Q_i| \geq n^{\varepsilon}$ with probability at least $\frac{1}{2}$ for sufficiently large d. ■

Now define A^t_j to be the set of $q \in \{0,1\}^d$ such that $j = l_t(q)$. In the nonadaptive domain we can assume that all cells are light, i.e., w.l.o.g for every $1 \leq j \leq m$ and $1 \leq t \leq r$ it holds that $|A^t_j| \leq \frac{2^d}{m}$. The reason is that a cell j for which A^t_j is large (for some t) could be split into $|A^t_j|/\frac{2^d}{m}$ light cells with the total number of new cells bounded by m. Our argument now analyzes *shattered* points, which fall into any light cell with very low probability after perturbation.

Definition 6 *A point s_i is shattered if $\max_{j,t}[\Pr \nu_{\varepsilon,\frac{\varepsilon}{\mu}}(s_i) \in A^t_j] \leq n^{\frac{-c_1\varepsilon}{2}}$ for suitable choice of constant c_1.*

Note first that $mw \geq \Omega(nd)$, just because the table needs enough space to store all the points to report. Since w is upper bounded by a constant polynomial in d, we have that m is polynomial in n. Now for every nonadaptive algorithm, our isoperimetry bound implies that the probability over the choice of s_i that s_i is not shattered is at most $m^{\frac{-c_1\varepsilon}{2}}$ by lemma 3.8.2. This probability is also at most $n^{-c_2\varepsilon}$ for suitable choice of constant c_2. Hence with probability at least $\frac{1}{3}$, it holds that s_i is shattered and $|Q_i| \geq n^{\varepsilon}$.

We show that in such a case with constant probability there exists a $q^* \in Q_i$, such that $l_{[r]} \subset L$, i.e., all the table lookups for point q^* are contained in $T[L]$. Our procedure will simply sample points from $\nu_{\frac{\varepsilon}{\mu},\varepsilon}(s_i)$ until it finds such a point $q^* \in Q_i$. Then by definition of good points, we can reconstruct p.

Since s_i is shattered it holds that there are at most $rQ_i/n^{c_2\varepsilon}$ points in Q that are mapped to the same cell. Now since $|Q_i| \geq n^{\varepsilon}$ we have that there are at least $n^{c_2\varepsilon}/r$ different good q which map into different rows of the table. Each of these q has all its probe locations map into $T[L]$ with probability at least $\frac{r}{n^{c_2\varepsilon}}$ so with probability $\geq \frac{1}{2}$ at least one point maps into $T[L]$ and we can reconstruct p_i thereby. ■

3.11 Lower bounds via classical problems on the Hamming cube

In this section we will lay out lower bounds on the Bregman approximate near neighbor via reductions from the Hamming cube. The first reduction follows from "symmetrizing" our input by a simple bit trick and hence is independent of any asymmetry or μ parameter. The second reduction follows from the observation that for large enough μ ($\mu \geq cd$ for suitable constant c), only one direction of bit flip essentially determines the nearest neighbor. Under this regime, Partial Match can be reduced to our problem and hence the latter inherits the corresponding lower bounds.

3.11.1 A lower bound via ℓ_1

We start by defining a combinatorial structure isomorphic to the Hamming cube, which we term the *pseudo-Hamming-cube*.

Definition 7 *Given any uniform Bregman divergence $D_\phi : \mathbb{R}^{2d} \times \mathbb{R}^{2d} \to \mathbb{R}$, a pseudo-Hamming-cube $C_{2\phi} \subset \mathbb{R}^{2d}$ is a set of 2^d points with a bijection $f_\phi : \{0,1\}^d \to C_\phi$ and a fixed constant $c_0 \in \mathbb{R}$ so that $D_\phi(f_\phi(x), f_\phi(y)) = c_0 ||x-y||_1$, $\forall x,y \in \{0,1\}^d$.*

Given a specific uniform Bregman divergence D_ϕ, we can compute a corresponding pseudo-Hamming cube explicitly.

Lemma 3.11.1 *For any uniform $D_\phi : \mathbb{R}^{2d} \times \mathbb{R}^{2d} \to \mathbb{R}$, there exists a pseudo-Hamming-cube $C_\phi \subset \mathbb{R}^{2d}$ and a suitable constant $c_0 \in \mathbb{R}$.*

Proof. Recall that for uniform D_ϕ, we have that $\phi(x) = \sum_{i=1}^{d} \phi_{\mathbb{R}}(x_i)$. Pick arbitrary a,b in the domain of $\phi_{\mathbb{R}}$ and hence obtain the two ordered pairs (a,b) and (b,a) in $\mathbb{R}^2$. Now stipulate $C_\phi = \{x_1 \times x_2 \ldots x_d \text{ s.t } x_i \in \{(a,b),(b,a)\}\} \subset \mathbb{R}^{2d}$. Note that $|C_\phi = 2^d|$. We define the isomorphism $f_\phi : \{0,1\}^d \to C_\phi$ by using the helper function $\bar{f} : \{0,1\} \to \{(a,b),(b,a)\}$ where $\bar{f}(0) = (a,b)$ and $\bar{f}(1) = (b,a)$. Now we state for $x \in \{0,1\}^d$,

$$f_\phi(x) = \bar{f}(x_1) \times \bar{f}(x_2) \times \ldots \bar{f}(x_d).$$

The insight is that the component $D_{\phi_{\mathbb{R}}}$ of D_ϕ on $\bar{f}(x_i)$ and $\bar{f}(y_i)$ between any two $x,y \in C_\phi$ is symmetrized, as

$$D_{\phi_{\mathbb{R}\times\mathbb{R}}}((a,b),(b,a)) = D_{\phi_{\mathbb{R}\times\mathbb{R}}}((b,a),(a,b)) = D_{\phi_{\mathbb{R}}}(b,a) + D_{\phi_{\mathbb{R}}}(a,b). \qquad (3.29)$$

Direct computation now shows that $\forall x, y \in \{0,1\}^d$, we have that:

$$D_\phi(f_\phi(x), f_\phi(y)) = \left(D_{\phi_\mathbb{R}}(b,a) + D_{\phi_\mathbb{R}}(a,b)\right) ||x-y||_1. \qquad (3.30)$$

This completes our proof, with $c_0 = D_{\phi_\mathbb{R}}(b,a) + D_{\phi_\mathbb{R}}(a,b)$. ■

Now that we have defined a distance preserving mapping from the ℓ_1 Hamming cube in $\mathbb{R}^d$ with a D_ϕ pseudo-Hamming-cube in $\mathbb{R}^{2d}$, it follows that LSH and ANN lower bounds for the ℓ_1 Hamming cube now transfer over to D_ϕ. In particular (and we list just a few here):

- A cell probe lower bound of $\Omega(\frac{d}{\log n})$ queries for any randomized algorithm that solves *exact* nearest neighbor search on the Bregman cube in polynomial space and word size polynomial in d, $\log n$ via (35).
- A cell probe lower lower bound of $\Omega(d^{1-o(1)})$ queries for any deterministic algorithm that returns a constant factor approximation to the Bregman nearest neighbor via (103).
- A cell probe lower bound of $\Omega\left(\frac{\log\log d}{\log\log\log d}\right)$ for any randomized algorithm that returns a constant factor approximation to the Bregman nearest neighbor via (41).

3.11.2 A lower bound via Partial Match

We consider the following version of the *Partial Match problem*: given point set $P \subset \{0,1\}^d$ and $q \in \{0,1\}^d$ determine whether q dominates any point in P, i.e., output YES if $\exists p \in P$, s.t. $q_i \geq p_i$, $\forall 1 \leq i \leq d$ and NO otherwise. This is known by folklore to be equivalent to the more popular statement of the problem where $q \in \{0,1,*\}^d$ and we must determine whether q matches any string in P (and where $*$ can match anything). See for instance (130) for a statement of this equivalence.

We construct our reduction from an instance of the Partial Match problem to a Bregman approximate near neighbor instance as follows. Set $\mu = 2d+1$, and define D as given in Section 3.4.2. It is clear that $D(q,x) \leq d$ if and only if x is a Partial Match for q and that $D(q,x) \geq 2d+1$ otherwise. We immediately obtain the following:

Theorem 3.11.1 *Let $P \subset \{0,1\}^d$, $q \in \{0,1\}^d$ and $\mu \geq 2d+1$. Any algorithm which returns a 2-approximate Bregman nearest neighbor $p \in P$ to query point q with constant probability will solve the Partial Match problem with constant probability.*

This simple reduction immediately implies that for $\mu \geq \Omega(d)$ and a constant factor approximate nearest neighbor, our problem inherits the lower bounds on Partial Match. These include, but are not limited to, the following:

- An $\Omega\left(\frac{d}{\log d}\right)$ lower bound on the number of cell probes for any randomized near-linear space algorithm, via a result by Patrascu and Thorup (131).
- An $\Omega\left(2^{\Omega(d/r)}\right)$ lower bound on the space required by any randomized algorithm which uses r queries, via the result by Patrascu (130).

3.11.3 Comparisons and comments on the behavior of the lower bounds with μ

It is worthwhile to contrast the bounds obtained from the simple reduction of Section 3.11.2 to our more involved main result of Theorem 3.1.1.

- The simple reduction from Partial Match in Section 3.11.2 implies a lower bound only for $\mu \geq \Omega(d)$, whereas our main result of Theorem 3.1.1 holds for far smaller asymptotic ranges of μ, upto $O\left(\frac{d}{\log n}\right)$.
- For $\mu = \Omega\left(\frac{d}{\log n}\right)$, Theorem 3.1.1 already implies a lower bound of $dn^{1+\Omega(d/r\log n)} = 2^{\Omega\left(\frac{d}{r}\right)}$ on the space required by a (nonadaptive) data structure that uses only r queries. The reduction from Partial Match achieves this same space lower bound at a far higher $\mu = \Omega(d)$ via the result of (130).
- For the strictly linear space ($O(nd)$) regime, Theorem 1.1 implies a lower bound of $\Omega(d)$ on number of queries required for our Bregman ANN. (Set $dn^{\Omega(1+d/r\log n)} = O(nd)$, to get that $n^{\Omega(d/r\log n)} = 2^{\Omega(d/r)}$ must be $O(1)$ and hence $r = \Omega(d)$.) This is in fact *stronger* than any lower bound for number of queries known on Partial Match (the best we are aware of is $\Omega(d/\log d)$ by (131) under any polynomial space).

As such, the parameter μ appears a natural way to interpolate between the well known problems of approximate nearest neighbor under ℓ_1 and the Partial Match problem. At constant values of μ, a constant factor ANN under D corresponds to a constant factor ANN under ℓ_1, whereas at $\mu \geq \Omega(d)$, a constant ANN under D solves Partial Match. The point

$\mu = \Omega(\frac{d}{\log n})$ then appears to be an interesting point along this interpolation where the space lower bounds are already asymptotically equal to those for Partial Match, with the qualifier that Theorem 3.1.1 holds for nonadaptive data structures. The question remains open of whether Partial Match lower bounds themselves could in fact be strengthened by a reduction to Bregman ANN.

CHAPTER 4

EMBEDDINGS AND DIMENSIONALITY REDUCTION FOR INFORMATION THEORETIC DISTANCES

The space of *information distances* includes many distances that are used extensively in data analysis. These include the well-known Bregman divergences, the α-divergences, and the f-divergences. In our previous two chapters, we explored fairly generic Bregman divergences, and the tradeoff between upper and lower bounds attainable for ANN. Here we show that by focusing on the specific geometry of a restricted subset of these measures, we can devise algorithmic primitives not otherwise generally attainable.

In particular, in this chapter we focus on a subclass of the f-divergences that admit embeddings into some (possibly infinite-dimensional) Hilbert space, with a specific emphasis on the JS divergence. These divergences are used in statistical tests and estimators (31), as well as in image analysis (132), computer vision (82, 105), and text analysis (61, 63). They were introduced by (54), and, in the most general case, also include measures such as the Hellinger, JS, and χ^2 divergences (here we consider a symmetrized variant of the χ^2 distance).

To work with the geometry of these divergences effectively at scale and in high dimensions, we need algorithmic tools that can provide provably high quality approximate representations of the geometry. The techniques of *sketching*, *embedding*, and *dimensionality reduction* have evolved as ways of dealing with this problem, as discussed briefly in Chapter 1. Unfortunately, while these tools have been well developed for norms like ℓ_1 and ℓ_2, we lack such tools for information distances. This is not just a theoretical concern: information distances are semantically more suited to many tasks in machine learning, and building the appropriate algorithmic toolkit to manipulate them efficiently would expand greatly the places where they can be used.

4.1 Our contributions

Our main result is a *structure-preserving* dimensionality reduction for information distances, where we wish to preserve not only the distances between pairs of points (distributions), but also the underlying simplicial structure of the space, so that we can continue to interpret coordinates in the new space as probabilities. The notion of a structure-preserving dimensionality reduction is implicit when dealing with normed spaces (since we always map a normed space to another), but requires an explicit mapping when dealing with more structured spaces. We prove an analog of the classical JL–lemma :

Theorem 4.1.1 *Given a set of n points in the high-dimensional simplex Δ_d, there exists a structure-preserving dimensionality reduction to a low-dimensional simplex Δ_k, that preserves the Jenson-Shannon, Hellinger, and χ^2 divergences, up to a factor of $(1+\varepsilon)$ when $k = O((\log n)/\varepsilon^2)$.*

The theorem extends to "well-behaved" f-divergences (See Section 4.6) for a precise definition). Moreover, the dimensionality reduction is constructive for any divergence with a finite-dimensional kernel (such as the Hellinger divergence), or an infinite-dimensional Kernel that can be sketched in finite space, as we show is feasible for the JS and χ^2 divergences. We note f-divergences that are not well-behaved (e.g., ℓ_1) do not admit a similar dimensionality reduction (13, 38, 101, 135). Establishing the above result requires us to show how to embed information distances into ℓ_2^2; this result will allow us to prove *sketching* results as well.

Theorem 4.1.2 *A set of points P in Δ_d under the Jensen–Shannon(JS) or χ^2 divergence can be deterministically embedded into $O(\frac{d^2}{\varepsilon}\log\frac{d}{\varepsilon})$ dimensions under ℓ_2^2 with ε additive error. The same space bound holds when sketching JS or χ^2 in the* aggregate *stream model.*

Corollary 4.1.1 *Assuming polynomial precision, an AMS sketch for Euclidean distance can reduce the dimension to $O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon}\log d\right)$ for a $(1+\varepsilon)$ multiplicative approximation in the aggregate stream setting.*

Theorem 4.1.3 *A set of points P under the JS or χ^2 divergence can be embedded into $\ell_2^{\bar{d}}$ with $\bar{d} = O\left(\frac{n^2 d^3}{\varepsilon^2}\right)$ with $(1+\varepsilon)$ multiplicative error.*

For both the techniques, applying the Euclidean JL–lemma can further reduce the dimension to $O\left(\frac{\log n}{\varepsilon^2}\right)$ in the offline setting.

These results are significant because (73) showed that these divergences cannot be sketched even up to a constant factor in sublinear space under a related streaming model (where the distributions are presented as incremental updates to individual probabilities, as opposed to all at once in the model we consider). In fact our result resolves affirmatively an open problem posed by them. We also give the first results on sketching of infinite-dimensional kernels, a challenge posed in (25).

The unifying approach of our three results—sketching, embedding into ℓ_2^2, and dimensionality reduction—is to analyze carefully the infinite-dimensional kernel of the information divergences. Quantizing and truncating the kernel yields the sketching result, and sampling repeatedly from it produces an embedding into ℓ_2^2. Finally given such an embedding, we show how to perform dimensionality reduction by proving that each of the divergences admits a region of the simplex where it is similar to ℓ_2^2. We point out that to the best of our knowledge, this is the first result that explicitly uses the kernel representation of these information distances to build approximate geometric structures; while the *existence* of a kernel for the Jensen–Shannon distance was well-known, this structure had never been exploited to give algorithms with robust theoretical guarantees.

4.2 Related work

The works of Fuglede and Topsøe (69), and later Vedaldi and Zisserman (150) study embeddings of information divergences into an infinite-dimensional Hilbert space by representing them as integrals along a one-dimensional curve in $\mathbb{C}$. Vedaldi and Zisserman give an explicit formulation of this kernel for JS and χ^2 divergences, for which a discretization (by quantizing and truncating) yields an additive error embedding into a finite-dimensional ℓ_2^2. However, they do not obtain explicit bounds on the target space dimension needed to derive a sketching algorithm; furthermore, they do not get any multiplicative approximation.

Kyng, Phillips and Venkatasubramanian (96) show a limited structure-preserving dimensionality reduction result for the Hellinger distance. Their approach works by showing that if the input points lie in a specific region of the simplex, then a standard random projection will keep the points on a lower-dimensional simplex while preserving the distances approximately. Unfortunately, this region is a small ball centered in the interior of the simplex, which further shrinks with the dimension. This is in sharp contrast to our work here, where the input points are unconstrained.

One can achieve a multiplicative approximation in the aggregate streaming model for information divergences that have a finite-dimensional embedding into ℓ_2^2. For instance, Guha, Mcgregor and Venkatasubramanian (74) observe that for the Hellinger distance that has a trivial such embedding, sketching is equivalent to sketching ℓ_2^2 and hence may be done up to a $(1+\varepsilon)$-multiplicative approximation in $\frac{1}{\varepsilon^2}\log n$ space. This immediately implies a constant factor approximation of JS and χ^2 divergences in the same space, but no bounds have been known prior to our work for a $(1+\varepsilon)$-sketching result for JS and χ^2 divergences in *any* streaming model.

There has been a wide range of work done on embedding in other spaces as well. Rahimi and Recht (134) embed shift-invariant kernels into ℓ_2^2 via random Fourier features; however their result does not hold for the more general kernels we consider in this paper. Avron, Nguyen and Woodruff (25) give a sketching technique for the polynomial kernel and pose the open question we address here of obtaining similar results for infinite-dimensional kernels. One of the most famous applications of dimension reduction is the Johnson–Lindenstrauss (JL) lemma, which states that any set of n points in ℓ_2^2 can be embedded into $O\left(\frac{\log n}{\varepsilon^2}\right)$ dimensions in the same space while preserving pairwise distances to within $(1\pm\varepsilon)$. The general literature of sketching and embeddability in normed spaces is too extensive to be reviewed here: see the work by Andoni, Krauthgamer and Razenshteyn (18) for a brief survey.

4.3 Background

In this section, we define precisely the class of information divergences that we work with, and their specific properties that allow us to obtain sketching, embedding, and dimensionality results. For what follows Δ_d denotes the *d-simplex*: $\Delta_d = \{(x_1,\ldots,x_d) \mid \sum x_i = 1$ and $x_i \geq 0, \forall i\}$. We will assume in this paper that all distributions are defined over a finite ground set $[n] = \{1,\ldots,n\}$.

Definition 8 *The* Jensen–Shannon *(JS),* Hellinger*, and χ^2 divergences between distributions p and q are defined as $JS(p,q) = \sum_i p_i \log\frac{2p_i}{p_i+q_i} + q_i \log\frac{2q_i}{p_i+q_i}$, $He(p,q) = \sum_i(\sqrt{p_i} -$*

$\sqrt{q_i})^2$ and $\chi^2(p,q) = \sum_i \frac{(p_i - q_i)^2}{p_i + q_i}$, respectively.

Definition 9 (Regular distance) *We call a distance function $D : X \to \mathbb{R}$* regular *if there exists a feature map $\phi : X \to V$, where V is a (possibly infinite-dimensional) Hilbert space, such that:*

$$D(x,y) = \|\phi(x) - \phi(y)\|^2 \quad \forall x,y \in X.$$

The works by Fuglede and Topsøe (69), and later Vedaldi and Zisserman (150) show that the JS divergence is regular: they give a feature map $\phi(x) = \int_{-\infty}^{+\infty} \Psi_x(\omega)\,\mathrm{d}\omega$, where $\Psi_x(\omega) : \mathbb{R} \to \mathbb{C}$ is given by $\Psi_x(\omega) = \exp(i\omega \ln x)\sqrt{\frac{2x\,\mathrm{sech}(\pi\omega)}{(\ln 4)(1+4\omega^2)}}$.

Hence we have for $x, y \in \mathbb{R}$, $\mathrm{JS}(x,y) = \|\phi(x) - \phi(y)\|^2 = \int_{-\infty}^{+\infty} \|\Psi_x(\omega) - \Psi_y(\omega)\|^2\,\mathrm{d}\omega$. The infinite-dimensional "embedding" for a given distribution $p \in \Delta_d$ is then the concatenation of the functions $\phi(p_i)$, i.e., $\phi(p) = (\phi_{p_1}, \ldots, \phi_{p_d})$.

4.4 Embedding JS into ℓ_2^2

We present two algorithms for embedding JS into ℓ_2^2. The first is deterministic and gives an additive error approximation whereas the second is randomized but yields a multiplicative approximation in an offline setting. The advantage of the first algorithm is that it can be realized in the streaming model, and if we make a standard assumption of polynomial precision in the streaming input, yields a $(1+\varepsilon)$-multiplicative approximation as well in this setting.

We derive some terms in the kernel representation of $\mathrm{JS}(x,y)$ which we will find convenient. First, the explicit formulation in Section 4.3 yields that for $x, y \in \mathbb{R}$:

$$\begin{aligned}
\mathrm{JS}(x,y) &= \int_{-\infty}^{+\infty} \left\| e^{i\omega \ln x}\sqrt{\frac{2x\,\mathrm{sech}(\pi\omega)}{(\ln 4)(1+4\omega^2)}} - e^{i\omega \ln y}\sqrt{\frac{2y\,\mathrm{sech}(\pi\omega)}{(\ln 4)(1+4\omega^2)}} \right\|^2 \mathrm{d}\omega \\
&= \int_{-\infty}^{+\infty} \left(\frac{2\,\mathrm{sech}(\pi\omega)}{(\ln 4)(1+4\omega^2)} \right) \|\sqrt{x}e^{i\omega \ln x} - \sqrt{y}e^{i\omega \ln y}\|^2\,\mathrm{d}\omega.
\end{aligned}$$

For convenience, we now define:

$$\begin{aligned}
h(x,y,\omega) &= \|\sqrt{x}e^{i\omega \ln x} - \sqrt{y}e^{i\omega \ln y}\|^2 \\
&= (\sqrt{x}\cos(\omega \ln x) - \sqrt{y}\cos(\omega \ln y))^2 + (\sqrt{x}\sin(\omega \ln x) - \sqrt{y}\sin(\omega \ln y))^2,
\end{aligned}$$

and

$$\kappa(\omega) = \frac{2\operatorname{sech}(\pi\omega)}{(\ln 4)(1+4\omega^2)} .$$

We can then write $\mathrm{JS}(p,q) = \sum_{i=1}^{d} f_J(p_i,q_i)$ where

$$f_J(x,y) = \int_{-\infty}^{\infty} h(x,y,\omega)\kappa(\omega)\,\mathrm{d}\omega = x\log\left(\frac{2x}{x+y}\right) + y\log\left(\frac{2y}{x+y}\right).$$

It is easy to verify that $\kappa(\omega)$ is a distribution, i.e., $\int_{-\infty}^{\infty}\kappa(\omega)d\omega = 1$.

4.4.1 Deterministic embedding

We will produce an embedding $\phi(p) = (\phi_{p_1},\ldots,\phi_{p_d})$, where each ϕ_{p_i} is an integral that we can discretize by quantizing and truncating carefully.

```
**Input:** $p = \{p_1,\ldots,p_d\}$ where coordinates are ordered by arrival.
**Output:** A vector $c^p$ of length $O\left(\frac{d^2}{\varepsilon}\log\frac{d}{\varepsilon}\right)$
$\ell \leftarrow 1$; $J \leftarrow \lceil \frac{32d}{\varepsilon}\ln\left(\frac{8d}{\varepsilon}\right)\rceil$,
for $j \leftarrow -J$ to $J$ do
    $w_j \leftarrow j \times \varepsilon/32d$
end
for $i \leftarrow 1$ to $d$ do
    for $j \leftarrow -J$ to $J-1$ do
        $a^p_\ell \leftarrow \sqrt{p_i}\cos(\omega_j \ln p_i)\sqrt{\int_{\omega_j}^{\omega_{j+1}}\kappa(\omega)d\omega}$
        $b^p_\ell \leftarrow \sqrt{p_i}\sin(\omega_j \ln p_i)\sqrt{\int_{\omega_j}^{\omega_{j+1}}\kappa(\omega)d\omega}$
        $\ell \leftarrow \ell + 1$
    end
end
return $a^p$ concatenated with $b^p$.
```

Algorithm 7: Embed $p \in \Delta_d$ under JS into ℓ_2^2.

To analyze Algorithm 7, we first obtain bounds on the function h and its derivative.

Lemma 4.4.1 *For* $0 \le x,y, \le 1$, *we have* $0 \le h(x,y,\omega) \le 2$ *and* $\left|\frac{\partial h(x,y,\omega)}{\partial\omega}\right| \le 16$.

Proof. Clearly $h(x,y,\omega) \geq 0$. Furthermore, since $0 \leq x,y \leq 1$, we have

$$h(x,y,\omega) \leq \left|\sqrt{x}e^{i\omega \ln x}\right|^2 + \left|\sqrt{y}e^{i\omega \ln y}\right|^2 = x+y \leq 2.$$

$$\begin{aligned}
&\text{Next, } \left|\frac{\partial h(x,y,\omega)}{\partial \omega}\right| \\
= \;& \left|2\left(\sqrt{x}\cos(\omega \ln x) - \sqrt{y}\cos(\omega \ln y)\right)\left(-\sqrt{x}\sin(\omega \ln x)\ln x + \sqrt{y}\sin(\omega \ln y)\ln y\right)\right. \\
& \left. +2\left(\sqrt{x}\sin(\omega \ln x) - \sqrt{y}\sin(\omega \ln y)\right)\left(\sqrt{x}\cos(\omega \ln x)\ln x - \sqrt{y}\cos(\omega \ln y)\ln y\right)\right| \\
\leq \;& \left|2\left(\sqrt{x}+\sqrt{y}\right)\left(\sqrt{x}\ln x + \sqrt{y}\ln y\right)\right| + 2\left|\left(\sqrt{x}+\sqrt{y}\right)\left(\sqrt{x}\ln x + \sqrt{y}\ln y\right)\right| \;\leq\; 16,
\end{aligned}$$

where the last inequality follows since $\max_{0\leq x\leq 1} |\sqrt{x}\ln x| < 1$. ■

The next two steps are useful to approximate the infinite-dimensional continuous representation by a finite-dimensional discrete representation by appropriately truncating and quantizing the integral.

Lemma 4.4.2 (Truncation) *For $t \geq \ln(4/\varepsilon)$,*

$$f_J(x,y) \geq \int_{-t}^{t} h(x,y,\omega)\kappa(\omega)d\omega \geq f_J(x,y) - \varepsilon\ .$$

Proof. The first inequality follows since $h(x,y,\omega) \geq 0$. For the second inequality, we use $h(x,y,\omega) \leq 2$:

$$\begin{aligned}
\int_{-\infty}^{-t} h(x,y,\omega)\kappa(\omega)d\omega + \int_{t}^{\infty} h(x,y,\omega)\kappa(\omega)d\omega &\leq 4\int_{t}^{\infty} \kappa(\omega)d\omega \\
&< 4\int_{t}^{\infty} \frac{4e^{-\pi\omega}}{\ln 4}d\omega < 4e^{-t} \leq \varepsilon,
\end{aligned}$$

where the last line follows if $t \geq \ln(4/\varepsilon)$. ■

Define $\omega_i = \varepsilon i/16$ for $i \in \{\ldots,-2,-1,0,1,2,\ldots\}$ and $\tilde{h}(x,y,\omega) = h(x,y,\omega_i)$ where $i = \max\{j \mid \omega_j \leq \omega\}$.

Lemma 4.4.3 (Quantization) *For any a,b,*

$$\int_{a}^{b} h(x,y,\omega)\kappa(\omega)d\omega = \int_{a}^{b} \tilde{h}(x,y,\omega)\kappa(\omega)d\omega \pm \varepsilon\ .$$

Proof. First note that

$$|\tilde{h}(x,y,\omega) - h(x,y,\omega)| \leq \left(\frac{\varepsilon}{16}\right) \cdot \max_{x,y\in[0,1],\omega} \left|\frac{\partial h(x,y,\omega)}{\partial \omega}\right| \leq \varepsilon \ .$$

Hence, $\left|\int_{-a}^{b} \tilde{h}(x,y,\omega)\kappa(\omega)d\omega - \int_{-a}^{b} h(x,y,\omega)\kappa(\omega)d\omega\right| \leq \left|\int_{-a}^{b} \varepsilon\kappa(\omega)d\omega\right| \leq \varepsilon.$ ■

Given a real number z, define vectors $\mathbf{v}^z$ and $\mathbf{u}^z$ indexed by $i \in \{-i^*, \ldots, -2, -1, 0, 1, 2, \ldots i^*\}$ where $i^* = \left\lceil 16\varepsilon^{-1}\ln(4/\varepsilon)\right\rceil$ by:

$$\mathbf{v}^z = \sqrt{z}\cos(\omega_i \ln z)\sqrt{\int_{\omega_i}^{\omega_{i+1}} \kappa(\omega)d\omega}, \quad \mathbf{u}^z = \sqrt{z}\sin(\omega_i \ln z)\sqrt{\int_{\omega_i}^{\omega_{i+1}} \kappa(\omega)d\omega},$$

and note that

$$(\mathbf{v}_i^x - \mathbf{v}_i^y)^2 + (\mathbf{u}_i^x - \mathbf{u}_i^y)^2 = h(x,y,\omega_i)\int_{\omega_i}^{\omega_{i+1}} \kappa(\omega)d\omega.$$

Therefore,

$$\begin{aligned}
\|\mathbf{v}^x - \mathbf{v}^y\|_2^2 + \|\mathbf{u}^x - \mathbf{u}^y\|_2^2 &= \int_{w_{-i^*}}^{w_{i^*+1}} \tilde{h}(x,y,\omega)\kappa(\omega)d\omega \\
&= \int_{w_{-i^*}}^{w_{i^*+1}} h(x,y,\omega)\kappa(\omega)d\omega \pm \varepsilon \\
&= \int_{-\infty}^{\infty} h(x,y,\omega)\kappa(\omega)d\omega \pm 2\varepsilon \;=\; f_J(x,y) \pm 2\varepsilon,
\end{aligned}$$

where the second to last line follows from lemma 4.4.3 and the last line follows from lemma 4.4.2, since $\min(|w_{-i^*}|, w_{i^*+1}) \geq \ln(4/\varepsilon)$.

Define the vector $\mathbf{a}^p$ to be the vector generated by concatenating $\mathbf{v}^{p_i}$ and $\mathbf{u}^{p_i}$ for $i \in [d]$. Then it follows that

$$\|\mathbf{a}^p - \mathbf{a}^q\|_2^2 = \mathrm{JS}(p,q) \pm 2\varepsilon d.$$

Hence we have reduced the problem of estimating $\mathrm{JS}(p,q)$ to ℓ_2 estimation. Rescaling $\varepsilon \leftarrow \varepsilon/(2d)$ ensures the additive error is ε while the length of the vectors $\mathbf{a}^p$ and $\mathbf{a}^q$ is $O\left(\frac{d^2}{\varepsilon}\log\frac{d}{\varepsilon}\right)$.

Theorem 4.4.1 *Algorithm 7 embeds a set P of points under JS into* $O\left(\frac{d^2}{\varepsilon}\log\frac{d}{\varepsilon}\right)$ *dimensions under* ℓ_2^2 *with* ε *additive error, independent of the size of* $|P|$.

Note that using the JL-lemma, the dimensionality of the target space can be reduced to $O\left(\frac{\log|P|}{\varepsilon^2}\right)$. Theorem 4.4.1, along with the AMS sketch of (11), and the standard assumption of polynomial precision immediately implies:

Corollary 4.4.1 *There is an algorithm that works in the aggregate streaming model to approximate JS to within $(1+\varepsilon)$-multiplicative factor using $O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon}\log d\right)$ space.*

As noted earlier, this is the first algorithm in the aggregate streaming model to obtain an $(1+\varepsilon)$-multiplicative approximation to JS, which contrasts against linear space lower bounds for the same problem in the update streaming model.

4.4.2 Randomized embedding

In this section we show how to embed n points of JS into $\ell_2^{\bar{d}}$ with $(1+\varepsilon)$ distortion where $\bar{d} = O(n^2 d^3 \varepsilon^{-2})$. [1] This can be reduced further to a dimension $O(\log n/\varepsilon^2)$ by simply applying the Eucllidean JL-lemma.

In spirit, our approach is along the lines of Rahimi and Recht (134) in that we sample from the kernel repeatedly to obtain each coordinate of the embedding, so that the final ℓ_2^2 distance is an unbiased estimate of the divergence. Showing this estimate is sufficiently concentrated is the main technical contribution of this section.

For fixed $x, y, \in [0,1]$, we first consider the random variable T where T takes the value $h(x,y,\omega)$ with probability $\kappa(\omega)$. (Recall that $\kappa(\cdot)$ is a distribution.) We compute the first and second moments of T.

Theorem 4.4.2 $E[T] = f_J(x,y)$ *and* $\mathrm{var}[T] \leq 36(f_J(x,y))^2$.

Proof. The expectation follows immediately from the definition:

$$E[T] = \int_{-\infty}^{\infty} h(x,y,\omega)\kappa(\omega)d\omega = f_J(x,y).$$

To bound the variance, it will be useful to define the function $f_H(x,y) = (\sqrt{x}-\sqrt{y})^2$ corresponding to the one-dimensional Hellinger distance that is related to $f_J(x,y)$ as follows. We now state two claims regarding $f_H(x,y)$ and $f_\chi(x,y)$:

Claim 4.4.1 *For all* $x, y \in [0,1]$, $f_H(x,y) \leq 2 f_J(x,y)$.

Proof. Let $f_\chi(x,y) = \frac{(x-y)^2}{x+y}$ correspond to the one-dimensional χ^2 distance. Then, we have

[1] If we ignore precision constraints on sampling from a continuous distribution in a streaming algorithm, then this also would yield a sketching bound of $O(d^3\varepsilon^{-2})$ for a $(1+\varepsilon)$ multiplicative approximation.

$$\frac{f_\chi(x,y)}{f_H(x,y)} = \frac{(x-y)^2}{(x+y)(\sqrt{x}-\sqrt{y})^2} = \frac{(\sqrt{x}+\sqrt{y})^2}{x+y} = \frac{x+y+2\sqrt{xy}}{x+y} \geq 1 \ .$$

This shows that $f_H(x,y) \leq f_\chi(x,y)$. To show $f_\chi(x,y) \leq 2f_J(x,y)$ we refer the reader to (148, Section 3). Combining these two relationships gives us our claim. ■

We then bound $h(x,y,\omega)$ in terms of $f_H(x,y)$ as follows.

Claim 4.4.2 *For all* $x,y \in [0,1], \omega \in \mathbb{R}$, $h(x,y,\omega) \leq f_H(x,y)(1+2|\omega|)^2$.

Proof. Without loss of generality, assume $x \geq y$.

$$\begin{aligned}
\sqrt{h(x,y,\omega)} &= |\sqrt{x}\cdot e^{i\omega \ln x} - \sqrt{y}\cdot e^{i\omega \ln y}| \\
&\leq |\sqrt{x}\cdot e^{i\omega \ln x} - \sqrt{y}\cdot e^{i\omega \ln x}| + |\sqrt{y}\cdot e^{i\omega \ln x} - \sqrt{y}\cdot e^{i\omega \ln y}| \\
&= |\sqrt{x}-\sqrt{y}| + \sqrt{y}\cdot|e^{i\omega \ln x} - e^{i\omega \ln y}| \\
&= |\sqrt{x}-\sqrt{y}| + \sqrt{y}\cdot 2\cdot|\sin(\omega \ln(x/y)/2)| \\
&\leq \sqrt{f_H(x,y)} + \sqrt{y}\cdot 2\cdot|\omega \ln(\sqrt{x/y})| \\
&\leq \sqrt{f_H(x,y)} + \sqrt{y}\cdot 2\cdot|\sqrt{x/y}-1|\cdot|\omega| \\
&= \sqrt{f_H(x,y)} + 2\sqrt{f_H(x,y)}\cdot|\omega|
\end{aligned}$$

and hence $h(x,y,\omega) \leq f_H(x,y)(1+2|\omega|)^2$ as required. ■

These claims allow us to bound the variance:

$$\begin{aligned}
\mathrm{var}[T] \leq E[T^2] = \int_{-\infty}^{\infty} (h(x,y,\omega))^2 \kappa(\omega)d\omega &\leq f_H(x,y)^2 \int_{-\infty}^{\infty} (1+2|\omega|)^4 \kappa(\omega)d\omega \\
&= f_H(x,y)^2 \cdot 8.94 \ < \ 36 f_J(x,y)^2,
\end{aligned}$$

■

This naturally gives rise to the following Algorithm 8. Let $\omega_1,\ldots,\omega_t$ be t independent samples chosen according to $\kappa(\omega)$. For any distribution p on $[d]$, define vectors $\mathbf{v}^p, \mathbf{u}^p \in \mathbb{R}^{td}$ where, for $i \in [d], j \in [t]$,

$$\mathbf{v}^p_{i,j} = \sqrt{p_i}\cdot\cos(\omega_j \ln p_i)/t, \quad \mathbf{u}^p_{i,j} = \sqrt{p_i}\cdot\sin(\omega_j \ln p_i)/t.$$

Let $\mathbf{v}^p_i$ be a concatenation of $\mathbf{v}^p_{i,j}$ and $\mathbf{u}^p_{i,j}$ over all $j \in [t]$. Then note that $E[\|\mathbf{v}^p_i - \mathbf{v}^q_i\|_2^2] = f_J(p_i,q_i)$ and $\mathrm{var}[\|\mathbf{v}^p_i - \mathbf{v}^q_i\|_2^2] \leq 36(f_J(p_i,q_i))^2/t$. Hence, for $t = 36n^2d^2\varepsilon^{-2}$, by an application of the Chebyshev bound,

$$\Pr[|\|\mathbf{v}^p_i - \mathbf{v}^q_i\|_2^2 - f_J(p_i,q_i)| \geq \varepsilon f_J(x,y)] \leq 36\varepsilon^{-2}/t = (nd)^{-2}. \tag{4.1}$$

```
**Input:** $p = \{p_1, \ldots, p_d\}$.
**Output:** A vector $c^p$ of length $O\left(n^2 d^3 \varepsilon^{-2}\right)$
$\ell \leftarrow 1$; $s \leftarrow \lceil 36 n^2 d^2 \varepsilon^{-2} \rceil$

for $j \leftarrow 1$ to $s$ do
    $\omega_j \leftarrow$ a draw from $\kappa(\omega)$;
end
for $i \leftarrow 1$ to $d$ do
    for $j \leftarrow 1$ to $s$ do
        $a^p_\ell \leftarrow \left(\sqrt{p_i}\cos(\omega_j \ln p_i)/\sqrt{s}\right)$
        $b^p_\ell \leftarrow \left(\sqrt{p_i}\sin(\omega_j \ln p_i)/\sqrt{s}\right)$
        $\ell \leftarrow \ell + 1$
    end
end
return $a^p$ concatenated with $b^p$.
```

Algorithm 8: Embeds point $p \in \Delta_d$ under JS into ℓ_2^2.

By an application of the union bound over all pairs of points:

$$\Pr[\exists i \in [d]\ ,\ p,q \in P| \|\mathbf{v}_i^p - \mathbf{v}_i^q\|_2^2 - f_J(p_i,q_i)| \geq \varepsilon f_J(p_i,q_i)] \leq 1/d.$$

And hence, if $\mathbf{v}^p$ is a concatenation of $\mathbf{v}_i^p$ over all $i \in [d]$, then with probability at least $1 - 1/d$ it holds for all $p,q \in P$:

$$(1-\varepsilon)\mathrm{JS}(p,q) \leq \|\mathbf{v}^p - \mathbf{v}^q\| \leq (1+\varepsilon)\mathrm{JS}(p,q).$$

The final length of the vectors is then $td = 36n^2d^3\varepsilon^{-2}$ for approximately preserving distances between every pair of points with probability at least $1 - \frac{1}{d}$. This can be reduced further to $O(\log n/\varepsilon^2)$ by simply applying the JL-lemma.

4.5 Embedding χ^2 into ℓ_2^2

We give here two algorithms for embedding the χ^2 divergence into ℓ_2^2. The computation and resulting two algorithms are highly analogous to Section 4.4. First, the explicit formulation given by (150) yields that for $x, y \in \mathbb{R}$:

$$\begin{aligned}\chi^2(x,y) &= \int_{-\infty}^{+\infty} \left\| e^{i\omega \ln x}\sqrt{x\,\mathrm{sech}(\pi\omega)} - e^{i\omega \ln y}\sqrt{y\,\mathrm{sech}(\pi\omega)} \right\|^2 \mathrm{d}\omega \\ &= \int_{-\infty}^{+\infty} (\mathrm{sech}(\pi\omega))\, \|\sqrt{x}e^{i\omega \ln x} - \sqrt{y}e^{i\omega \ln y}\|^2\, \mathrm{d}\omega.\end{aligned}$$

For convenience, we now define:

$$h(x,y,\omega) = \|\sqrt{x}e^{i\omega \ln x} - \sqrt{y}e^{i\omega \ln y}\|^2$$

and $\kappa_\chi(\omega) = \mathrm{sech}(\pi\omega)$.

We can then write $\chi^2(p,q) = \sum_{i=1}^d f_\chi(p_i,q_i)$ where

$$f_\chi(x,y) = \int_{-\infty}^{\infty} h(x,y,\omega)\kappa_\chi(\omega)\,\mathrm{d}\omega = \frac{(x-y)^2}{x+y}.$$

It is easy to verify that $\kappa_\chi(\omega)$ is a distribution, i.e., $\int_{-\infty}^{\infty} \kappa_\chi(\omega) d\omega = 1$.

4.5.1 Deterministic embedding

We will produce an embedding $\phi(p) = (\phi_{p_1}, \ldots, \phi_{p_d})$, where each ϕ_{p_i} is an integral that we discretize appropriately.

Lemma 4.5.1 *For* $0 \leq x,y, \leq 1$, *we have* $0 \leq h(x,y,\omega) \leq 2$ *and* $\left|\frac{\partial h(x,y,\omega)}{\partial \omega}\right| \leq 16.$

```
Input: $p = \{p_1, \dots, p_d\}$ where coordinates are ordered by arrival.
Output: A vector $c^p$ of length $O\left(\frac{d^2}{\varepsilon}\log\frac{d}{\varepsilon}\right)$
$\ell \leftarrow 1$; $J \leftarrow \lceil \frac{32d}{\varepsilon}\ln\left(\frac{6d}{\varepsilon}\right)\rceil$,

for $j \leftarrow -J$ to $J$ do
    $w_j \leftarrow j \times \varepsilon/32d$
end
for $i \leftarrow 1$ to $d$ do
    for $j \leftarrow -J$ to $J-1$ do
        $a^p_\ell \leftarrow \sqrt{p_i}\cos(\omega_j \ln p_i)\sqrt{\int_{\omega_j}^{\omega_{j+1}} \kappa_\chi(\omega)d\omega}$
        $b^p_\ell \leftarrow \sqrt{p_i}\sin(\omega_j \ln p_i)\sqrt{\int_{\omega_j}^{\omega_{j+1}} \kappa_\chi(\omega)d\omega}$
        $\ell \leftarrow \ell + 1$
    end
end
return $a^p$ concatenated with $b^p$.
```

Algorithm 9: Embed $p \in \Delta_d$ under χ^2 into ℓ_2^2.

Similar to Section 4.4, the next two steps analyze truncating and quantizing the integral.

Lemma 4.5.2 (Truncation) *For $t \geq \ln(3/\varepsilon)$,*

$$f_\chi(x,y) \geq \int_{-t}^{t} h(x,y,\omega)\kappa_\chi(\omega)d\omega \geq f_\chi(x,y) - \varepsilon \,.$$

Proof. The first inequality follows since $h(x,y,\omega) \geq 0$. For the second inequality, we use $h(x,y,\omega) \leq 2$:

$$\begin{aligned}\int_{-\infty}^{-t} h(x,y,\omega)\kappa_\chi(\omega)d\omega + \int_{t}^{\infty} h(x,y,\omega)\kappa_\chi(\omega)d\omega &\leq 4\int_{t}^{\infty}\kappa_\chi(\omega)d\omega \\ &< 4\int_{t}^{\infty} 2e^{-\pi\omega}d\omega < 3e^{-t} \leq \varepsilon,\end{aligned}$$

where the last line follows if $t \geq \ln(3/\varepsilon)$. ■

Define $\omega_i = \varepsilon i/16$ for $i \in \{\ldots,-2,-1,0,1,2,\ldots\}$ and $\tilde{h}(x,y,\omega) = h(x,y,\omega_i)$ where $i = \max\{j \mid \omega_j \leq \omega\}$. We recall the following lemma from Section 4.4:

Lemma 4.5.3 (Quantization) *For any a, b,*

$$\int_a^b h(x,y,\omega)\kappa_\chi(\omega)d\omega = \int_a^b \tilde{h}(x,y,\omega)\kappa_\chi(\omega)d\omega \pm \varepsilon \,.$$

Given a real number z, define vectors $\mathbf{v}^z$ and $\mathbf{u}^z$ indexed by $i \in \{-i^*,\ldots,-2,-1,0,1,2,\ldots i^*\}$ where $i^* = \lceil 16\varepsilon^{-1}\ln(3/\varepsilon)\rceil$ by:

$$\mathbf{v}^z = \sqrt{z}\cos(\omega_i \ln z)\sqrt{\int_{\omega_i}^{\omega_{i+1}} \kappa_\chi(\omega)d\omega}, \quad \mathbf{u}^z = \sqrt{z}\sin(\omega_i \ln z)\sqrt{\int_{\omega_i}^{\omega_{i+1}} \kappa_\chi(\omega)d\omega},$$

and note that

$$(\mathbf{v}_i^x - \mathbf{v}_i^y)^2 + (\mathbf{u}_i^x - \mathbf{u}_i^y)^2 = h(x,y,\omega_i)\int_{\omega_i}^{\omega_{i+1}} \kappa_\chi(\omega)d\omega.$$

Therefore,

$$\begin{aligned}\|\mathbf{v}^x - \mathbf{v}^y\|_2^2 + \|\mathbf{u}^x - \mathbf{u}^y\|_2^2 &= \int_{w_{-i^*}}^{w_{i^*+1}} \tilde{h}(x,y,\omega)\kappa_\chi(\omega)d\omega \\ &= \int_{w_{-i^*}}^{w_{i^*+1}} h(x,y,\omega)\kappa_\chi(\omega)d\omega \pm \varepsilon \\ &= \int_{-\infty}^{\infty} h(x,y,\omega)\kappa_\chi(\omega)d\omega \pm 2\varepsilon \;=\; f_\chi(x,y) \pm 2\varepsilon,\end{aligned}$$

where the second to last line follows from lemma 4.5.3 and the last line follows from lemma 4.5.2, since $\min(|w_{-i^*}|, w_{i^*+1}) \geq \ln(3/\varepsilon)$.

Define the vector $\mathbf{a}^p$ to be the vector generated by concatenating $\mathbf{v}^{p_i}$ and $\mathbf{u}^{p_i}$ for $i \in [d]$. Then it follows that

$$\|\mathbf{a}^p - \mathbf{a}^q\|_2^2 = \chi^2(p,q) \pm 2\varepsilon d.$$

Hence we have reduced the problem of estimating $\chi^2(p,q)$ to ℓ_2 estimation. Rescaling $\varepsilon \leftarrow \varepsilon/(2d)$ ensures the additive error is ε while the length of the vectors $\mathbf{a}^p$ and $\mathbf{a}^q$ is $O\left(\frac{d^2}{\varepsilon}\log\frac{d}{\varepsilon}\right)$.

Theorem 4.5.1 *Algorithm 9 embeds a set P of points under χ^2 into $O\left(\frac{d^2}{\varepsilon}\log\frac{d}{\varepsilon}\right)$ dimensions under ℓ_2^2 with ε additive error, independent of the size of $|P|$.*

Theorem 4.5.1, along with the AMS sketch of (11), and the standard assumption of polynomial precision immediately implies:

Corollary 4.5.1 *There is an algorithm that works in the aggregate streaming model to approximate χ^2 to within $(1+\varepsilon)$-multiplicative factor using $O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon}\log d\right)$ space.*

4.5.2 Randomized embedding

In this section we show how to embed n points of χ^2 into $\ell_2^{\bar{d}}$ with $(1+\varepsilon)$ distortion where $\bar{d} = O(n^2 d^3 \varepsilon^{-2})$. [2]

For fixed $x, y, \in [0,1]$, we first consider the random variable T where T takes the value $h(x,y,\omega)$ with probability $\kappa_\chi(\omega)$. (Recall that $\kappa_\chi(\cdot)$ is a distribution.) We compute the first and second moments of T.

Theorem 4.5.2 *$E[T] = f_\chi(x,y)$ and* $\text{var}[T] \leq 23(f_\chi(x,y))^2$.

Proof. The expectation follows immediately from the definition:

$$E[T] = \int_{-\infty}^{\infty} h(x,y,\omega)\kappa_\chi(\omega)d\omega = f_\chi(x,y).$$

To bound the variance we will again use the function $f_H(x,y) = (\sqrt{x}-\sqrt{y})^2$ corresponding to the one-dimensional Hellinger distance. We now state two claims relating $f_H(x,y)$ and $f_\chi(x,y)$:

[2] If we ignore precision constraints on sampling from a continuous distribution in a streaming algorithm, then this also would yield a sketching bound of $O(d^3\varepsilon^{-2})$ for a $(1+\varepsilon)$ multiplicative approximation.

Claim 4.5.1 *For all $x, y \in [0,1]$, $f_H(x,y) \leq f_\chi(x,y)$.*

Proof. Let $f_\chi(x,y) = \frac{(x-y)^2}{x+y}$ correspond to the one-dimensional χ^2 distance. Then, we have

$$\frac{f_\chi(x,y)}{f_H(x,y)} = \frac{(x-y)^2}{(x+y)(\sqrt{x}-\sqrt{y})^2} = \frac{(\sqrt{x}+\sqrt{y})^2}{x+y} = \frac{x+y+2\sqrt{xy}}{x+y} \geq 1 \ .$$

This shows that $f_H(x,y) \leq f_\chi(x,y)$. ■

We then recall Claim 4.4.2 bounding $h(x,y,\omega)$ in terms of $f_H(x,y)$ as follows.

Claim 4.5.2 *For all $x, y \in [0,1], \omega \in \mathbb{R}$, $h(x,y,\omega) \leq f_H(x,y)(1+2|\omega|)^2$.*

These claims allow us to bound the variance:

$$\begin{aligned} \mathrm{var}[T] \leq E[T^2] = \int_{-\infty}^{\infty} (h(x,y,\omega))^2 \kappa_\chi(\omega) d\omega &\leq f_H(x,y)^2 \int_{-\infty}^{\infty} (1+2|\omega|)^4 \kappa_\chi(\omega) d\omega \\ &= f_H(x,y)^2 \cdot 22.77 < 23 f_\chi(x,y)^2, \end{aligned}$$

■

This naturally gives rise to the following Algorithm 10.

```
Input: $p = \{p_1, \ldots, p_d\}$.
Output: A vector $a^p$ of length $O\left(n^2 d^3 \varepsilon^{-2}\right)$
$\ell \leftarrow 1$; $s \leftarrow \lceil 23 n^2 d^2 \varepsilon^{-2} \rceil$

for $j \leftarrow 1$ to $s$ do
    $\omega_j \leftarrow$ a draw from $\kappa_\chi(\omega)$;
end
for $i \leftarrow 1$ to $d$ do
    for $j \leftarrow 1$ to $s$ do
        $a^p_\ell \leftarrow \left(\sqrt{p_i}\cos(\omega_j \ln p_i)/\sqrt{s}\right)$
        $b^p_\ell \leftarrow \left(\sqrt{p_i}\sin(\omega_j \ln p_i)/\sqrt{s}\right)$
        $\ell \leftarrow \ell + 1$
    end
end
return $a^p$ concatenated with $b^p$.
```

Algorithm 10: Embeds point $p \in \Delta_d$ under χ^2 into ℓ_2^2.

Let $\omega_1, \ldots, \omega_t$ be t independent samples chosen according to $\kappa_\chi(\omega)$. For any distribution p on $[d]$, define vectors $\mathbf{v}^p, \mathbf{u}^p \in \mathbb{R}^{td}$ where, for $i \in [d], j \in [t]$,

$$\mathbf{v}^p_{i,j} = \sqrt{p_i} \cdot \cos(\omega_j \ln p_i)/t, \quad \mathbf{u}^p_{i,j} = \sqrt{p_i} \cdot \sin(\omega_j \ln p_i)/t.$$

Let $\mathbf{v}^p_i$ be a concatenation of $\mathbf{v}^p_{i,j}$ and $\mathbf{u}^p_{i,j}$ over all $j \in [t]$. Then note that $E[\|\mathbf{v}^p_i - \mathbf{v}^q_i\|_2^2] = f_\chi(p_i, q_i)$ and $\text{var}[\|\mathbf{v}^p_i - \mathbf{v}^q_i\|_2^2] \leq 23(f_\chi(p_i, q_i))^2/t$. Hence, for $t = 23n^2d^2\varepsilon^{-2}$, by an application of the Chebyshev bound,

$$\Pr[|\|\mathbf{v}^p_i - \mathbf{v}^q_i\|_2^2 - f_\chi(p_i, q_i)| \geq \varepsilon f_\chi(x,y)] \leq 23\varepsilon^{-2}/t = (nd)^{-2}. \tag{4.2}$$

By an application of the union bound over all pairs of points:

$$\Pr[\exists i \in [d]\ ,\ p,q \in P | \|\mathbf{v}^p_i - \mathbf{v}^q_i\|_2^2 - f_\chi(p_i, q_i)| \geq \varepsilon f_\chi(p_i, q_i)] \leq 1/d.$$

And hence, if $\mathbf{v}^p$ is a concatenation of $\mathbf{v}^p_i$ over all $i \in [d]$, then with probability at least $1 - 1/d$,

$$(1-\varepsilon)\chi^2(p,q) \leq \|\mathbf{v}^p - \mathbf{v}^q\| \leq (1+\varepsilon)\chi^2(p,q).$$

The final length of the vectors is then $td = 23n^2d^3\varepsilon^{-2}$ for approximately preserving distances between every pair of points with probability at least $1 - \frac{1}{d}$. This can be reduced further to $O(\log n/\varepsilon^2)$ by simply applying the JL-lemma.

4.6 Dimensionality reduction

The JL-lemma has been instrumental for improving the speed and approximation ratios of learning algorithms. In this section, we give a proof of the JL-analogue for a general class of divergences that includes the information divergences studied here. Specifically, we show that a set of n points lying on a high-dimensional simplex can be embedded to a $k = O(\log n/\varepsilon^2)$-dimensional simplex, while approximately preserving the information distances between all pairs of points. This dimension reduction amounts to reducing the support of the distribution from d to k, while approximately maintaining the divergences.

Our proof uses ℓ_2^2 as an intermediate space. On a high level, we first embed the points into a high (but finite) - dimensional ℓ_2^2 space, using the techniques we developed in Section 4.4.2. We then use the Euclidean JL-lemma to reduce the dimensionality, and remap the points into the interior of a simplex. Finally, we show that far away from the simplex boundaries, this class of divergences has the same structure as ℓ_2^2, hence the embedding back into information spaces can be done with a simple translation and rescaling. Note that for divergences that have an embedding into finite-dimensional ℓ_2^2, the proof is constructive.

Definition 10 (f-divergence) *Let p and q be two distributions on $[n]$. A convex function $f : [0,\infty) \to \mathbb{R}$ such that $f(1) = 0$ gives rise to an f-*divergence *$D_f : \Delta_d \to \mathbb{R}$ as: $D_f(p,q) = \sum_{i=1}^{d} p_i \cdot f\left(\frac{q_i}{p_i}\right)$, where we define $0 \cdot f(0/0) = 0$, $a \cdot f(0/a) = a \cdot \lim_{u\to 0} f(u)$, and $0 \cdot f(a/0) = a \cdot \lim_{u\to\infty} f(u)/u$.*

Definition 11 (Well-behaved divergence) *A* well-behaved *f-divergence is a regular f-divergence such that $f(1) = 0$, $f'(1) = 0$, $f''(1) > 0$, and $f'''(1)$ exists.*

```
Input: Set $P = \{p_1, \ldots, p_n\}$ of points on $\Delta_d$, error parameter $\varepsilon$, constant $c_0(\varepsilon, f)$
Output: A set $\bar{P}$ of points on $\Delta_k$ where $k = O\left(\frac{\log n}{\varepsilon^2}\right)$
  1. Embed $P$ into $\ell_2^2$ to obtain $P_1$ with error parameter $\varepsilon/4$.
  2. Apply Euclidean JL–lemma with error $\frac{\varepsilon}{4}$ to obtain $P_2$ in dimension $k = O\left(\frac{\log n}{\varepsilon^2}\right)$
  3. Remap $P_2$ to the plane $L = \{x \in \mathbb{R}^{k+1} \mid \sum_i x_i = 0\}$ to obtain $P_3$
  4. Scale $P_3$ to a ball of radius $c_0 \cdot \frac{\varepsilon}{k+1}$ and center at the centroid of $\Delta_{k+1}$ to obtain $\bar{P}$.
```

Algorithm 11: Dimension reduction for D_f

To analyze the above algorithm, we recall the JL–lemma (33, 88):

Lemma 4.6.1 *For any set of points P in a (possibly infinite-dimensional) Hilbert space H, there exists a randomized map $f\colon H \to \mathbb{R}^k$, $k = O(\frac{\log n}{\varepsilon^2})$ such that whp, $\forall p, q \in P$,*

$$(1-\varepsilon)\|p-q\|_2^2 \le \|f(p) - f(q)\|_2^2 \le (1+\varepsilon)\|p-q\|_2^2.$$

Corollary 4.6.1 *For any set of points P in H there exists a constant t and a randomized map $f\colon H \to \Delta_{k+1}$, $k = O(\frac{\log n}{\varepsilon^2})$ such that $\forall p, q \in P$:*

$$(1-\varepsilon)\|p-q\|_2^2 \le t\|f(p) - f(q)\|_2^2 \le (1+\varepsilon)\|p-q\|_2^2.$$

Furthermore for any small enough constant r, we may bound the domain of f to be a ball B of radius r centered at the simplex centroid, $(1/k{+}1, 1/k{+}1, \ldots, 1/k{+}1)$.

We now show that any well-behaved f divergence is nearly Euclidean near the simplex centroid.

Lemma 4.6.2 *Consider any well-behaved f divergence D_f, and let B_r be a ball of radius r such that $B_r \subset \Delta_k$ and B_r is centered at the simplex centroid. Then for any fixed $0 < \varepsilon < 1$, there exists a choice of r and scaling factor t (both dependent on k) such that $\forall p, q \in B$:*

$$(1-\varepsilon)\|p-q\|_2^2 \leq tD_f(p,q) \leq (1+\varepsilon)\|p-q\|_2^2.$$

We note that the required value of r can be computed easily for the Hellinger and χ^2 divergence, and that r behaves as $\frac{1}{k} \cdot c$ where $c = c(f, \varepsilon)$ is a sufficiently small constant depending only on ε and the function f and not on k or n . To conclude the proof note that the overall distortion is bounded by the combination of errors due to the initial embedding into P_1, the application of JL-lemma, and the final reinterpretation of the points in Δ_{k+1}. The overall error is thus bounded by, $(1+\varepsilon/4)^3 \leq 1+\varepsilon$.

Theorem 4.6.1 *Consider a set $P \in \Delta_d$ of n points under a well-behaved f-divergence D_f. Then there exists a $(1+\varepsilon)$ distortion embedding of P into Δ_k under D_f for some choice of k bounded as $O\left(\frac{\log n}{\varepsilon^2}\right)$. Furthermore this embedding can be explicitly computed even for a well-behaved f-divergence with an infinite-dimensional kernel, if the kernel can be approximated in finite dimensions within a multiplicative error as we show for JS and χ^2.*

4.7 Experiments

We analyze the empirical performance of our algorithms and demonstrate the effect that each of the parameters has on the quality of the final solution. We show that there is very little loss incurred both in sampling from the kernel (embedding the points into ℓ_2^2), and in remapping the points to lie on the d-dimensional simplex, for a set of parameters far smaller than those guaranteed by the analysis.

Recall that the dimension reduction procedure in Algorithm 11 has three parameters: s, the number of samples used to embed the points into ℓ_2^2; k, the target dimension of the Euclidean JL-lemma, and c_0, the scaling parameter used to embed the points in the final simplex.

Synthetic data. To study the quality of the embedding with respect to these three parameters, we generated distributions on $d = 100, 1000$, and $10,000$ dimensions. We used the Cauchy distribution, and the Log-Factorial[3] distribution as the seeds. To generate a point in the dataset, we randomly permuted the coordinates in one of these distributions.

To explore the dependence on the parameters, we set the defaults to $s = 100$, $k = 300$, and $c_0 = 0.1$. Note that the value of s is *far lower* than that implied by the analysis (the value of c_0 is far higher). In the three panels of Figure 4.1 we vary one of the parameters while keeping the others fixed; all of these are averaged over 100 pairwise computations. We track the error introduced by embedding into ℓ_2^2, reducing the dimension to d, and re-embedding back into the simplex Δ_d. As expected, we show that the overall error decreases with increasing the number of samples, and with lowering c_0. These contributions are on the order of 0.075% to 0.3%, and far outweighed by the error introduced by the JL-lemma itself, which is on the order of 9-10% and forms the core of the reduction.

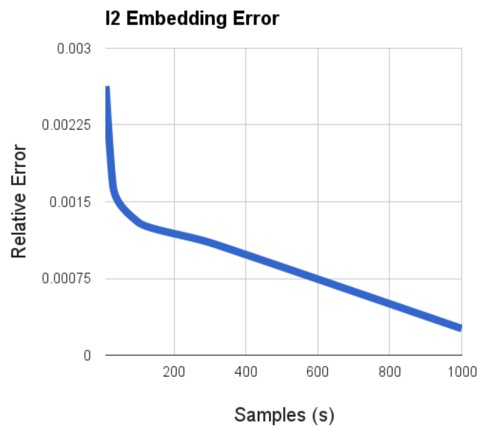
The only known method for dimension reduction in the simplex is due to (96), which essentially eschews the kernel embedding into ℓ_2^2 and proceeds to apply the JL-lemma directly on the distribution points. While it provably works only in a limited domain, we nevertheless investigate its performance. While the error of our method on the synthetic dataset ranges from 5-30% depending on the value of the target dimension, k, the error produced by the baseline method ranges from 95-430%, that is the baseline approximates the distances by a *factor* of $2-4$, instead of 5-30%.

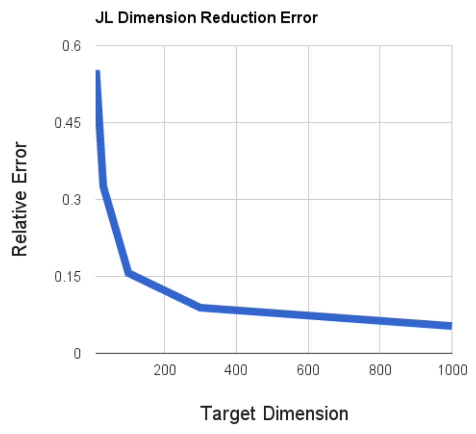
To further demonstrate the efficacy of our approach we use the word distributions in different book genres as gathered from the free sample on http://www.wordfrequency.info. We use Algorithm 11 to reduce the dimensionality twenty-fold from 6000 to just 300 dimensions. Using the same fixed set of parameters, we show the average (over 10 runs) error between the different genres in the Table 4.1.

Table 4.1. Relative error of the JS divergence after embedding into 300 dimensions

	Spoken	Fiction	Popular	Newspaper	Academic
Spoken	-	1.5%	3.3%	1.87%	0.3%
Fiction	-	-	1.4%	3.75%	2.95%
Popular	-	-	-	5.2%	5.15%
Newspaper	-	-	-	-	1.25%

[3] The distribution has values $p_1, \ldots, p_d$, where $p_i \propto \log(1+i) = \frac{\log(1+i)}{\log(d+1)!}$.





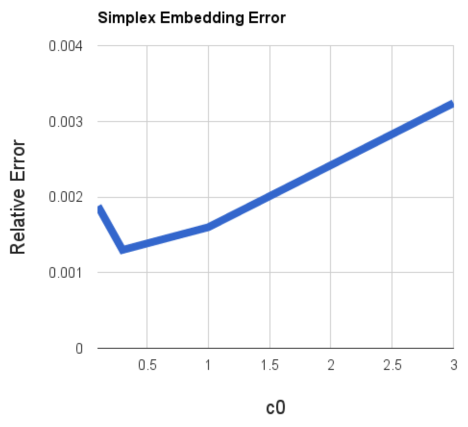


Figure 4.1. The error due to the number of samples, the JL-lemma, and the simplex embedding.

CHAPTER 5

SPECTRAL ALGORITHMS FOR NEAREST NEIGHBOR SEARCH

We study spectral algorithms for the high-dimensional Nearest Neighbor Search problem (NNS). In particular, we consider a semirandom setting, where a dataset P in $\mathbb{R}^d$ is chosen arbitrarily from an unknown subspace of low dimension $k \ll d$, and then perturbed by fully d-dimensional Gaussian noise. We design spectral NNS algorithms whose query time depends polynomially on d and $\log n$ (where $n = |P|$) for large ranges of k, d and n. Our algorithms use a repeated computation of the top PCA vector/subspace, and are effective even when the random-noise magnitude is *much larger* than the interpoint distances in P. Our motivation is that, in practice, a number of NNS algorithms use spectral methods, which still lack a good theoretical justification, often outperforming the random-projection methods that seem otherwise to be theoretically optimal.

5.1 Background and motivation

A fundamental tool in high-dimensional computational geometry is the random projection method. Most notably, the Johnson-Lindenstrass lemma (88) says that projecting onto a uniformly random k-dimensional subspace of $\mathbb{R}^d$ approximately preserves the distance between any (fixed) points $x, y \in \mathbb{R}^d$ (up to scaling), except with probability exponentially small in k. This turns out to be a very powerful approach as it effectively reduces the dimension from d to a usually much smaller k via a computationally cheap projection, and as such has had a tremendous impact on algorithmic questions in high-dimensional geometry.

A classical application of random projections is to the high-dimensional Nearest Neighbor Search (NNS) problem. Here, we are given a dataset of n points from $\mathbb{R}^d$, which we preprocess to subsequently find, given a query point $q \in \mathbb{R}^d$, its closest point from the dataset. It is now well-known that exact NNS admits algorithms whose running times have good dependence on n, but exponential dependence on the dimension d (46, 110); however these are unsatisfactory for moderately large d.

To deal with this "curse of dimensionality," researchers have studied algorithms for *approximate* NNS, and indeed in the high-dimensional regime, many, if not all, of these algorithms rely heavily on the random projection method. Consider the case of Locality-Sensitive Hashing (LSH), introduced in (84), which has been a theoretically and practically successful approach to NNS. All known variants of LSH for the Euclidean space, including (16, 58, 84), involve random projections.[1] For example, the space partitioning algorithm of (58) can be viewed as follows. Project the dataset onto a random k-dimensional subspace, and impose a randomly-shifted uniform grid. Then, to locate the near(est) neighbor of a point q, look up the points in the grid cell q falls into. Usually, this space partitioning is repeated a few times to amplify the probability of success (see also (127)).

While random projections work well and have provable guarantees, it is natural to ask whether one can improve the performance by replacing "random" projections with "best" projections. Can one optimize the projection to use — and the space partitioning more generally — as a *function of the dataset at hand*? For example, in some tasks requiring dimensionality reduction, practitioners often rely on Principal Component Analysis (PCA) and its variants. Indeed, in practice, this consideration led to numerous successful heuristics such as PCA tree (109, 143, 152) and its variants (called randomized kd-tree) (113, 140), spectral hashing (155), semantic hashing (138), and WTA hashing (157), to name just a few. Oftentimes, these heuristics outperform algorithms based on vanilla random dimension reductions. All of them adapt to the dataset, including many that perform some *spectral* decomposition of the dataset. However, in contrast to the random projection method, none of these methods have rigorous correctness or performance guarantees.

Bridging the gap between random projections and data-aware projections has been recognized as a big open question in Massive Data Analysis, see, e.g., a recent National

[1]While (84) is designed for the Hamming space, their algorithm is extended to the Euclidean space by an embedding of ℓ_2 into ℓ_1, which itself uses random projections (87).

Research Council report (123, Section 5). The challenge here is that random projections are themselves (theoretically) optimal not only for dimensionality reduction (10), but also for some of its algorithmic applications (86, 156), including NNS in certain settings (17). We are aware of only one line of work addressing this question: data-dependent LSH, which were introduced recently (16, 19), provably improve the query time polynomially. However, their *space partitions* are very different from the aforementioned practical heuristics (e.g., they are not spectral-based), and do not explain why data-aware *projections* help at all.

In this chapter, we formulate a semirandom model of data and show that data-aware projections provably guarantee good performance in this model, in contrast to random projections, which will not do the job.

As argued above, for worst-case inputs we are unlikely to beat the performance of random projections, and thus it seems justified to revert to the framework of smoothed analysis (142). We thus consider a semirandom model, where the dataset is formed by first taking an arbitrary (worst-case) set of n points in a k-dimensional subspace of $\mathbb{R}^d$, and then perturbing each point by adding to it Gaussian noise $N_d(0, \sigma^2 I_d)$. The query point is selected using a similar process. Our algorithms are able to find the query's nearest neighbor as long as there is a small gap (1 vs $1+\varepsilon$) in the distance to the nearest neighbor versus other points in the *unperturbed* space — this is a much weaker assumption than assuming the same for the perturbed points.

Most importantly, our results hold even when the noise magnitude is *much larger* than the distance to the nearest neighbor. The noise vector has length (about) $\sigma\sqrt{d}$, and so for $\sigma \gg 1/\sqrt{d}$, the noise magnitude exceeds the original distances. In such a case, a Johnson-Lindenstrauss projection to a smaller dimension will not work — the error due to the projection will lose all the information on the nearest neighbor, and it's not even clear what would be gained.

We describe the precise model in Section 5.2.

5.1.1 Algorithmic results

We propose two spectral algorithms for nearest neighbor search, which achieve essentially the same performance as NNS algorithms in k and $O(k \log k)$-dimensional space, respectively. These spectral algorithms rely on computing a PCA subspace or vector, respectively — the span of the singular vectors corresponding to the top singular values of an $n \times d$ matrix representing some n points in $\mathbb{R}^d$. Our algorithms are inspired by PCA-based methods that are commonly used in practice for high-dimensional NNS, and we believe that our rigorous analysis may help explain (or direct) those empirical successes. We defer the precise statements to the respective technical sections (specifically, Theorems 5.6.1 and 5.7.1), focusing here on a *qualitative* description.

The first algorithm performs *iterative PCA*. Namely, it employs PCA to extract a subspace of dimension (at most) k, identifies the points captured well by this subspace, and then repeats iteratively on the remaining points. The algorithm performs at most $O(\sqrt{d}\log n)$ PCAs in total, and effectively reduces the original NNS problem to $O(\sqrt{d}\log n)$ instances of NNS in k dimensions. Each of these NNS instances can be solved by any standard low-dimensional $(1+\varepsilon)$-approximate NNS, such as (21, 22, 23, 45, 49, 75), which can give, say, query time $(1/\varepsilon)^{O(k)}\log^2 n$. See Section 5.6, and the crucial technical tool it uses in Section 5.5. As a warmup, we initially introduce a simplified version of the algorithm for a (much) simpler model in Section 5.4.

The second algorithm is a variant of the aforementioned PCA tree, and constructs a tree that represents a recursive space partition. Each tree node corresponds to finding the top PCA direction, and partitioning the dataset into slabs perpendicular to this direction. We recurse on each slab until the tree reaches depth $2k$. The query algorithm searches the tree by following a small number of children (slabs) at each node. This algorithm also requires an additional preprocessing step that ensures that the dataset is not overly "clumped." The overall query time is $(k/\varepsilon)^{O(k)} \cdot (d\log n)^{O(1)}$. See Section 5.7.

While the first algorithm is randomized, the second algorithm is deterministic and its failure probability comes only from the semirandom model (randomness in the instance).

5.1.2 Related work

There has been work on understanding how various tree data structures adapt to a *low-dimensional* point set, including (57, 152). For example, (152) shows that PCA trees adapt to a form of "local covariance dimension," a spectrally-inspired notion of dimension, in the sense that a PCA tree halves the "diameter" of the pointset after a number of levels dependent on this dimension notion (as opposed to the ambient dimension d). Our work differs in a few respects. First, our datasets do not have a small local covariance dimension. Second, our algorithms have guarantees of performance and correctness for NNS for a worst-case query point (e.g., the true nearest neighbor can be any dataset point). In contrast, (152) proves a guarantee on diameter progress, which does not necessarily imply performance guarantees for NNS, and, in fact, may only hold for average query point (e.g., when the true nearest neighbor is random). Indeed, for algorithms considered in (152), it is easy to exhibit cases where NNS fails.[2]

For our model, it is tempting to design NNS algorithms that find the original k-dimensional subspace and thus "de-noise" the dataset by projecting the data onto it. This approach would require us to solve the ℓ_∞-regression problem with high precision.[3] Unfortunately, this problem is NP-hard in general (72), and the known algorithms are quite expensive, unless k is constant. Har-Peled and Varadarajan (80) present an algorithm for ℓ_∞-regression achieving $(1+\varepsilon)$-approximation in time $O(nde^{e^{O(k^2)}}\varepsilon^{-2k-3})$, which may be prohibitive when $k \geq \Omega(\sqrt{\log\log n})$. In fact, there is a constant $\delta > 0$ such that it is Quasi-NP-hard, i.e., implies NP $\subseteq$ DTIME$(2^{(\log n)^{O(1)}})$, to even find a $(\log n)^\delta$ approximation to the best fitting k-subspace when $k \geq d^\varepsilon$ for any fixed $\varepsilon > 0$ (149).

We also note that the problem of finding the underlying k-dimensional space is somewhat reminiscent of the learning mixtures of Gaussians problem (56); see also (20, 111, 151) and references therein. In the latter problem, the input is n samples generated from a mixture of k Gaussian distributions with unknown mean (called centers), and the goal is to identify these centers (the means of the k Gaussians). Our setting can be viewed as having n centers in a k-dimensional subspace of $\mathbb{R}^d$, and the input contains exactly one sample from each of the n centers. Methods known from learning mixtures of Gaussians rely crucially on obtaining multiple samples from the same center (Gaussian), and thus do not seem applicable here.

Finally, the problem of nearest neighbor search in Euclidean settings with "effective low-dimension" has received a lot of attention in the literature, including (32, 49, 78, 85, 90, 95) among many others. Particularly related is also the work (76), where the authors consider the case when the dataset is high-dimensional, but the query comes from a

[2] For example, if we consider the top PCA direction of a dataset and the median threshold, we can plant a query–near-neighbor pair on the two sides of the partition. Then, this pair, which won't affect top PCA direction much, will be separated in the PCA tree right from the beginning.

[3] As we explain later, related problems, such as ℓ_2-regression, would not be sufficient.

(predetermined) low-dimensional subspace. These results do not seem readily applicable in our setting because our dataset is really high-dimensional, say in the sense that the doubling dimension is $\Omega(d)$.

5.1.3 Techniques and ideas

We now discuss the main technical ideas behind our two algorithms. First, we explain why some natural ideas do *not* work. The very first intuitive line of attack to our problem is to compute a k-dimensional PCA of the pointset, project it into this k-dimensional subspace, and solve the k-dimensional NNS problem there. This approach fails because the noise is too large, and PCA only optimizes *the sum of distances* (i.e., an average quantity, as opposed to the "worst case" quantity). In particular, suppose most of the points lie along some direction $\vec{u}$ and only a few points lie in the remaining dimensions of our original subspace S (which we call sparse directions). Then, the k-PCA of the dataset will return a top singular vector close to $\vec{u}$, but the remaining singular vectors will be mostly determined by random noise. In particular, the points with high component along sparse directions may be very far away from the subspace returned by our PCA, and hence "poorly-captured" by the PCA space. Then, the NNS data structure on the projected points will fail for some query points. If the difference between a query point q and its nearest neighbor p^* is along $\vec{u}$, whereas the difference between q and the other, poorly-captured points is along the sparse directions, the poorly-captured points will cluster around q in the k-PCA space, at a distance much closer than $\|q - p^*\|$.

Our first algorithm, instead, runs k-PCAs *iteratively*, while pruning away points "well-captured" by the PCA (i.e., close to the PCA subspace). In particular, this allows us to discover the points in sparse directions in *later* iterations. Furthermore, to ensure correctness of nearest neighbor queries in the presense of large noise, we do not necessarily take all the top k singular values, but only those that exceed some threshold value; this guarantees that all directions in our PCA subspace have a large component inside U. Showing this guarantee analytically is nontrivial, starting with even the definition of what it means to be "close" for two spaces, which may have different dimensions. For this purpose, we employ the so-called $\sin\theta$ machinery, which was developed by Davis and Kahan (59) and by Wedin (154), to bound the perturbations of the singular *vectors* of a matrix in presence of noise. Notice the difference from the more usual theory of perturbations of the singular *values*. For example, in contrast to singular values, it is not true that the top singular vector is "stable" when we perturb a given matrix.

The actual algorithm has one more important aspect: in each iteration, the PCA space is computed on a *sample* of the (surviving) data points. This modification allows us to control spurious conditioning induced by earlier iterations. In particular, if instead we compute the PCA of the full data, once we argue that a vector $\tilde{p}$ "behaves nicely" in one iteration, we might effectively condition on the direction of its noise, potentially jeopardizing later iterations. (While we do not know if sampling is really necessary for the algorithm to work, we note that, practically, it is a very reasonable idea to speed up preprocessing nonetheless.)

The second algorithm is based on the PCA-tree, which partitions the space recursively, according to the top PCA direction. This can be seen as another (extreme) form of "iterative PCA." At each node, the algorithm extracts one top PCA direction, which always contains the "maximum" information about the dataset. Then it partitions the dataset into a few slabs, thereby partitioning the datasets into smaller parts "as quickly as possible." This allows the tree to narrow down on the sparse directions quicker. The performance of the PCA tree depends exponentially on its depth; hence the crux of the argument is to bound the depth. While it seems plausibly easy to show that a partitioning direction should never be repeated, this would give too loose a bound, as there could be a total of $\approx \exp(k)$ essentially distinct directions in a k-dimensional space. Instead, we perform a mild form of orthonormalization as we progress down the tree, to ensure only $O(k)$ directions are used in total. In the end, the query time is roughly $k^{O(k)}$, i.e., equivalent to a NNS in an $O(k \log k)$-dimensional space.

We note that this algorithm has two interesting aspects. First, one has to use *centered* PCA, i.e., PCA on the data centered at zero: otherwise, every small error in the PCA direction may move points a lot for subsequent iterations, misleading a noncentered PCA. Second, from time to time, we need to do "de-clumping" of the data, which essentially means that the data is sparsified if the points are too close to each other. This operation also appears necessary; otherwise, a cluster of points that are close in the original space might mislead the PCA due to their noise components. Furthermore, in contrast to the first algorithm, we cannot afford to iterate through $\approx d$ iterations to eliminate "bad" directions one by one.

5.2 The model

We assume throughout the dataset is generated as follows.[4] Let U be a k-dimensional subspace of $\mathbb{R}^d$. Let $P = \{p_1, \dots, p_n\}$ be a set of n points all living in U and having at least unit norm, and let $q \in U$ be a query point. We assume that $d = \Omega(\log n)$. The point set P is perturbed to create $\tilde{P} = \{\tilde{p}_1, \dots, \tilde{p}_n\}$ by adding to each point independent Gaussian noise, and the query point q is perturbed similarly. Formally,

$$\tilde{p}_i = p_i + t_i \text{ where } t_i \sim N_d(0, \sigma I_d), \qquad \forall p_i \in P, \tag{5.1}$$

$$\tilde{q} = q + t_q \text{ where } t_q \sim N_d(0, \sigma I_d). \tag{5.2}$$

Let us denote the nearest neighbor to q in P by p^* and let $\tilde{p}^*$ be its perturbed version. We shall actually consider the *near-neighbor* problem, by assuming that in the unperturbed space, there is one point $p^* \in P$ within distance 1 from the query, and all other points are at distance at least $1+\varepsilon$ from the query, for some known $0 < \varepsilon < 1$. Formally,

$$\exists p^* \in P \text{ such that } \|q - p^*\| \le 1 \text{ and } \forall p \in P \setminus \{p^*\},\ \|q - p\| \ge 1 + \varepsilon. \tag{5.3}$$

We note that even if there is more than one such point p^* so that $\|q - p^*\| \le 1$, our algorithms will return one of these close p^* correctly. Also our analysis in Section 5.6 extends trivially to show that for any x such that $x \ge 1$ and $\|q - p^*\| = x$, our first algorithm the iterative PCA actually returns a $(1+\varepsilon)$-approximate nearest neighbor to q. We omit the details of this extended case for ease of exposition.

5.2.1 Preliminary observations

For the problem to be interesting, we need that the perturbation does not change the nearest neighbor, i.e., $\tilde{p}^*$ remains the closest point to $\tilde{q}$. We indeed show this is the case as long as $\sigma \ll \varepsilon / \sqrt[4]{d \log n}$. Notice that the total noise magnitude is roughly $\sigma\sqrt{d}$, which can be much larger than 1 (the original distance to the nearest neighbor). Hence after the noise is added, the ratio of the distance to the nearest neighbor and to other (nearby) points becomes very close to 1. This is the main difficulty of the problem, as, for example, it is the case where random dimensionality reduction would lose nearest neighbor information. We recommend keeping in mind the following parameter settings: $k = 20$ and $\varepsilon = 0.1$ are

[4] An exception is the warm-up Section 5.4, where the noise is small adversarial.

constants, while $d = \log^3 n$ and $\sigma = \Theta(1/\log n)$ depend asymptotically on $n = |P|$. In this case, for example, our algorithms actually withstand noise of magnitude $\Theta(\sqrt{\log n}) \gg 1$.

Here and in the rest of the chapter, we will repeatedly employ concentration bounds for Gaussian noise, expressed as tail inequalities on the χ^2 distribution. We state here for reference bounds from (99), where χ_d^2 is the same distribution as $\|N_d(0, I_d)\|_2^2$. The term *with high probability (w.h.p.)* will mean that probability $1 - n^{-C}$ for sufficiently large $C > 0$.

Theorem 5.2.1 *((99)) Let $X \sim \chi_d^2$. For all $x \geq 0$,*

$$\Pr\left[X \geq d\left(1 + 2\sqrt{\tfrac{x}{d}}\right) + x\right] \leq e^{-x}, \quad \textit{and} \quad \Pr\left[X \leq d\left(1 - 2\sqrt{\tfrac{x}{d}}\right)\right] \leq e^{-x}.$$

Corollary 5.2.1 *For $n \geq 1$, let $X \sim \chi_d^2$. Then* $\Pr[|X - d| \geq d + 4\sqrt{d\log n} + 4\log n] \leq \frac{2}{n^4}$.

We now show that after the perturbation of P, q, the nearest neighbor of $\tilde{q}$ will remain $\tilde{p}^*$, w.h.p.

Lemma 5.2.1 *Consider the above model* (5.1)-(5.3) *for $n > 1$, $\varepsilon \in (0,1)$, dimensions $k < d = \Omega(\log n)$, and noise standard deviation $\sigma \leq c\varepsilon/\sqrt[4]{d\log n}$, where $c > 0$ is a sufficiently small constant. Then w.h.p. the nearest neighbor of $\tilde{q}$ (in $\tilde{P}$) is $\tilde{p}^*$.*

Proof. Write $(X_1, \ldots, X_d)^{\mathrm{T}} = \tilde{q} - q \sim N_d(0, \sigma I_d)$, and similarly $(Y_1, \ldots, Y_d)^{\mathrm{T}} = \tilde{p}^* - p \sim N_d(0, \sigma I_d)$. Let $Z_i = X_i - Y_i$, and note that the Z_i's are independent Gaussians, each with mean 0 and variance $2\sigma^2$. Then by direct computation

$$\|\tilde{q} - \tilde{p}^*\|^2 = 1 + \sum_{i=1}^{d} (X_i - Y_i)^2 + \sum_{i=1}^{d} (X_i - Y_i)(q_i - p_i^*) = 1 + \sum_{i=1}^{d} Z_i^2 + \sum_{i=1}^{d} Z_i(q_i - p_i^*). \tag{5.4}$$

For the term $\sum_{i=1}^d Z_i^2$, Theorem 5.2.1 gives us

$$\Pr\left[\left|\sum_{i=1}^{d} Z_i^2 - 2\sigma^2 d\right| \geq 2\sigma^2\left(x + d \cdot 2\sqrt{\tfrac{x}{d}}\right)\right] \leq 2e^{-x}.$$

Setting $x = 4\log n \leq O(d)$, observe that $2\sigma^2(x + 2\sqrt{xd}) \leq O(\sigma^2\sqrt{xd}) \leq O(c^2\varepsilon^2)$, and thus we have $\Pr\left[\left|\sum_{i=1}^d Z_i^2 - 2d\sigma^2\right| \geq O(c^2\varepsilon^2)\right] \leq \frac{2}{n^4}$.

Now the term $\sum_{i=1}^d Z_i(q_i - p_i^*)$ is a Gaussian with mean 0 and variance $\sum_{i=1}^d (q_i - p_i^*)^2 \operatorname{var}[Z_i] = 2\sigma^2\|q - p^*\|^2 = 2\sigma^2$, and thus with high probability $|\sum_{i=1}^d Z_i(q_i - p_i^*)| \leq$

$O(\sigma\sqrt{\log n})$. Substituting for σ and recalling $d = \Omega(\log n)$, the righthand-side can be bounded by $O(c\varepsilon)$. Altogether, with high probability

$$\|\tilde{q} - \tilde{p}^*\|^2 \leq 1 + 2d\sigma^2 \pm O(c^2\varepsilon^2) \pm O(c\varepsilon). \tag{5.5}$$

Similarly, for every other point $p \neq p^*$, with high probability,

$$\|\tilde{q} - \tilde{p}\|^2 \geq \|q - p\|^2 + 2d\sigma^2 \pm O(c^2\varepsilon^2) \pm O(c\varepsilon)\|q - p\|, \tag{5.6}$$

and we can furthermore take a union bound over all such points $p \neq p^*$. Now comparing Eqns. (5.5) and (5.6) when $\|q - p\|^2 \geq 1 + \varepsilon$ and $c > 0$ is sufficiently small, gives us the desired conclusion. ■

Remark 5.2.1 *The problem remains essentially the same if we assume the noise has no component in the space U. Indeed, we can absorb the noise inside U into the "original" points (P and q). With high probability, this changes the distance from q to every point in P by at most $O(\sigma\sqrt{k\log n}) \ll \varepsilon$. Hence, in the rest of the article, we will assume the noise is perpendicular to U.*

5.3 Short review of spectral properties of matrices

We review some basic definitions of spectral properties of matrices.

5.3.1 Spectral norm and principal component analysis

The *spectral norm* of a matrix $X \in \mathbb{R}^{n\times d}$ is defined as $\|X\| = \sup_{y\in\mathbb{R}^d:\|y\|=1}\|Xy\|$, where all vector norms $\|\cdot\|$ refer throughout to the ℓ_2-norm. The Frobenius norm of X is defined as $\|X\|_F = (\sum_{ij} X_{ij}^2)^{1/2}$, and let X^{T} denote the transpose of X. A *singular vector* of X is a unit vector $v \in \mathbb{R}^d$ associated with a *singular value* $s \in \mathbb{R}$ and a unit vector $u \in \mathbb{R}^n$ such that $Xv = su$ and $u^{\mathrm{T}}X = sv^{\mathrm{T}}$. (We may also refer to v and u as a pair of right-singular and left-singular vectors associated with s.)

Fact 5.3.1 *For every two real matrices X and Y of compatible dimensions (i) $\|X+Y\| \leq \|X\| + \|Y\|$; (ii) $\|XY\| \leq \|X\|\cdot\|Y\|$; and (iii) $\|X\| = \|X^{\mathrm{T}}\|$.*

We can think of i-th row in X as a point $x_i \in \mathbb{R}^d$, and define the corresponding point set $P(X) = \{x_1, \ldots, x_n\}$. Then a unit vector $y \in \mathbb{R}^d$ maximizing $\|Xy\|$ corresponds to a best-fit line for the point set $P(X)$. The PCA (and *Singular Value Decomposition (SVD)* more generally) is a classical generalization of this notion to a best-fit k-dimensional subspace for $P(X)$, as described next (see, e.g., (89, Chapter 1)).

Theorem 5.3.1 ((89)) *Let $X \in \mathbb{R}^{n\times d}$, and define the vectors $v_1,\ldots,v_d \in \mathbb{R}^d$ inductively by*

$$v_j = \operatorname*{arg\,max}_{\|v\|=1;\ \forall i<j, v^{\mathrm{T}}v_i=0} \|Xv\|$$

(where ties for any $\arg\max$ *are broken arbitrarily). Then each $V_j = \operatorname{span}\{v_1,\ldots,v_k\}$ attains the minimum of $\sum_{i=1}^{n} d(x_i,W)^2$ over all j-dimensional subspaces W. Furthermore, $v_1,\ldots,v_d$ are all singular vectors with corresponding singular values $s_1 = \|Xv_1\|,\ldots,s_d = \|Xv_d\|$, and clearly $s_1 \geq \ldots \geq s_d$.*

We will later need the following basic facts (see, e.g., (145)). We denote the singular values of a matrix $X \in \mathbb{R}^{n\times d}$ by $s_1(X) \geq s_2(X) \geq \ldots \geq s_d(X)$.

Fact 5.3.2 *Let $P(X) \subset \mathbb{R}^d$ be the point set corresponding to the rows of a matrix $X \in \mathbb{R}^{n\times d}$. Then*

1. $\|X\|_F^2 = \sum_{p\in P(X)} \|p\|^2 = \sum_{i=1}^{d} s_i(X)^2$ *and* $\|X\| = s_1(X)$.
2. *$P(X)$ lies in a subspace of dimension k if and only if $s_{k+1}(X) = 0$.*

Fact 5.3.3 *For any matrix X, let $X^{\mathrm{T}}X$ be the* covariance *matrix of X. Then the right singular vectors of X are also right singular vectors of $X^{\mathrm{T}}X$. Also, $s_i(X^{\mathrm{T}}X) = s_i^2(X)$.*

Fact 5.3.4 *For matrices X and E of compatible dimensions, $|s_j(X+E) - s_j(X)| \leq \|E\|$.*

5.3.2 Spectral norms of random matrices

In our analysis, we will also need bounds on norms of random matrices, which we derive using standard random matrix theory. We state below a bound on the spectral norm of $T \in \mathbb{R}^{n\times d}$, a matrix of iid Gaussian vectors. We also consider its restriction T_A to any subset of rows of T, and bound the spectral norm in terms of the size of the subset, $s = |A|$, expressed as a function $\eta(s) = O(\sigma\sqrt{s\cdot\log n})$.

Theorem 5.3.2 ((98, 137)) *Let matrix $T \in \mathbb{R}^{n\times d}$ have entries drawn independently from $N(0,\sigma)$. Then with probability approaching 1 asymptotically as n and d increase, $\|T\| \leq 3\sigma \max\{\sqrt{n}, \sqrt{d}\}$.*

Lemma 5.3.1 *With high probability, for every subset A of the rows of T, with $|A| \geq d$, the corresponding submatrix T_A of T, has spectral norm $\|T_A\| \leq \eta(|A|) = O(\sigma\sqrt{|A|\cdot \log n})$.*

Proof. Fix some A of size s. It is known from random matrix theory that $\|T_A\| \leq \sigma\sqrt{Cs\log n}$ with probability at least $1-e^{-\Omega(Cs\log n)} = 1-n^{-\Omega(Cs)}$ (137, Proposition 2.4). For a large enough constant $C > 1$, since there are at most $\binom{n}{s} \leq (n/s)^{O(s)}$ such subsets A, by a union bound, with high probability none of the sets will fail. Another union bound over all sizes s completes the claim. ■

5.4 Warmup: Iterative PCA under small adversarial noise

To illustrate the basic ideas in our "iterative PCA" approach, we first study it in an alternative, simpler model that differs from Section 5.2 in that the noise is *adversarial* but of *small magnitude*. The complete "iterative PCA" algorithm for the model from Section 5.2 will appear in Section 5.6.

In the bounded noise model, for fixed $\varepsilon \in (0,1)$, we start with an n-point dataset P and a point q, both lying in a k-dimensional space $U \subset \mathbb{R}^d$, such that

$$\exists p^* \in P \text{ such that } \|q-p^*\| \leq 1 \text{ and } \forall p \in P\setminus\{p^*\},\ \|q-p\| \geq 1+\varepsilon. \tag{5.7}$$

The set $\tilde{P}$ consists of points $\tilde{p}_i = p_i + t_i$ for all $p_i \in P$, where the noise t_i is arbitrary, but satisfies $\|t_i\| \leq \varepsilon/16$ for all i. Similarly, $\tilde{q} = q + t_q$ with $\|t_q\| \leq \varepsilon/16$.

Theorem 5.4.1 *Suppose there is a $(1+\varepsilon/4)$-approximate NNS data structure for n points in a k-dimensional Euclidean space with query time F_{query}, space F_{space}, and preprocessing time F_{prep}. Then for the above adversarial-noise model, there is a data structure that preprocesses $\tilde{P}$, and on query $\tilde{q}$ returns $\tilde{p}^*$. This data structure has query time $O((dk+F_{\text{query}})\log n)$, space $O(F_{\text{space}})$, and preprocessing time $O(n+d^3+F_{\text{prep}})$.*

First we show that the nearest neighbor "remains" p^* even after the perturbations (similarly to lemma 5.2.1. Let $\alpha = \varepsilon/16$.

Claim 5.4.1 *The nearest neighbor of $\tilde{q}$ in $\tilde{P}$ is $\tilde{p}^*$.*

Proof. For all i, we have $\|\tilde{p}_i - p_i\| \leq \|t_i\| \leq \alpha$, hence by the triangle inequality, $\|\tilde{q}-\tilde{p}^*\| \leq \|\tilde{q}-q\| + \|q-p^*\| + \|p^*-\tilde{p}^*\| \leq \|q-p^*\| + 2\alpha$. For all $p \neq p^*$, a similar argument gives $\|\tilde{q}-\tilde{p}\| \geq \|q-p^*\| + \varepsilon - 2\alpha$. ∎

We now describe the algorithm used to prove Theorem 5.4.1. Our algorithm first finds a small collection $\mathscr{U}$ of k-dimensional subspaces, such that every point of $\tilde{P}$ is "captured well" by at least one subspace in $\mathscr{U}$. We find this collection $\mathscr{U}$ by iteratively applying PCA, as follows (see Algorithm 12). First compute the top (principal) k-dimensional subspace of $\tilde{P}$. It "captures" all points $\tilde{p} \in \tilde{P}$ within distance $\sqrt{2}\alpha$ from the subspace. Then we repeat on the remaining noncaptured points, if any are left. In what follows, let $p_{\tilde{U}}$ denote the projection of a point p onto $\tilde{U}$, and define the distance between a point x and a set (possibly a subspace) S as $d(x,S) = \inf_{y \in S}\|x-y\|$.

```
$j \leftarrow 0$; $\tilde{P}_0 \leftarrow \tilde{P}$
**while** $\tilde{P}_j \neq \emptyset$ **do**
  $\tilde{U}_j \leftarrow$ the $k$-dimensional PCA subspace of $\tilde{P}_j$
  $M_j \leftarrow \{\tilde{p} \in \tilde{P}_j :\ d(\tilde{p}, \tilde{U}_j) \leq \sqrt{2}\alpha\}$
  $\tilde{P}_{j+1} \leftarrow \tilde{P}_j \setminus M_j$
  $j \leftarrow j+1$
**end while**
**return** $\tilde{\mathscr{U}} = \{\tilde{U}_0, \ldots, \tilde{U}_{j-1}\}$ and the associated point sets $\{M_0, M_1, \ldots, M_{j-1}\}$.
```

Algorithm 12: Iteratively locate subspaces

The remainder of the preprocessing algorithm just constructs for each subspace $\tilde{U} \in \mathscr{U}$ a data structure for k-dimensional NNS, whose dataset is the points captured by $\tilde{U}$ projected onto this subspace $\tilde{U}$ (treating $\tilde{U}$ as a copy of $\mathbb{R}^k$). Overall, the preprocessing phase comprises of $O(\log n)$ PCA computations and constructing $O(\log n)$ data structures for a k-dimensional NNS.

The query procedure works as follows. Given a query point $\tilde{q}$, project $\tilde{q}$ onto each $\tilde{U} \in \mathscr{U}$ to obtain $\tilde{q}_{\tilde{U}}$, and find in the data structure corresponding to this $\tilde{U}$ a $(1+\varepsilon/4)$-approximate nearest neighbor point $\tilde{p}_{\tilde{U}}$ to $\tilde{q}_{\tilde{U}}$. Then compute the distance between $\tilde{q}$ and each $\tilde{p}$ (original points corresponding to $\tilde{p}_{\tilde{U}}$), and report the the closest one to $\tilde{q}$.

We now proceed to analyze the algorithm.

Claim 5.4.2 *Algorithm 12 terminates within $O(\log n)$ iterations.*

Proof. Let U be the PCA subspace of P and let $\tilde{U}$ be the PCA subspace of $\tilde{P}$. Since $\tilde{U}$ minimizes (among all k-dimensional subspaces) the sum of squared distances from all $\tilde{p} \in \tilde{P}$ to $\tilde{U}$,

$$\sum_{\tilde{p}\in\tilde{P}} d(\tilde{p},\tilde{U})^2 \le \sum_{\tilde{p}\in\tilde{P}} d(\tilde{p},U)^2 \le \sum_{\tilde{p}\in\tilde{P}} \|\tilde{p}-p\|^2 \le \alpha^2 n.$$

Hence, at most half of the points in $\tilde{P}$ may have distance to $\tilde{U}$ which is greater than $\sqrt{2}\alpha$. The current set M will capture the other (at least a half fraction) points, and the algorithm then proceeds on the remaining set. Each subsequent iteration thus decreases the number of points by a constant factor. After $O(\log n)$ iterations all points of $\tilde{P}$ must be captured. ■

Claim 5.4.3 *The data structure for the subspace $\tilde{U}$ that captures $\tilde{p}^*$ always reports this point as the $(1+\varepsilon/4)$-approximate nearest neighbor of $\tilde{q}$ (in $\tilde{U}$).*

We can now complete the proof of Theorem 5.4.1. By Claim 5.4.3, $\tilde{p}^*$ is always reported by the k-dimensional data structure it is assigned to. But this is the closest point overall, by Claim 5.4.1, and thus our algorithm eventually reports this point $\tilde{p}^*$, which proves the correctness part of Theorem 5.4.1. To argue the time and space guarantees, we just note that computing one PCA on n points takes time $O(n+d^3)$, and there are in total $O(\log n)$ PCAs to compute, and obviously also k-dimensional NNS data structures to query against.

5.5 Stability of a top PCA subspace

Before continuing to the full iterative-PCA algorithm, we need to address the challenge of controlling the stability of the PCA subspace under random noise. In particular, we will need to show that the PCA subspace $\tilde{U}$ computed from the noisy dataset $\tilde{P}$ is "close" to the original subspace U. We establish this rigorously using the $\sin\theta$ machinery developed by Davis and Kahan (59) and by Wedin (154).

Throughout, $s_j(M)$ denotes the j-th largest singular value of a real matrix M, and $\|M\| = s_1(M)$ denotes its spectral norm, while $\|M\|_F$ denotes the Frobenius norm of M. All vector norms, i.e. $\|v\|$ for $v \in \mathbb{R}^d$, refer to the ℓ_2-norm.

5.5.1 Wedin's $\sin\theta$ Theorem

The $\sin\theta$ *distance* between two subspaces B and A of $\mathbb{R}^d$ is defined as

$$\sin\theta(B,A) = \max_{x\in B, \|x\|=1} \min_{y\in A} \|x-y\|.$$

Observe that the minimum here is just the distance to a subspace $\mathrm{dist}(x,A)$, and it is attained by orthogonal projection. Thus, for all $x' \in B$ (not necessarily of unit length) $\mathrm{dist}(x',A) = \|x'\|\cdot \mathrm{dist}\left(\frac{x'}{\|x'\|},A\right) \leq \|x'\|\cdot\sin\theta(B,A)$. See Figure 5.1 for an example.

For a matrix $X \in \mathbb{R}^{n\times d}$ and an integer $m \in \{1,\ldots,d\}$, let $R_m(X)$ (resp. $L_m(X)$) denote the matrix formed by the top m right (resp. left) singular vectors of X taken in column (resp. row) order, and define $SR_m(X)$ (resp. $SL_m(X)$) as the subspace whose basis is these right (resp. left) singular vectors.

Now consider a matrix $X \in \mathbb{R}^{n\times d}$, and add to it a "perturbation" matrix $Y \in \mathbb{R}^{n\times d}$, writing $Z = X+Y$. The theorem below bounds the $\sin\theta$ distance between the top singular spaces before and after the perturbation, namely the subspaces $SR_m(Z)$ and $SR_k(X)$ for some dimensions m and k, in terms of two quantities:

1. The projection of the perturbation Y on $SR_m(Z)$ and on $SL_m(Z)$. Let $Y_R = \|YR_m(Z)\|$ and $Y_L = \|L_m(Z)Y^{\mathrm{T}}\|$.

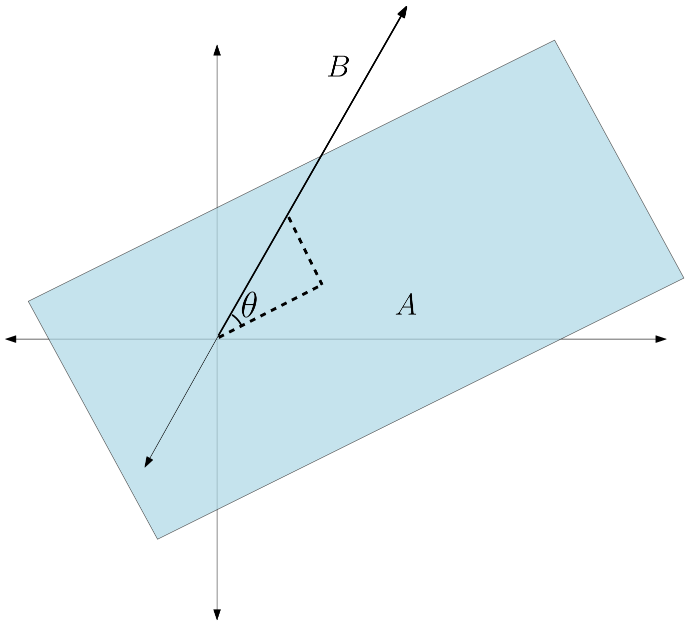


Figure 5.1. Illustration of Wedin's $\sin\theta$ theorem

2. The gap between the top m singular values of Z and the bottom $d-k$ singular values of X. Formally, define $\gamma = s_m(Z) - s_{k+1}(X)$.

Theorem 5.5.1 (Wedin's $\sin\theta$ Theorem (154)) *In the above setting, if $m \le k \le d$ and $\gamma > 0$, then*

$$\sin\theta(SR_m(Z), SR_k(X)) \le \frac{\max\{Y_R, Y_L\}}{\gamma}.$$

5.5.2 Instantiating the $\sin\theta$ theorem

We now apply the $\sin\theta$-Theorem to our semirandom model from Section 5.2. Let $X \in \mathbb{R}^{n\times d}$ be the matrix corresponding to our original point set P (of size $n \ge d$) lying in a subspace U of dimension $k \le d$. Let $T \in \mathbb{R}^{n\times d}$ be a perturbation matrix (noise), and then $\tilde{X} = X + T$ corresponds to our perturbed point set $\tilde{P}$. Our next theorem uses $\|T\|$ directly without assuming anything about its entries, although in our context where the entries of T are drawn from independent Gaussians of magnitude σ, Theorem 5.3.2 implies that w.h.p. $\|T\| \le O(\sigma\sqrt{n+d})$. In fact, if the matrix T is random, then m (and possibly also γ) should be interpreted as random variables that depend on T.

Theorem 5.5.2 *Let $\tilde{X} = X + T$ be defined as above, and fix a threshold $\gamma_1 > 0$. If $m \le k$ is such that at least m singular values of $\tilde{X}$ are at least γ_1, then*

$$\sin\theta(SR_m(\tilde{X}), SR_k(X)) \le \frac{\|T\|}{\gamma_1},$$

where $SR_k(M)$ denotes, as before, the span of the top k right-singular vectors of a matrix M.

Proof. Towards applying Theorem 5.5.1, define $T_R = \|TR_m(\tilde{X})\|$ and $T_L = \|L_m(\tilde{X})T^{\mathrm{T}}\|$. The columns of $R_m(\tilde{X})$ being orthonormal implies $\|R_m(\tilde{X})\| \le 1$, and now by Fact 5.3.1, $T_R = \|TR_m(\tilde{X})\| \le \|T\|$. We can bound also T_L similarly. Recalling Fact 5.3.2, X has at most k nonzero singular values because the point set P lies in a k-dimensional subspace, hence the gap is $\gamma = s_m(\tilde{X}) - 0 \ge \gamma_1$. Plugging this into the $\sin\theta$ Theorem yields the bound $\sin\theta(SR_m(\tilde{X}), SR_k(X)) \le \|T\|/\gamma \le \|T\|/\gamma_1$. ■

5.6 Iterative PCA algorithm

We now present the iterative PCA algorithm, that solves the NNS problem for the semirandom model from Section 5.2. In particular, the underlying pointset lives in a

k-dimensional space, but each point is also added a Gaussian noise $N_d(0, \sigma^2 I_d)$, which has norm potentially much larger than the distance between a query and its nearest neighbor. The algorithm reduces the setting to a classical k-dimensional NNS problem.

Theorem 5.6.1 *Suppose there is a $(1+\varepsilon/8)$-approximate NNS data structure for n points in a k-dimensional space with query time F_{query}, space F_{space}, and preprocessing time F_{prep}. Assume the Gaussian-noise model* (5.1)-(5.3)*, with $\sigma(k^{1.5}\sqrt{\log n} + \sqrt[4]{k^3 d \log n}) < c\varepsilon$ for sufficiently small constant $c > 0$.*

Then there is a data structure that preprocesses $\tilde{P}$, and on query $\tilde{q}$ returns $\tilde{p}^*$ with high probability. This data structure has query time $O((dk + F_{\text{query}})\sqrt{d \log n} + d^{O(1)})$, uses space $O(F_{\text{space}}\sqrt{d \log n} + d^{O(1)})$, and preprocessing time $O((nd^2 + d^3 + F_{\text{prep}})\sqrt{d \log n})$.

5.6.1 Algorithm description

The iterative-PCA algorithm computes a collection $\mathcal{U}$ of $O(\sqrt{d \log n})$ subspaces, such that every point in the perturbed dataset $\tilde{P}$ is within squared distance $\Psi = d\sigma^2 + 0.001\varepsilon^2$ of some subspace in the collection $\mathcal{U}$. For each such subspace $\tilde{U}_j^S \in \mathcal{U}$, we project onto $\tilde{U}_j^S$ the points captured by this subspace $\tilde{U}_j^S$, and construct on the resulting pointset a k-dimensional NNS data structure. We consider only singular vectors corresponding to sufficiently large singular values, which helps ensure robustness to noise. In particular, this threshold is $\delta(n) \triangleq c\varepsilon\sqrt{\frac{n}{k}}$ for small constant $c \leq 0.001$. Also, the PCA space is computed on a sample of the current pointset only.

See Algorithm 13 for a detailed description of computing $\mathcal{U}$.

We now present the overall NNS algorithm in detail. The preprocessing stage runs Algorithm 13 on the pointset $\tilde{P}$, stores its output, and constructs a k-dimensional NNS data structure for each of the pointsets $M_0, \ldots, M_{j-1}$ (here j refers to the final value of this variable). Note that we also have a "left-over" set $R = \tilde{P}_j \bigcup \cup_{l=0}^{j-1} \tilde{P}_l^S$, which includes the points remaining at the end plus the sampled points used to construct the subspaces $\mathcal{U}$.

The query stage uses those j data structures to compute a $(1+\varepsilon/8)$-approximates NNS of q in each of $M_0, \ldots, M_{j-1}$, and additionally finds the NNS of q inside R by exhaustive search. It finally reports the closest point found. In the rest of this section, we will analyze this algorithm, thus proving Theorem 5.6.1.

We make henceforth three assumptions that hold without loss of generality. First, we

```
Define $\Psi \triangleq d\sigma^2 + 0.001\varepsilon^2,\ r \triangleq O(d^9 k^3 \frac{\log n}{\varepsilon^2\sigma^2}),$ and $\delta(n) \triangleq c\varepsilon\sqrt{\frac{n}{k}}$ for a small constant $c \leq 0.001$.
$j \leftarrow 0, \tilde{P}_0 \leftarrow \tilde{P}$
while $|\tilde{P}_j| > r$ do
    Sample $r$ points from $\tilde{P}_j$ (with repetition) to form the set/matrix $\tilde{P}^S_j$
    $m \leftarrow$ number of singular values of $\tilde{P}^S_j$ that are at least $\delta(r)$
    $\tilde{U}^S_j \leftarrow$ the subspace spanned by the $m$ top singular vectors of $\tilde{P}^S_j$
    $M_j \leftarrow$ all $\tilde{p} \in \tilde{P}_j \setminus \tilde{P}^S_j$ at distance $\text{dist}(\tilde{p}, \tilde{U}^S_j) \leq \sqrt{\Psi}$
    $\tilde{P}_{j+1} \leftarrow \tilde{P}_j \setminus (M_j \cup \tilde{P}^S_j)$
    $j \leftarrow j+1$
end while
return the subspaces $\tilde{\mathscr{U}} = \{\tilde{U}^S_0, \ldots, \tilde{U}^S_{j-1}\}$, their pointsets $\{M_0, M_1, \ldots, M_{j-1}\}$, and the remaining set $R = \tilde{P}_j \bigcup \cup_{l=0}^{j-1} \tilde{P}^S_l$.
```

Algorithm 13: Iteratively locate subspaces.

assume that $\|p_i\| \geq 1$, which is without loss of generality as we can always move the pointset away from the origin. Overall, this ensures that $\|P\|_F^2 \geq |P|$.

Second, we assume that all points $\tilde{P}$ have norm at most $L \triangleq d^{3/2}$, which follows by applying a standard transformation of partitioning the dataset by a randomly shifted grid with side-length d. This transformation ensures that the query and the nearest neighbor, at distance $O(\sigma\sqrt{d})$ are in the same grid cell with probability at least $1-o(1)$ (see, e.g., (8)).

Third, we assume that $\sigma \gg \varepsilon/\sqrt{d}$, as otherwise we can apply the algorithm from Section 5.4 directly. (The algorithm in the current section works also for small σ, but the exposition becomes simpler if we assume a larger σ.) In the context of our model of Section 5.2 and lemma 5.2.1, this is equivalent to asserting $d \gg \log n$.

Finally, we remark that the algorithm can be changed to not use explicitly the value of σ, by taking only the closest $O\left(\sqrt{\frac{\log n}{d}}\right)$ fraction of points to a given space $\tilde{U}_j^S$. We omit the details.

5.6.2 Analysis

We now present a high-level overview of the proof. First, we characterize the space $\tilde{U}_j^S$, and in particular show that it is close to (a subspace of) the original space U, using the $\sin\theta$ machinery and matrix concentration bounds. Second, we use the closeness of $\tilde{U}_j^S$ to U to argue that: (a) projection of the noise onto $\tilde{U}_j^S$ is small; and (b) the projection of a point $\tilde{p}$ is approximately $\|p\|$, *on average*. Third, we use these bounds to show that the space $\tilde{U}_j^S$ captures a good fraction of points to be put into M_j, thus allowing us to bound the number of iterations. Fourth, we show that, for each point $\tilde{p} = p + t$ that has been “captured” into M_j, its projection into $\tilde{U}_j^S$ is a faithful representation of p, in the sense that, for such a point, the distance to the projection of $\tilde{q}$ onto $\tilde{U}_j^S$ is close to the original distance (before noise). This will suffice to conclude that the k-dimensional NNS for that set M_j shall return the right answer (should it happen to have the nearest neighbor p^*).

Slightly abusing notation, let P represent both the pointset and the corresponding $n \times d$ matrix, and similarly for $\tilde{P}$ or a subset thereof like $\tilde{P}_j$. Let T be the noise matrix, i.e., its rows are the vectors t_i and $\tilde{P} = P + T$.

Using bounds from random matrix theory (see lemma 5.3.1), w.h.p. every restriction of T to a subset of at least d rows gives a submatrix T' of spectral norm $\|T'\| \leq \eta(|T'|) = O(\sigma\sqrt{|T'| \cdot \log n})$. In addition, by Corollary 5.2.1 and the parameters of our model in

Theorem 5.6.1, w.h.p.

$$\forall p_i \in P, \qquad |\|t_i\|^2 - \sigma^2 d| \le 0.0001\varepsilon^2. \tag{5.8}$$

We assume in the rest of the proof that these events occur. Since both are high probability events, we may use a union bound and assume they occur over all iterations without any effect of conditioning on the points.

The thrust of our proof below is to analyze one iteration of Algorithm 13. We henceforth use j to denote an arbitrary iteration (not its final value), and let $n_j = |\tilde{P}_j| > r$ denote the number of points at that iteration.

5.6.3 Analysis: Characterization of the PCA space of the sampled set

Define $\tilde{U}_j^S$ and $\tilde{U}_j$ to be the PCA space of $\tilde{P}_j^S$ and $\tilde{P}_j$, respectively, i.e., full current set and sampled set. Suppose the dimension of $\tilde{U}_j^S$ and $\tilde{U}_j$ is $m \le k$ and $m \le \ell \le k$, respectively, where m is set according to the thresholding step in Algorithm 13 and ℓ will be specified later. We show that the computed PCA space $\tilde{U}_j^S$ is close to U using the $\sin\theta$ machinery established in Section 5.5.2. We consider the point sets as matrices, and concatenate the following two observations for deriving our result:

- The PCA space of sampled noisy set (scaled) is close to that of the full noisy set.
- The PCA space of the noisy set is close to that of the unperturbed set.

We now analyze the effects of sampling. We sample r points from the current set $\tilde{P}_j$ of size n_j. We use the following standard matrix concentration.

Theorem 5.6.2 (Rudelson and Vershynin (136), Thm 3.1) *Suppose we sample (with replacement) r row vectors from an n-size set $A \subset \mathbb{R}^d$ (represented as a $n \times d$ matrix), and call them set $Y = \{y_1, \ldots y_r\}$. Then, for any $t \in (0,1)$, and $L = \max_{a \in A} \|a\|$:*

$$\Pr\left[\left\|\frac{n}{r}\sum_{y\in Y} y^{\mathrm{T}}y - A^{\mathrm{T}}A\right\| > t\|A^{\mathrm{T}}A\|\right] \le 2e^{-\Omega\left(\left(t^2/L^2\right)\cdot r/\log r\right)}.$$

Corollary 5.6.1 *For $t \in (0,1)$ if we sample $r = O(\log^2 n \cdot L^2/t^2)$ vectors from $\tilde{P}_j$ to obtain $\tilde{P}_j^S$, we have that w.h.p.:*

$$\|\frac{n_j}{r}(\tilde{P}_j^S)^{\mathrm{T}}\tilde{P}_j^S - \tilde{P}_j^{\mathrm{T}}\tilde{P}_j\| \le L^2 n_j \cdot t.$$

Proof. Instantiate the theorem for $A = \tilde{P}_j$ to obtain $\|\frac{n_j}{r}(\tilde{P}^s_j)^{\mathrm{T}}\tilde{P}^s_j - \tilde{P}^{\mathrm{T}}_j\tilde{P}_j\| \leq t\|\tilde{P}^{\mathrm{T}}\tilde{P}\|$. We simplify the right hand side of this equation. First, by Fact 5.3.3 we have that $t\|\tilde{P}^{\mathrm{T}}_j\tilde{P}_j\| = t\|\tilde{P}_j\|^2$. Next by Fact 5.3.2, $t\|\tilde{P}_j\|^2 = \sum_{\tilde{p}\in\tilde{P}_j}\|\tilde{p}\|^2 t \leq n_j \max_{\tilde{p}\in\tilde{P}_j}\|\tilde{p}\|^2 t \leq L^2 n_j t$. To prove this event succeeds with high probability over all points n, we need only substitute the value of r directly into the theorem statement. ■

Corollary 5.6.2 *We set $t = O\left(\frac{\varepsilon\sigma\sqrt{\log n}}{L^2k^{1.5}}\right)$, for which we need to sample $r = \Omega(L^6k^3\log n/\varepsilon^2\sigma^2) = \Omega(d^9k^3\log n/\varepsilon^2\sigma^2)$. Then we obtain:*

$$\|\frac{n_j}{r}(\tilde{P}^s_j)^{\mathrm{T}}\tilde{P}^s_j - \tilde{P}^{\mathrm{T}}_j\tilde{P}_j\| \leq n_j \cdot O\left(\varepsilon\sigma\frac{\sqrt{\log n}}{k^{1.5}}\right) \ll \left(\frac{\delta(n_j)}{k}\right)^2,$$

for σ in the range given by our model of Theorem 5.6.1.

We now aim to show that $\sin\theta(\tilde{U}^s_j,\tilde{U}_j)$ and $\sin\theta(\tilde{U}_j,U_j)$ are small, and the triangle inequality for $\sin\theta$ will then show that $\sin\theta(\tilde{U}^s_j,U_j)$ is also small. Recall first that the $\sin\theta$ machinery of Theorem 5.5.2 requires a "gap" between the bottom most singular value considered in one subspace and the top most not considered in the other. We observe the following lemma:

Lemma 5.6.1 *There exists ℓ, where $m \leq \ell \leq k+1$, such that $s_\ell(\tilde{P}_j) - s_{\ell+1}(\tilde{P}_j) \geq \Omega\left(\frac{\delta(n_j)}{k}\right)$. Hence $s_\ell(\tilde{P}_j) - s_{k+1}(P_j) \geq \Omega\left(\frac{\delta(n_j)}{k}\right)$.*

Proof. First recall that by the threshold step of our algorithm, $s_m\left(\frac{n_j}{r}(\tilde{P}^s_j)^{\mathrm{T}}\tilde{P}^s_j\right) \geq \delta(n_j)^2$. Now by Fact 5.3.4 and r set as in Corollary 5.6.2, we have that $s_m(\tilde{P}^{\mathrm{T}}_j\tilde{P}_j) \geq \delta(n_j)^2 - \delta(n_j)^2/k^2 \geq \frac{3}{4}\delta(n_j)^2$. Hence we have $s_m(\tilde{P}_j) > 3\delta(n_j)/4$. Also since P_j is drawn from a k- dimensional subspace, $s_{k+1}(P_j) = 0$ and therefore $s_{k+1}(\tilde{P}_j) \leq \|P_j - \tilde{P}_j\| = O(\sigma\sqrt{n_j\log n}) < \delta(n_j)/16$ by lemma 5.3.1 and the parameters of our model. Now since $s_m(\tilde{P}_j) \geq 3\delta(n_j)/4$ and $s_{k+1}(\tilde{P}_j) \leq \delta(n_j)/16$, then there must exist ℓ with $m \leq \ell \leq k+1$ that satisfies the claim. See Figure 5.2 for illustration of the argument. ■

Using now that $a^2 - b^2 \geq (a-b)^2$ for $a \geq b \geq 0$ we obtain:

Corollary 5.6.3 *There exists ℓ, $m \leq \ell \leq k+1$, such that $s_\ell\left(\tilde{P_j}^{\mathrm{T}}\tilde{P_j}\right) - s_{\ell+1}\left(\tilde{P_j}^{\mathrm{T}}\tilde{P_j}\right) \geq \Omega\left(\frac{\delta(n_j)^2}{k^2}\right)$. Hence $s_m\left(\frac{n_j}{r}\left(\tilde{P}^s_j\right)^{\mathrm{T}}\tilde{P}^s_j\right) - s_{\ell+1}(\tilde{P}^{\mathrm{T}}_j\tilde{P}_j) \geq \Omega\left(\frac{\delta(n_j)^2}{k^2}\right)$.*

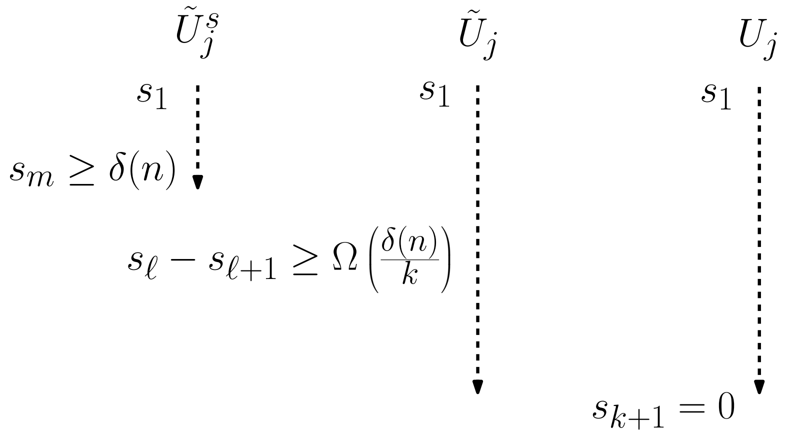


Figure 5.2. Take s_l as the last singular value included in $\tilde{U}_j$.

Now crucially using this ℓ of lemma 5.6.1 to define our cut off point for the singular vectors included in $\tilde{U}_j$, and instantiating the $\sin\theta$ theorem, we obtain the following:

Lemma 5.6.2 $\sin\theta(\tilde{U}_j^s, \tilde{U}_j) \leq O\left(\frac{\sigma k^{1.5}\sqrt{\log n}}{\varepsilon}\right)$ *and* $\sin\theta(\tilde{U}_j, U_j) \leq O\left(\frac{\sigma k^{1.5}\sqrt{\log n}}{\varepsilon}\right)$.

Proof. First by lemma 5.6.1, we have $s_l(\tilde{P}_j) - s_{k+1}(P_j) \geq O\left(\frac{\delta(n)}{k}\right)$ and recall by lemma 5.3.1, $\|\tilde{P}_j - P_j\| = O(\sigma\sqrt{n_j \log n})$. So instantiating the $\sin\theta$ theorem, Theorem 5.5.2:

$$\begin{aligned}
\sin\theta(\tilde{U}_j, U_j) &= \frac{\|\tilde{P}_j - P_j\|}{s_l(\tilde{P}_j) - s_{k+1}(P_j)} \\
&= O\left(\frac{\sigma\sqrt{n_j \log n}}{\delta(n_j)/k}\right) \\
&= O\left(\frac{\sigma\sqrt{n_j \log n}}{\varepsilon\sqrt{n_j}/k^{1.5}}\right) = O\left(\frac{\sigma k^{1.5}\sqrt{\log n}}{\varepsilon}\right).
\end{aligned}$$

Similarly for $\sin\theta(\tilde{U}_j^s, \tilde{U}_j)$, we upper bound $\|\frac{n_j}{r}(\tilde{P}_j^s)^{\mathrm{T}}\tilde{P}_j^s - \tilde{P}_j^{\mathrm{T}}\tilde{P}_j\|$ by Corollary 5.6.2. We lower bound the "gap" in singular values using Corollary 5.6.3, and hence by instantiating the $\sin\theta$ theorem:

$$\sin\theta(\tilde{U}_j^S, \tilde{U}_j) = O\left(\frac{\varepsilon n_j \sigma\sqrt{\log n}/k^{1.5}}{\delta(n_j)^2/k^2}\right)$$
$$= O\left(\frac{\varepsilon n_j \sigma\sqrt{\log n}/k^{1.5}}{\varepsilon^2 n_j/k^3}\right) = O\left(\frac{\sigma k^{1.5}\sqrt{\log n}}{\varepsilon}\right).$$

■

Since $\sin\theta$ is concave in the right regime, lemma 5.6.2 now gives us as a simple corollary the main claim of this subsection:

$$\sin\theta(\tilde{U}_j^S, U_j) \le O\left(\frac{\sigma k^{1.5}\sqrt{\log n}}{\varepsilon}\right). \tag{5.9}$$

5.6.4 Analysis: Noise inside the PCA space

We now show that the noise vector t_i of each point $\tilde{p}_i = p_i + t_i$ has a small component inside $\tilde{U}_j^S$. We use the $\sin\theta$ bound in Eqn. (5.9) for this. (Tighter analysis is possible directly via the randomness of the vector t_i on the first iteration, but conditioning of points selected together at each iteration of our algorithm makes this complicated at latter stages.)

Following Remark 5.2.1, we shall assume that noise t_i is perpendicular to U. Define $V \in \mathbb{R}^{d\times k}$ as the projection matrix onto the space U, so e.g. $t_i V$ is the zero vector, and define analogously $\tilde{V}_j^S \in \mathbb{R}^{d\times m}$ to be the projection matrix onto the m-dimensional space $\tilde{U}_j^S$.

Lemma 5.6.3 $\|t_i\tilde{V}_j^S\| \le \|t_i\| \cdot \sin\theta(\tilde{U}_j^S, U)$.

Proof. Let x be a unit vector in the direction of $t_i\tilde{V}_j^S$, namely, $x = \frac{t_i\tilde{V}_j^S}{\|t_i\tilde{V}_j^S\|}$. This implies $\|t_i\tilde{V}_j^S\| = x^{\mathrm{T}} t_i$. Now decompose $x \in \tilde{U}_j^S$ as $x = \sqrt{1-\beta^2}u + \beta v$ for unit vectors $u \in U$ and $v \perp U$ and some $\beta \ge 0$. Then $\beta = d(x,U) \le \sin\theta(\tilde{U}_j^S, U)$, and we conclude

$$\|t_i\tilde{V}_j^S\| = x^{\mathrm{T}} t_i = \sqrt{1-\beta^2}u^{\mathrm{T}} t_i + \beta v^{\mathrm{T}} t_i = 0 + \beta v^{\mathrm{T}} t_i \le \|t_i\| \cdot \sin\theta(\tilde{U}_j^S, U).$$

■

Corollary 5.6.4 *For every point p_i and iteration j, $\|t_i\tilde{V}_j^S\| \le O\left(\frac{1}{\varepsilon}\sigma^2 k^{1.5}\sqrt{d\log n}\right)$.*

Proof. Substitute into lemma 5.6.3 the bounds $\|t_i\| \leq O(\sigma\sqrt{d})$ from Eqn. (5.8) and from Eqn. (5.9),

$$\sin\theta(\tilde{U}_j^S, U) \leq O\left(\frac{1}{\varepsilon}\sigma k^{1.5}\sqrt{\log n}\right).$$

■

We note that according to our model parameters in Theorem 5.6.1, this implies that $\|t_i\tilde{V}_j^S\| \leq c\varepsilon$ for a small constant c that depends only on the choice of constant in our model, and we shall use this assumption henceforth.

5.6.5 Analysis: Projection of the data into the PCA space

We now show that the component of a data point $\tilde{p}_i$ inside the PCA space $\tilde{U}_j^S$ of some iteration j, typically recovers most of the "signal," i.e., the unperturbed version p_i. More precisely, we compare the length seen inside the PCA space $\|\tilde{p}_i\tilde{V}_j^S\|$ with the original length $\|p_i\|$. While the upper bound is immediate, the lower bound holds only *on average*.

Lemma 5.6.4 *W.h.p., for all* $\tilde{p}_i \in \tilde{P}_j$, $\|\tilde{p}_i\tilde{V}_j^S\|^2 - \|p_i\|^2 \leq d\sigma^2 + 0.0001\varepsilon^2$.

Proof. Using Pythagoras' Theorem, $\|\tilde{p}_i\tilde{V}_j^S\|^2 \leq \|\tilde{p}_i\|^2 = \|p_i\|^2 + \|t_i\|^2$, and the lemma follows by the noise bound (5.8). ■

Lemma 5.6.5 $\sum_{\tilde{p}_i\in\tilde{P}_j}(\|\tilde{p}_i\tilde{V}_j^S\|^2 - \|p_i\|^2) \geq -k\delta(n_j)^2$.

Proof. Let V be the projection matrix into U and P_j the non-noised version of $\tilde{P}_j$. Observe that $\tilde{P}_jV = P_j$, since the noise is orthogonal to U. Hence, by definition of PCA space (Theorem 5.3.1):

$$\sum_{p_i\in P_j}\|p_i\|^2 = \|P_j\|_F^2 = \|\tilde{P}_jV\|_F^2 \leq \sum_{l=1}^{k} s_l^2(\tilde{P}_j).$$

By Corollary 5.6.2, we further have that

$$\frac{n_j}{r}\sum_{l=1}^{k} s_l^2(\tilde{P}_j^S) \geq \sum_{l=1}^{k} s_l^2(\tilde{P}_j) - k\cdot\left(\frac{\delta(n_j)}{k}\right)^2 \geq \sum_i \|p_i\|^2 - k\cdot\left(\frac{\delta(n_j)}{k}\right)^2.$$

Or simply,

$$\frac{n_j}{r}\sum_{l=1}^{k} s_l^2(\tilde{P}_j^S) \geq \sum_i \|p_i\|^2 - k\cdot\left(\frac{\delta(n_j)}{k}\right)^2. \tag{5.10}$$

We also have:

$$\frac{n}{r}\sum_{l=1}^{m} s_l^2(\tilde{P}_j^s) = \frac{n}{r}\|(\tilde{V}_j^s)^T(\tilde{P}_j^s)^{\mathrm{T}}\tilde{P}_j^s\tilde{V}_j^s\|_F = \|(\tilde{V}_j^s)^T\frac{n}{r}(\tilde{P}_j^s)^{\mathrm{T}}\tilde{P}_j^s\tilde{V}_j^s\|_F$$
$$= \|(\tilde{V}_j^s)^T(\tilde{P}_j^{\mathrm{T}}\tilde{P}_j + Z)\tilde{V}_j^s\|_F \leq \sum_{\tilde{p}_i\in\tilde{P}_j}\|\tilde{p}_i\tilde{V}_j^s\|^2 + \|(\tilde{V}_j^s)^T Z\tilde{V}_j^s\|_F,$$

where Z has spectral norm at most $\delta(n_j)^2/k^2$, and hence $\|(\tilde{V}_j^s)^T Z\tilde{V}_j^s\|_F \leq \frac{k\delta(n_j)^2}{k^2} \leq \frac{\delta(n_j)^2}{k}$.

Rearranging yields us:

$$\frac{n}{r}\sum_{l=1}^{m} s_l^2(\tilde{P}_j^s) \leq \sum_{\tilde{p}_i\in\tilde{P}_j}\|\tilde{p}_i\tilde{V}_j^s\|^2 + \frac{\delta(n_j)^2}{k}.$$

Finally, we have:

$$\frac{n_j}{r}\sum_{l=1}^{k} s_l^2(\tilde{P}_j^s) = \frac{n_j}{r}\sum_{j=1}^{m} s_j^2(\tilde{P}_j^s) + \frac{n_j}{r}\sum_{j=m+1}^{k} s_j^2(\tilde{P}_j^s)$$
$$\leq \sum_{p_i\in\tilde{P}_j}\|\tilde{p}_i\tilde{V}_j^s\|^2 + \frac{\delta(n_j)^2}{k} + \frac{n_j}{r}\sum_{j=m+1}^{k} s_j^2(\tilde{P}_j^s)$$
$$\leq \sum_{p_i\in\tilde{P}_j}\|\tilde{p}_i\tilde{V}_j^s\|^2 + \frac{\delta(n_j)^2}{k} + \frac{n_j}{r}(k-1)\delta(r)^2$$
$$\leq \sum_{p_i\in\tilde{P}_j}\|\tilde{p}_i\tilde{V}_j^s\|^2 + \frac{\delta(n_j)^2}{k} + (k-1)\delta(n_j)^2.$$

where we employed the threshold $s_j \leq \delta(r)$ for singular values taken by Algorithm 13 with $j > m$, and that $\frac{n_j}{r}\delta(r)^2 = \delta(n_j)^2$ by straightforward substitution of the formula $\delta(x) = \frac{c\varepsilon\sqrt{x}}{\sqrt{k}}$. The lemma follows by combining the above with Equation 5.10. ■

5.6.6 Analysis: Number of iterations

We now show that each iteration captures in M_j a good fraction of the remaining points, thereby bounding the number of iterations overall. In particular, we give a lower bound on the number of indexes i such that $\tilde{p}_i$ is close to the m dimensional PCA subspace $\tilde{U}_j^s$, using results from Section 5.6.5. Note that the square of this distance for a point $\tilde{p}_i$ is precisely $\|\tilde{p}_i\|^2 - \|\tilde{p}_i\tilde{V}\|^2$. Let X and Y be quantities according to lemmas 5.6.4 and 5.6.5, such that

$$\|\tilde{p}_i\tilde{V}_j^s\|^2 - \|p_i\|^2 \leq Y. \tag{5.11}$$

$$Xn_j \leq \sum_i (\|\tilde{p}_i\tilde{V}_j^s\|^2 - \|p_i\|^2). \tag{5.12}$$

Now let f be the fraction of i's such that $\|\tilde{p}_i\tilde{V}_j^s\|^2 - \|p_i\|^2 \leq -0.0002\varepsilon^2$. Then

$$Xn_j \leq \sum_i \|\tilde{p}_i\tilde{V}_j^s\|^2 - \sum_i \|p_i\|^2 \leq (1-f)n_jY - 0.0002fn_j\varepsilon^2. \tag{5.13}$$

Rearrangement of terms gives us that $f \leq \frac{Y-X}{Y+0.0002\varepsilon^2}$. By lemma 5.6.5 , we can set $Xn_j = -k\delta^2 = -c^2\varepsilon^2 n_j = -0.00001\varepsilon^2 n_j$ and so $X \leq -0.00001\varepsilon^2$. And by lemma 5.6.4, we have $Y \leq d\sigma^2 + 0.0001\varepsilon^2$. Elementary calculations now yield

$$f \leq 1 - \Omega\left(\frac{\varepsilon^2}{d\sigma^2}\right) \leq 1 - \Omega\left(\sqrt{\frac{\log n}{d}}\right).$$

(This last upper bound on f can be made tighter as dependence on σ, but we opt for the looser bound which holds in our range of parameters for simplicity.) Now for the rest of the $(1-f) \geq \Omega\left(\sqrt{\frac{\log n}{d}}\right)$ fraction of the points, the distance to the PCA subspace $\tilde{U}$ is, by Pythagoras' Theorem,

$$\|\tilde{p}\|^2 - \|\tilde{p}\tilde{V}_j^s\|^2 = \|p\|^2 + \|t\|^2 - \|\tilde{p}\tilde{V}_j^s\|^2 \leq \|t\|^2 + 0.0002\varepsilon^2.$$

Since $\|t\|^2 \leq d\sigma^2 + 0.0001\varepsilon^2$, we get the required inequality that a large fraction of the points is within squared distance $d\sigma^2 + 0.001\varepsilon^2 = \Psi$. It follows that the fraction of points captured by $\tilde{U}_j^s$, i.e., in a single iteration, is at least $\Omega\left(\sqrt{\frac{\log n}{d}}\right)$, which immediately implies a bound on the number of iterations, as follows.

Lemma 5.6.6 *Algorithm 13 terminates in at most* $O\left(\sqrt{\frac{d}{\log n}}\log n\right) = O(\sqrt{d\log n})$ *iterations.*

5.6.7 Analysis: Correctness

It now remains to show that the data structure that captures the actual nearest neighbor $\tilde{p}^*$ will still report $\tilde{p}^*$ as the nearest neighbor to $\tilde{q}$ in the k-dimensional data structure.

Suppose $\tilde{p}^*$ has been captured in j^{th} iteration, i.e., $\tilde{p}^* \in M_j$. For simplicity of exposition, let $\tilde{U} = \tilde{U}^s_j$ and $\tilde{V} = \tilde{V}^s_j$.

Note that all distance computations for a query $\tilde{q}$ are of the form $\|\tilde{q}\tilde{V} - \tilde{p}\tilde{V}\| = \|(\tilde{q} - \tilde{p})\tilde{V}\|$, where $\tilde{p}$ is a point that is close to $\tilde{U}$. Let $\tilde{q} = q + t_q$ and $\tilde{p} = p + t$. Then we have for a point $\tilde{p}$:

$$\|(\tilde{q} - \tilde{p})\tilde{V}\| = \|(q-p)\tilde{V}\| \pm O\left(\|t_q\tilde{V}\| + \|t\tilde{V}\|\right).$$

Considering the noise of the query point q, by Corollary 5.2.1, we have $\|t_q\tilde{V}\| \leq O(\sigma\sqrt{k} + \sigma\sqrt{\log n})$ w.h.p. Similarly, considering the noise t of a point, by Corollary 5.6.4, we have $\|t\tilde{V}\| \leq O(\frac{\sigma^2}{\varepsilon}k^{1.5}\sqrt{d\log n})$. By the model specified in Theorem 5.6.1, we can set both these terms to be smaller than 0.01ε. Hence we have:

$$\|(\tilde{q} - \tilde{p})\tilde{V}\| = \|(q-p)\tilde{V}\| \pm 0.02\varepsilon. \tag{5.14}$$

Furthermore, we have $\sin\theta(\tilde{U}, U) \leq 0.01\varepsilon$. We now decompose U into two components. Let U_{in} be the projection of $\tilde{U}$ onto U, and U_{out} be the space orthogonal to U_{in} but lying in U. See Figure 5.3. We note that U_{out} is also orthogonal to $\tilde{U}$: otherwise some component of U_{out} would lie in the projection of $\tilde{U}$ onto U, which is a contradiction. Let V_{in} and V_{out} be the corresponding projection matrices. We likewise decompose each point p as $p_{in} \in U_{in}$ and $p_{out} \in U_{out}$.

We make the following claim, which shows that bounding $\|p_{out}\|$ for all points in M_j suffices for correctness of our algorithm.

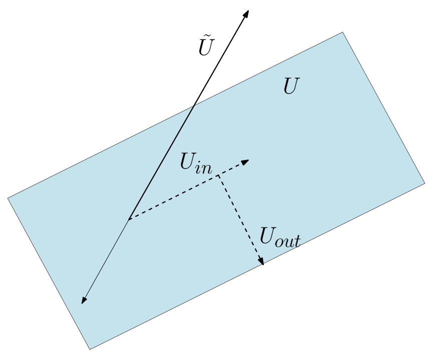


Figure 5.3. U_{in} is projection of $\tilde{U}$ into U, and U_{out} is the orthogonal complement of U_{in} in U.

Lemma 5.6.7 *If $\|p_{out}\| \leq 0.1\varepsilon$ for all p captured by subspace $\tilde{U}$ (also capturing p^*), then $\tilde{p}^*$ remains a nearest neighbor to $\tilde{q}$ after projection to $\tilde{U}$. Also for any $p' \neq p^*$, we have that $\|(\tilde{q}-\tilde{p}')\tilde{V}\| \geq \left(1+\frac{\varepsilon}{8}\right)\|(\tilde{q}-\tilde{p}^*)\tilde{V}\|$.*

Proof. Consider Equation 5.14.

$$\begin{aligned}
\|(\tilde{q}-\tilde{p})\tilde{V}\| &= \|(q-p)\tilde{V}\| \pm 0.02\varepsilon \\
&= \|(q_{in}-p_{in}+q_{out}-p_{out})\tilde{V}\| \pm 0.02\varepsilon \\
&\quad \text{(Decomposing } q,\, p \text{ into components in } U_{in} \text{ and } U_{out}) \\
&= \|(q_{in}-p_{in})\tilde{V}+(q_{out}-p_{out})\tilde{V}\| \pm 0.02\varepsilon \\
&= \|(q_{in}-p_{in})\tilde{V}\| \pm 0.02\varepsilon \quad (\text{Since } U_{out} \perp \tilde{U}) \\
&= (1 \pm 0.01\varepsilon)\|q_{in}-p_{in}\| \pm 0.02\varepsilon \quad (\text{Since } \sin\theta(U_{in},\tilde{U}) \leq 0.01\varepsilon).
\end{aligned}$$

To summarize these last calculations, we have:

$$\|(\tilde{q}-\tilde{p})\tilde{V}\| = (1 \pm 0.01\varepsilon)\|q_{in}-p_{in}\| \pm 0.02\varepsilon. \tag{5.15}$$

Next note by Pythagoras:

$$\|q_{in}-p_{in}\|^2 = \|q-p_{in}\|^2 - \|q_{out}\|^2. \tag{5.16}$$

Also observe from the triangle inequality and the assumption of our lemma that $\|p_{out}\| \leq 0.1\varepsilon$, $\forall p \in P$ captured by the subspace, $\|q-p_{in}\| = \|q-p\| \pm \|p_{out}\| = \|q-p\| \pm 0.1\varepsilon$.

Hence, if p^* is captured by the data structure, $\|q-p^*_{in}\|^2 = (\|q-p^*\| \pm 0.1\varepsilon)^2 \leq 1+0.25\varepsilon$. Similarly for $p' \neq p^*$, we have $\|q-p'_{in}\|^2 \geq (1+\varepsilon-0.1\varepsilon)^2 \geq 1+1.8\varepsilon$. This gives:

$$\frac{1+1.8\varepsilon}{1+0.25\varepsilon} \leq \frac{\|q-p'_{in}\|^2}{\|q-p^*_{in}\|^2} \leq \frac{\|q-p'_{in}\|^2-\|q_{out}\|^2}{\|q-p^*_{in}\|^2-\|q_{out}\|^2} = \frac{\|q_{in}-p'_{in}\|^2}{\|q_{in}-p^*_{in}\|^2},$$

where we crucially used in the second step the fact that subtracting the same quantity from both numerator and denominator of a fraction can only increase it. Some elementary algebraic manipulation shows then:

$$\frac{\|q_{in} - p'_{in}\|}{\|q_{in} - p^*_{in}\|} \geq 1 + \frac{\varepsilon}{4}. \tag{5.17}$$

Now we lower bound $\|q_{in} - p'_{in}\|$. We do so as follows:

$$\begin{aligned} \|q_{in} - p'_{in}\|^2 - \|q_{in} - p^*_{in}\|^2 =& \|q - p'_{in}\|^2 - \|q - p^*_{in}\|^2 && \text{(By Equation 5.16)} \\ \geq& (1+0.9\varepsilon)^2 - (1+0.1\varepsilon)^2 \geq 1.6\varepsilon. \end{aligned}$$

This implies:

$$\|q_{in} - p'_{in}\| \geq 1.2\varepsilon. \tag{5.18}$$

Finally, we come to our main claim of the ratio of $\|(\tilde{q} - \tilde{p}')\tilde{V}\|$ to $\|(\tilde{q} - \tilde{p}^*)\tilde{V}\|$. By Equation 5.15, this is at least:

$$\frac{\|(\tilde{q} - \tilde{p}')\tilde{V}\|}{\|(\tilde{q} - \tilde{p}^*)\tilde{V}\|} \geq \frac{(1-0.01\varepsilon)\|q_{in} - p'_{in}\| - 0.02\varepsilon}{(1+0.01\varepsilon)\|q_{in} - p^*_{in}\| + 0.02\varepsilon}.$$

Substituting the lower bound on $\frac{\|q_{in} - p'_{in}\|}{\|q_{in} - p^*_{in}\|}$ from Equation 5.17, and the lower bound on $\|q_{in} - p'_{in}\|$ from Equation 5.18 completes our claim. ■

Next we derive a sufficient condition for $\|p_{out}\|$ to be bounded as desired. For a vector a, we define $a\tilde{V}$ and $a^{\perp\tilde{U}}$ to be the components lying in and orthogonal to subspace $\tilde{U}$, respectively.

The quantity of interest is c_p, defined as the cosine of the angle between $(p_{in})^{\perp\tilde{U}}$ and $t^{\perp\tilde{U}}$, for a point $\tilde{p} = p_{in} + p_{out} + t$.

Lemma 5.6.8 *Decompose p as $p = p_{in} + p_{out} + t$, where $t \perp U$. Suppose $c_p \leq C\frac{\varepsilon}{\sigma\sqrt{d}}$ for suitable choice of constant C. Then $\|p_{out}\| \leq 0.1\varepsilon$ w.h.p.*

Proof. We show first that to upper-bound p_{out}, it suffices to lower bound $\|\left(p_{in} + t\right)^{\perp\tilde{U}}\|^2$ by $d\sigma^2 - 0.009\varepsilon^2$. Indeed, for captured points, we have by construction:

$$\|\tilde{p}\|^2 - \|\tilde{p}\tilde{V}\|^2 \leq \Psi, \tag{5.19}$$

where $\Psi = d\sigma^2 + 0.001\varepsilon^2$.

We convert the inequality to our desired condition as follows:

$$\|\tilde{p}\|^2 - \|\tilde{p}\tilde{V}\|^2 \leq d\sigma^2 + 0.001\varepsilon^2$$
$$\|p_{out}\|^2 + \|p_{in}+t\|^2 - \|\tilde{p}\tilde{V}\|^2 \leq d\sigma^2 + 0.001\varepsilon^2$$
(Since by Pythagoras, $\|\tilde{p}\|^2 = \|p_{out}\|^2 + \|p_{in}+t\|^2$.)
$$\|p_{out}\|^2 + \|(p_{in}+t)\tilde{V}\|^2 + \|(p_{in}+t)^{\perp\tilde{U}}\|^2 - \|\tilde{p}\tilde{V}\|^2 \leq d\sigma^2 + 0.001\varepsilon^2$$
(Decomposing $p_{in}+t$ orthogonal to and lying in subspace $\tilde{U}$.)
$$\|p_{out}\|^2 + \|\tilde{p}\tilde{V}\|^2 + \|(p_{in}+t)^{\perp\tilde{U}}\|^2 - \|\tilde{p}\tilde{V}\|^2 \leq d\sigma^2 + 0.001\varepsilon^2$$
(Since $p_{out} \perp \tilde{V}$ by construction, hence $\tilde{p}\tilde{V} = (p_{in}+t)\tilde{V}$.)
$$\|p_{out}\|^2 + \|(p_{in}+t)^{\perp\tilde{U}}\|^2 \leq d\sigma^2 + 0.001\varepsilon^2$$
$$\|p_{out}\|^2 \leq 0.001\varepsilon^2 + d\sigma^2 - \|(p_{in}+t)^{\perp\tilde{U}}\|^2.$$

Clearly now if $\|\left(p_{in}+t\right)^{\perp\tilde{U}}\|^2 \geq d\sigma^2 - 0.009\varepsilon^2$, then $\|p_{out}\|^2 \leq 0.01\varepsilon^2$ and would complete our proof.

We now show how the bound on c_p implies the required lower bound on $\|\left(p_{in}+t\right)^{\perp\tilde{U}}\|^2$. First note by the law of cosines, that

$$\|(p_{in}+t)^{\perp\tilde{U}}\|^2 = \|p_{in}^{\perp\tilde{U}}\|^2 + \|t^{\perp\tilde{U}}\|^2 - 2\|p_{in}^{\perp\tilde{U}}\|\|t^{\perp\tilde{U}}\|c_p. \tag{5.20}$$

Next note that $\|t^{\perp\tilde{U}}\|^2 = d\sigma^2 \pm 0.001\varepsilon^2$ w.h.p. and a suitably small constant c in our bound on σ in the model parameters. This follows from decomposing $\|t\|^2 = \|t^{\perp\tilde{U}}\|^2 + \|t\tilde{V}\|^2$ by Pythagoras, the concentration on $\|t\|^2$ by Equation 5.8 and the upper bound on $\|t\tilde{V}\|^2$ by Corollary 5.6.4. We now solve to find the desired condition on c_p.

$$\|p_{in}^{\perp\tilde{U}}\|^2 + \|t^{\perp\tilde{U}}\|^2 - 2\|p_{in}^{\perp\tilde{U}}\|\|t^{\perp\tilde{U}}\|c_p \geq d\sigma^2 - 0.009\varepsilon^2$$
$$\|p_{in}^{\perp\tilde{U}}\|^2 + (d\sigma^2 \pm 0.001\varepsilon^2) - 2\|p_{in}^{\perp\tilde{U}}\|\|t^{\perp\tilde{U}}\|c_p \geq d\sigma^2 - 0.009\varepsilon^2$$
$$\|p_{in}^{\perp\tilde{U}}\|^2 - 2\|p_{in}^{\perp\tilde{U}}\|\|t^{\perp\tilde{U}}\|c_p \geq -0.008\varepsilon^2$$
$$2\|p_{in}^{\perp\tilde{U}}\|\|t^{\perp\tilde{U}}\|c_p \leq \|p_{in}^{\perp\tilde{U}}\|^2 + 0.008\varepsilon^2$$
$$c_p \leq \frac{1}{2\|t^{\perp\tilde{U}}\|}\left(\|p_{in}^{\perp\tilde{U}}\| + 0.008\left(\frac{\varepsilon^2}{\|p_{in}^{\perp\tilde{U}}\|}\right)\right).$$

Noting that the minimum of $x+\frac{\alpha}{x}$ is $\sqrt{\alpha}$ for any fixed α and that $\|t^{\perp\tilde{U}}\|$ is $O(\sqrt{d}\sigma)$ we obtain that $c_p \le C\frac{\varepsilon}{\sqrt{d}\sigma}$ is a sufficient constraint for suitable choice of constant C. ■

The final component is to prove that c_p is indeed small. Since the vector t is random, this is generally not an issue: $t^{\tilde{U}}$ and $(p_{in})^{\tilde{U}}$ will be independent for all points $\tilde{p} \in \tilde{P}_j \setminus \tilde{P}_j^s$ (but not for the sampled points – this is the reason they are never included in M_j). However, in subsequent iterations, this may introduce some conditioning on t_i, so we need to be careful and argue about all iteration "at once." In fact, we show a result slightly stronger than that required by lemma 5.6.8 (within the parameters of our model):

Lemma 5.6.9 *Fix a "horizon" j, and a point $\tilde{p} = p+t$ that has not been sampled into any set $\tilde{P}_l^s$ for all iterations $l \le j$ ($\tilde{p}$ may or may not have been captured at some iteration $l \le j$). Then $c_p \le O\left(\frac{\sqrt{\log n}}{\sqrt{d}}\right)$ at iteration j, with high probability.*

Proof. The assumption on not having been sampled means that we can think of running the algorithm, disallowing $\tilde{p}$ to be in the sample. In this situation, we can run the entire algorithm, up to iteration j, without $\tilde{p}$ — it does not affect the rest of the points in any way. Hence, the algorithm can sample sets $\tilde{P}_0^s, \dots \tilde{P}_j^s$, and fix spaces $\tilde{U}_0^s, \dots \tilde{U}_j^s$, without access to $\tilde{p}$. Furthermore, for each $l = 0 \dots j$, the vector $p_{in}^{\perp\tilde{U}_l^s}$ is some vector, *independent* of the noise t. Hence we can "reveal" the vector t, after having fixed all vectors $p_{in}^{\perp\tilde{U}_l^s}$, for $l = 0 \dots j$. The vector t will have angle $O(\frac{\sqrt{\log n}}{\sqrt{d}})$ with all of them, with high probability. Note that, at this moment, it does not actually matter whether $\tilde{p}$ was captured early on or not. (Again, t is admittedly conditioned via bounds on $\|T\|$ and $\|t\|$, but since these are "whp" events, they do not affect the conclusion.) ■

We now remark on the resulting parameters of the algorithm. Processing an iteration of the preprocessing stage takes $O(rd^2+d^3+ndk) = O(nd^2)$ time for: computing $\tilde{P}_j^s$, the PCA space, and M_j, respectively. Hence, over $O(\sqrt{d}\log n)$ iterations, together with preprocessing of the k-dimensional NNS data structures, we get preprocessing time $O((nd^2+d^3+F_{\text{prep}})\sqrt{d}\log n)$.

Space requirement is essentially that of $O(\sqrt{d}\log n)$ instances of k-dimensional NNS data structure, plus the space to store $O(\sqrt{d}\log n)$ spaces $\tilde{U}_j^s$, and the left-over set R. The query time is composed of: computing the projections into $O(\sqrt{d}\log n)$ subspaces, querying the k-dimensional NNS data structures, and computing the distances to left-over points in R. Overall, this comes out to $O(dk \cdot \sqrt{d \log n} + \sqrt{d \log n} \cdot F_{\text{query}} + d|R|)$.

5.7 PCA tree

We now present our second spectral algorithm, which is closely related to the PCA tree (143, 152). We first give the algorithm and then present its analysis. Overall, we prove the following theorem.

Theorem 5.7.1 *Consider the Gaussian-noise model* (5.1)-(5.3)*, and assume its parameters satisfy* $\sigma < \kappa \cdot \min\left\{ \frac{\varepsilon}{\sqrt{k \log n}}, \frac{\varepsilon}{\sqrt{k}\sqrt[4]{d \log n}} \right\}$*, for sufficiently small constant* $\kappa > 0$*. There exists a data structure that preprocesses* $\tilde{P}$*, and then given the query* $\tilde{q}$*, returns the nearest neighbor* $\tilde{p}^*$ *w.h.p.*[5] *And w.h.p. the query time is* $(k/\varepsilon)^{O(k)} \cdot d^2$*, the space requirement is* $O(nd)$*, and the preprocessing time is* $O(n^2 d + nd^3)$*.*

The algorithm itself is deterministic.

5.7.1 Algorithm description

The algorithm constructs one-space partitioning tree hierarchically, where each tree node is associated with a subset of the pointset $\tilde{P}$. We start with the root of the tree, associated with all n points $\tilde{P}$. Now at each tree node x, we take the pointset associated with x, termed $\tilde{P}_x^{in}$. First, we perform a process called "de-clumping," which just discards part of the dataset, to obtain a set $\tilde{P}_x \subseteq \tilde{P}_x^{in}$. We describe this process at the end.

The main operation at a node is to take the top centered-PCA direction of $\tilde{P}_x$, termed v_x. By centered-PCA we mean subtracting from each vector in $\tilde{P}_x$ their average $a = \frac{1}{|\tilde{P}_x|} \sum_{\tilde{p} \in \tilde{P}_x} \tilde{p}$, and then taking the top PCA direction. Now, let $\theta \triangleq \frac{\varepsilon}{1000 k^{3/2}}$ and let Θ be the partition of the real line into segments of length θ, namely $\Theta = \{[\theta i, \theta(i+1)) \mid i \in \mathbb{Z}\}$. Then we partition $\tilde{P}_x$ into parts depending on which segment from Θ the projection of a point $\tilde{p} \in \tilde{P}_x$ onto v_x falls into. Then, we orthogonalize with respect to v_x, namely, transform each point $\tilde{p} \in \tilde{P}_x$ into $\tilde{p}' = \tilde{p} - \langle \tilde{p}, v_x \rangle v_x$. For each nonempty segment of Θ we produce a child of x associated with the points that fall into that segment, and repeat recursively on it. We stop once the current tree node has at most d points associated with

[5] The probability is over the randomness from the model.

it.

During a query, we follow the tree into all the buckets (slabs) that intersect a ball of radius $1+\varepsilon/2$ around $\tilde{q}$. In each leaf, compute the exact distance from q to all points associated to that leaf. Finally, we report the closest point found.

We now describe the de-clumping procedure that is done at each node. We compute the top centered-singular value of $\tilde{P}_x^{in}$. If this value is at least $\lambda_C = \lambda_C(|\tilde{P}_x^{in}|) \triangleq \frac{\varepsilon}{16}\sqrt{|\tilde{P}_x^{in}|/k}$, then set $\tilde{P}_x \triangleq \tilde{P}_x^{in}$. Otherwise, find the closest pair of points in $\tilde{P}_x^{in}$, and let δ denote their squared-distance. Remove all the pairs of points in $\tilde{P}_x^{in}$ that have squared-distance at most $\delta + \varepsilon^2/2$, to obtain $\tilde{P}_x$. (The removal is iterative, proceeding in arbitrary order.)

5.7.2 Analysis: Tree depth

The key to the analysis is to show that our PCA tree has depth at most $2k$. The rest of the analysis will follow as we show in later sections.

In the analysis, we use the centered-PCA directions. For this purpose, we first define the centering operation $c(A)$ for a set/matrix of points: $c(A) \triangleq A - \frac{1}{|A|}\sum_{p\in A} p$. Then the centered singular value, denoted $\|A\|_c$, is $\|c(A)\|$. Note that the norm still satisfies the triangle inequality.

Lemma 5.7.1 (Tree Depth) *The constructed PCA tree has depth at most $2k$.*

Proof. We first analyze the effect of orthogonalization on the points $\tilde{p}$. Fix some node x at a level $1 \le l \le 2k$, and some point $\tilde{p} = p + t_p$ reaching it. Call $\tilde{p}^x \in \tilde{P}_x^{in}$ as its version at node x, after the anterior orthogonalizations at the ancestor nodes. Also, define $n_x \triangleq |\tilde{P}_x|$.

We view each step of orthogonalization as two displacement processes. If we have orthogonalized the point with respect to some vector v, this is equivalent to snapping (projecting) the point $\tilde{p}^x$ to the hyperplane defined by $\{z \in \mathbb{R}^d \mid zv = \theta \cdot \lfloor \frac{\tilde{p}^x \cdot v}{\theta} \rfloor\}$, and then moving the hyperplane towards the origin. Most importantly, all points from node x going to the same child will be all snapped to the same hyperplane. The snapping to the hyperplane moves the point $\tilde{p}^x$ by a vector of norm at most θ. Note that, while the algorithm also moves the hyperplane to pass through the origin, this does not change the relative distances of the points in this hyperplane.

Thus we can write each point $\tilde{p}^x$ reaching node x as $\tilde{p}^x = \tilde{p} + m^x + m_p^x$, where m^x is the sum of all the hyperplane moves (and is dependent on the node x only), and m_p^x which is the sum of the "snapping moves" and depends on the actual point. We observe that m_p^x has small norm, and, in particular $\|m_p^x\|^2 \leq l \cdot \theta^2 \leq 2k\theta^2$, since each move is in an orthogonal direction with respect to the previous moves.

Below we assume that lemma 5.3.1 holds. Also, for any two points p_1, p_2, the norm of difference of the noises is $\|t_{p_1} - t_{p_2}\|^2 = 2\sigma^2 d \pm 0.1\varepsilon^2$ according to Corollary 5.2.1 for $\sigma \ll \varepsilon / \sqrt[4]{d \log n}$.

The main part of the proof is to prove that the top PCA direction v_x at each node x is close to U. We prove this by induction over levels.

Claim 5.7.1 (Induction hypothesis) *Consider some node x at level l, which contains at least $d = \Omega(\log n)$ points. Let v_x be the top centered-PCA direction of $\tilde{P}_x$. The projection of v_x onto $U^\perp$ is at most $\gamma = O(\sigma\sqrt{\log n} \cdot \sqrt{k}/\varepsilon)$.*

Before proving the induction hypothesis, we need to show an additional claim, which characterizes the result of de-clumping in the current node x: essentially that the top PCA direction is heavy in the space U. For the claim below, we assume that Claim 5.7.1 is true at all levels *above* the current node x.

For a node x, we define helper sets/matrices P_x, M_x, T_x as follows. First, consider points $\tilde{P}_x$, take their non-noised versions living in U (as in the model), and move using the vector m^x; this gives the set P_x. Also, let T_x be the noise vectors of $\tilde{P}_x$. Define matrix M_x as being composed of movements m_p^x for all points p in $\tilde{P}_x$. Note that $\tilde{P}_x = P_x + M_x + T_x$, and that $\|T_x\|_c \leq \|T_x\| \leq \eta(n_x) \leq \lambda_C(n_x)$, where $\eta(n_x)$ is the function from lemma 5.3.1.

Claim 5.7.2 *Suppose we performed the de-clumping step on $\tilde{P}_x^{in}$, to obtain $\tilde{P}_x$. For v_x the top centered-PCA direction of $\tilde{P}_x$, we have that $\|c(P_x)v_x\| \geq \lambda_C - \eta(n_x)$.*

Proof. Suppose the top singular value of $c(\tilde{P}_x^{in})$ is at least λ_C (in which case no de-clumping is done and $\tilde{P}_x = \tilde{P}_x^{in}$). Hence, considering $P_x = \tilde{P}_x - (\tilde{P}_x - P_x)$, which also implies $c(P_x) = c(\tilde{P}_x) - c(\tilde{P}_x - P_x)$, we have

$$\|c(P_x)v_x\| \geq \|c(\tilde{P}_x)v_x\| - \|c(\tilde{P}_x - P_x)v_x\| \geq \lambda_C - \eta(|\tilde{P}_x|),$$

since $\|c(\tilde{P}_x - P_x)v_x\| = \|c(T_x)v_x\| \leq \eta(|\tilde{P}_x|)$ by lemma 5.3.1, and the fact that $M_x v_x = 0$.

Otherwise, the algorithm throws out some points from $\tilde{P}_x^{in}$. Define P_x^{in} similarly to P_x: take original (nonmoved, no noise) versions of the points from $\tilde{P}_x^{in}$, plus the overall movement m^x. In this situation, there must be two points $p_1, p_2 \in P_x^{in}$ such that $\|p_1 - p_2\| \leq \varepsilon/4$: otherwise, the top singular value of $\tilde{P}_x^{in} = P_x^{in} - (P_x^{in} - \tilde{P}_x^{in})$ would be at least $\|P_x^{in}\|_c - \eta(|\tilde{P}_x^{in}|) \geq \frac{\|p_1 - p_2\|}{2} \cdot \sqrt{|\tilde{P}_x^{in}|/k} - \eta(|\tilde{P}_x^{in}|) \geq \lambda_c$, a contradiction.

We now want to characterize the minimal distance δ in $\tilde{P}_x^{in}$. Note that the projection of $m_{p_1}^x$ and $m_{p_2}^x$ into $U^\perp$ is at most $2k\theta\gamma$, since each of the basis vectors of m_p^x has projection into $U^\perp$ at most γ. Hence, the square of the component of $\tilde{p}_1 - \tilde{p}_2$ in $U^\perp$ is equal to:

$$\begin{aligned}(\|t_{p_1} - t_{p_2}\| \pm 2k\theta\gamma)^2 &= 2\sigma^2 d \pm 0.1\varepsilon^2 \pm (2k\theta\gamma)^2 \pm 5\sigma\sqrt{d} \cdot 2k\theta\gamma \\ &= 2\sigma^2 d \pm 0.1\varepsilon^2 \pm 0.01\varepsilon^2 \pm 0.01\varepsilon^2,\end{aligned}$$

for $\sigma \ll \varepsilon/\sqrt{\log n}$ and $\sigma \ll \frac{\varepsilon}{\sqrt[4]{d \log n}}$. Thus, for $\tilde{p}_1 = p_1 + t_1$ and $\tilde{p}_2 = p_2 + t_2$:

$$\delta = \|\tilde{p}_1 - \tilde{p}_2\|^2 \geq (\|t_1 - t_2\| - 2k\theta\gamma)^2 \geq 2\sigma^2 d - 0.12\varepsilon^2. \tag{5.21}$$

After the de-clumping, the distance between any two distinct points $p', p'' \in P_x$, with noise vectors t', t'', respectively, must satisfy:

$$(\|p' - p''\| + 2\sqrt{2k}\theta)^2 \geq \delta + \varepsilon^2/2 - (\|t' - t''\| + 2k\theta\gamma)^2 \geq \varepsilon^2/2 - 2 \cdot 0.12\varepsilon^2 \geq \varepsilon^2/4.$$

Hence $\|p' - p''\| \geq \varepsilon/2 - 0.01\varepsilon > \varepsilon/4$, which means that $\|\tilde{P}_x\|_c \geq \lambda_c$ (as already argued above). Hence we can apply the same argument as above, this time for $\tilde{P}_x$ instead of $\tilde{P}_x^{in}$. ■

We now prove the induction hypothesis, namely Claim 5.7.1, for the current node x.

Proof.[Proof of Claim 5.7.1] Let $\tilde{P}_x$ be the points contained in x. By Claim 5.7.2, we have $\lambda_{PM} \triangleq \|P_x + M_x\|_c \geq \|c(P_x + M_x)v_x\| = \|c(P_x)v_x\| \geq \lambda_c - \eta(|\tilde{P}_x|) \geq \lambda_c/2$.

Decompose the top centered-PCA direction v_x of $\tilde{P}_x$ as $v_x = \sqrt{1-\alpha^2}u + \alpha u'$, where $u \in U$ and $u' \perp U$ are of norm 1. Note that α is exactly the projection of v_x onto $U^\perp$.

We will bound α by lower and upper bounding $\|\tilde{P}_x\|_c$ as a function of α and λ_{PM}. We start with the upper bound on $\|\tilde{P}_x\|_c$. For this we decompose the matrix $c(P_x + M_x)$ into the component in U, called $c(P_x + M_x^U)$, and the perpendicular one, called $c(M_x^\perp)$. Note that $c(P_x)$ lies inside U, despite the movement m_x, because of the centering $c(\cdot)$. We now bound the spectral norm of $\|c(M_x^\perp)\|$, using the inductive hypothesis that the projection of each of at most $2k$ basis vectors of m_p^x onto $U^\perp$ is at most γ:

$$\begin{aligned}\|c(M_x^\perp)\| &\leq \max_p \sqrt{n_x}\|m_p^x\| \cdot \sqrt{2k}\gamma \leq \gamma \cdot \sqrt{n_x} 2k \cdot \theta \\ &\leq \gamma \cdot \frac{\varepsilon}{500}\sqrt{n_x/k} \leq \frac{\gamma}{9} \cdot \frac{\varepsilon}{32}\sqrt{n_x/k} \leq \frac{\gamma}{9}\lambda_{PM}.\end{aligned}$$

Thus, we have that $\lambda_{PM^U} \triangleq \|P_x + M_x^U\|_c = \lambda_{PM} \pm \frac{\gamma}{9}\lambda_{PM}$.

We can now upper bound the norm of $\tilde{P}_x$:

$$\|\tilde{P}_x\|_c \leq \|c(P_x + M_x)v_x\| + \|c(T_x)v_x\| \leq \|c(P_x + M_x)v_x\| + \alpha\eta(n_x). \tag{5.22}$$

We need to bound the first term now. For this, we compute the following ratio, using that the projection of v_x into $U^\perp$ is of magnitude α:

$$\begin{aligned}
\frac{\|c(P_x + M_x)v_x\|^2}{\lambda^2_{PM^U}} &\leq \frac{\|c(P_x{+}M_x^U)v_x\|^2{+}\|c(M_x^\perp)v_x\|^2{+}2\|c(P_x{+}M_x^U)v_x\|\|c(M_x^\perp)v_x\|}{\lambda^2_{PM^U}} \\
&\leq \frac{(1{-}\alpha^2)\lambda^2_{PM^U}{+}\alpha^2{\cdot}(\gamma/9)^2{\cdot}\lambda^2_{PM}{+}2\alpha\sqrt{1{-}\alpha^2}{\cdot}\lambda_{PM^U}{\cdot}(\gamma/9)\lambda_{PM}}{\lambda^2_{PM^U}} \\
&\leq (1-\alpha^2) + \alpha^2(\gamma/9)^2(1+\gamma/9)^2 + 2\alpha\frac{\gamma}{9}\cdot(1+\gamma/9) \\
&\leq 1 - \alpha^2/2 + \alpha\gamma/3.
\end{aligned} \tag{5.23}$$

On the other hand, we want to prove a lower bound on $\|\tilde{P}_x\|_c$. We define u_{PM^U} to be the centered-PCA direction of $P_x + M_x^U$: $\lambda_{PM^U} = \|(P_x + M_x^U)u_{PM^U}\|_c$. Remember that u_{PM^U} lies inside U.

$$\|\tilde{P}_x\|_c \geq \|c(\tilde{P}_x)u_{PM^U}\| = \|c(P_x + M_x^U)u_{PM^U}\| = \lambda_{PM^U}. \tag{5.24}$$

Putting together the upper bound (5.22), (5.23) and the lower bound (5.24), we obtain

$$\lambda_{PM^U} \leq \lambda_{PM^U}\sqrt{1 - \alpha^2/2 + \alpha\gamma/3} + \alpha\eta(n_x)$$

and, using that $\sqrt{1-x} \leq 1 - x/2$ for all $0 \leq x \leq 1$, we conclude:

$$\alpha \leq 4 \cdot \left(\frac{\eta(n_x)}{\lambda_{PM^U}} + \gamma/6\right) \leq \gamma,$$

as long as $4\frac{\eta(n_x)}{\lambda_{PM^U}} < \gamma/3$. Since $\lambda_{PM^U} \geq \lambda_{PM}/2 \geq \lambda_c/4$, this is satisfied if

$$\gamma = \Theta(\sigma\sqrt{\log n}\sqrt{k}/\varepsilon).$$

We are done proving the inductive hypothesis. ■

The final step is to show that, because all vectors v_x along a root-to-leaf path are perpedicular to each other and are heavy in U, there cannot be too many of them, and hence the tree depth is bounded.

Claim 5.7.3 *Fix some dimension $k > 1$ subspace $U \subset \mathbb{R}^d$. Suppose there exist $2k+1$ vectors $v_1, \ldots v_{2k+1}$, such that the projection of each into U has norm at least $1 - 1/4k$. Then at least two of the vectors are not orthogonal to each other.*

Proof. For the sake of contradiction assume that all vectors $v_1 \ldots v_{2k+1}$ are mutually orthogonal. Then let u_i be the projection of v_i into U, and let $\delta_i = v_i - u_i$. We want to bound the dot product $u_i u_j$. Consider

$$0 = v_i^{\mathrm{T}} \cdot v_j = (u_i^{\mathrm{T}} + \delta_i^{\mathrm{T}})(u_j + \delta_j) = u_i^{\mathrm{T}} u_j + \delta_i^{\mathrm{T}} \delta_j.$$

Since $|\delta_i^{\mathrm{T}} \delta_i| \le 0.25/k$, we have that $|u_i^{\mathrm{T}} u_j| \le 0.25/k$. Even after normalization, we have that $\left| \frac{u_i^{\mathrm{T}} u_j}{\|u_i\| \cdot \|u_j\|} \right| \le 1/k$. Following Alon's result (10), we conclude that u_i's have rank at least $(2k+1)/2 > k$, which is impossible if all u_i live in the k-dimensional space U. Thus we have reached a contradiction. ■

We now combine the two claims together. Note that $\alpha \le 1/4k$. We conclude that the height of the tree is at most $2k$, thus concluding the proof of lemma 5.7.1. ■

5.7.3 Analysis: Correctness

Now that we have established an upper bound on the tree depth, we will show that the query algorithm will indeed return the nearest neighbor $\tilde{p}^*$ for the query $\tilde{q}$ (modelled as in Section 5.2). We show this in two steps: first we prove the result, assuming the point $\tilde{p}^*$ was not thrown out during de-clumping. Then we show that the de-clumping indeed does not throw out the point $\tilde{p}^*$.

Lemma 5.7.2 *The query algorithm returns the point $\tilde{p}^*$, assuming it was not thrown out during the de-clumping process.*

Proof. Consider some nonleaf tree node x containing $\tilde{p}^*$. We need to argue that, at node x, the query algorithm follows the child where $\tilde{p}^*$ goes.

As before, let $\tilde{p}^x$ be the orthogonalized version of $\tilde{p}^*$ (above x) and m_p^x is the total amount of "hyperplane snapping" that happened to $\tilde{p}^*$. We also have that v_x has projection at most γ onto $U^\perp$ (from Claim 5.7.1). Hence, we have:

$$|v_x \tilde{p}^x - v_x \tilde{q}| \le |v_x(t_{p^*} - t_q)| + |v_x(p^* - q)|,$$

again using that $v_x m_p^x = 0$. Note that, with high probability,

$$|v_x(t_{p^*} - t_q)| \leq 3\sigma\sqrt{d} \cdot \gamma \leq \varepsilon/8\sqrt{k}.$$

Since this is true also for all ancestors y of x, and since all v_y, together with v_x, are mutually orthogonal, we have that:

$$\begin{aligned}
&\sum_{y \text{ ancestor of } x} |v_y(\tilde{p}^x - \tilde{q})|^2 \\
&\quad = \sum_{y \text{ ancestor of } x} |v_y(t_{p^*} - t_q) + v_y(p^* - q)|^2 \\
&\quad \leq \left(\sqrt{\sum_{y \text{ ancestor of } x} \left| v_y(m_p^y + |t_{p^*} - t_q|) \right|^2} \right. \\
&\qquad\qquad \left. + \sqrt{\sum_{y \text{ ancestor of } x} |v_y(p^* - q)|^2} \right)^2 \\
&\quad \leq \left(\sqrt{2k \cdot \left(\varepsilon/8\sqrt{k}\right)^2} + \sqrt{\sum_{y \text{ ancestor of } x} |v_y(p^* - q)|^2} \right)^2 \\
&\quad \leq (\varepsilon/4\sqrt{2} + 1)^2 \\
&\quad < \varepsilon/2 + 1.
\end{aligned}$$

This means that the bucket (child node of x) of $\tilde{p}^x$ intersects a ball of radius $1 + \varepsilon/2$ around $\tilde{q}$, and hence the query algorithm will go into that bucket (child). ■

To complete the correctness argument, we also need to prove that p^* is never thrown out due to the de-clumping process.

Claim 5.7.4 *p^* is never thrown out due to the de-clumping process.*

Proof. For the sake of contradiction, suppose $\tilde{p}_x^*$ is thrown out at some node x. This means there is some point $\tilde{p}_x$ such that $\|\tilde{p}_x - \tilde{p}_x^*\|^2 \leq \delta + \varepsilon^2/2$. Since $\|q - p\| - \|q - p^*\| \geq \varepsilon$, we have that $\|p - p^*\| \geq \varepsilon$.

We have:

$$\begin{aligned}
\delta + \varepsilon^2/2 \geq \|\tilde{p}_x - \tilde{p}_x^*\|^2 &= \|p - p^* + (m_p^x - m_{p^*}^x) + (t_p - t_{p^*})\|^2 \\
&= \|(p - p^* + (m_p^x - m_{p^*}^x))U\|^2 + \|(m_p^x - m_{p^*}^x + t_p - t_{p^*})U^{\perp}\|^2.
\end{aligned}$$

We want to bound $\|(m_p^x - m_{p^*}^x + t_p - t_{p^*})U_\perp\|^2$. Indeed, note that the projection of m_p^x onto $U^\perp$ can be at most $2k\theta \cdot \gamma \leq O(\sigma\sqrt{\log n})$, by Claim 5.7.1. Hence:

$$\begin{aligned}\|(m_p^x - m_{p^*}^x + t_p - t_{p^*})U_\perp\| &\geq -O(\sigma\sqrt{\log n}) + \sqrt{2\sigma^2 d - 0.1\varepsilon^2}\\ &\geq -O(\sigma\sqrt{\log n}) + \sigma\sqrt{2d} - \frac{0.06\varepsilon^2}{\sigma\sqrt{2d}}\\ &\geq \sigma\sqrt{2d} - \frac{0.09\varepsilon^2}{\sigma\sqrt{2d}},\end{aligned}$$

as long as $O(\sigma\sqrt{\log n}) < \frac{0.03\varepsilon^2}{\sigma\sqrt{2d}}$, which is equavalent to saying that $\sigma \leq O\left(\frac{\varepsilon}{\sqrt[4]{d}\sqrt[4]{\log n}}\right)$.

Putting it all together, we have:

$$\begin{aligned}\delta + \varepsilon^2/2 \geq \|\tilde{p}_x - \tilde{p}_x^*\|^2 &\geq (\varepsilon - 2\cdot\sqrt{2k}\theta)^2 + (\sigma\sqrt{2d} - \frac{0.09\varepsilon^2}{\sigma\sqrt{2d}})^2\\ &\geq 0.8\varepsilon^2 + 2\sigma^2 d - 0.2\varepsilon^2\\ &> 2\sigma^2 d + 0.6\varepsilon^2.\end{aligned}$$

But, as proven in Eqn. (5.21), we have that $\delta \leq 2\sigma^2 d + 0.12\varepsilon^2$. We've reached a contradiction. ■

5.7.4 Analysis: Performance

The space and preprocessing bounds follow immediately from the construction. We just need to argue about the query time.

Claim 5.7.5 *The query time is* $(k/\varepsilon)^{O(k)}d^2$.

Proof. At each node of the tree, there are at most $O(1/\theta) = O(k^{3/2}/\varepsilon)$ child nodes that are followed. Hence, in total, we reach $O(1/\theta)^{2k} = (k/\varepsilon)^{O(k)}$ leaves. The factor of d^2 comes from the fact that each leaf has at most d points to check the distance against. ■

CHAPTER 6

CONCLUSION

6.1 Open questions and challenges

During the course of this dissertation, we encountered several questions as natural followups, and avenues for deeper investigation. We highlight a few specific ones that we feel can lead to productive research contributions in the future. In general, we feel the entire domain of non-Euclidean algorithmic geometry has barely been scratched upon, and that the next decade will see richer theory emerge in these fields.

- In Chapter 2, we show that with some structural assumptions, Euclidean algorithms for $(1+\varepsilon)$-ANN can be adapted for the low-dimensional case. Can Euclidean constructions also be adapted for approximate Minimum Enclosing Ball or high- dimensional ANN in the case of the Bregman divergences?

- One open question is to convert our lower bounds of Chapter 3 to be nonadaptive. Panigrahy, Talwar and Wieder (128) do give such a conversion for ℓ_2^2 under symmetric perturbations, but it is not immediately clear how to generalize their argument to asymmetric perturbation operators. A second intriguing direction we have already referred to is whether our analysis of asymmetric isoperimetry can open a different avenue of attack for lower bounds on Partial Match.

- Finally, the question remains as to whether directed hypercontractivity offers insights or generalizations for other expansion related problems previously considered on the undirected hypercube. In this regard, recent work by Chakrabarty and Seshadhri (42) and Khot *et al.*(92) on formulations of directed hypercontractivity for property testing represent a promising direction.

- We gave bounds for embedding certain classes of information divergences in Chapter 4. Can we show corresponding lower bounds for embedding more generic Bregman divergences in low dimensions? In particular, we feel the KL-divergence is a good starting point for proving the unfeasibility of such an embedding.

- In Chapter 5, we presented a model of data lying close to a low-dimensional subspace. Can our models be extended to data lying close to a low dimensional *manifold*? A popular spectral heuristic is converting Euclidean data to binary codewords for succinct representation, and fast search algorithms. Xinyang, Price and Constantine (158) give theoretical bounds and analysis for codes derived from random projections. In the spirit of our work, can more “data-aware” bounds be given for codes derived from PCA based heuristics such as those employed by Torralba and Weiss (155)?

APPENDIX

UPPER BOUND FOR BREGMAN ANN IN LOW DIMENSIONS

Lemma A.1 *Given any interval $I = [x_1 x_2]$ on the real line, there exists a finite μ such that $\sqrt{D_{s\phi}}$ is μ-defective with respect to I. We require all order derivatives of ϕ to be defined and bounded over the closure of I, and ϕ'' to be bounded away from zero.*

Proof. Consider three points $a, b, q \in I$.

Due to symmetry of the cases and conditions, there are three cases to consider: $a < q < b$, $a < b < q$ and $q < b < a$.

Case 1: Here $a < q < b$. The following is trivially true by the monotonicity of $\sqrt{D_{s\phi}}$.

$$\left|\sqrt{D_{s\phi}(q,a)} - \sqrt{D_{s\phi}(q,b)}\right| < \sqrt{D_{s\phi}(a,b)}. \tag{A.1}$$

Cases 2 and 3: For the remaining symmetric cases, $a < b < q$ and $q < b < a$, note that since $\sqrt{D_{s\phi}(q,a)} - \sqrt{D_{s\phi}(q,b)}$ and $\sqrt{D_{s\phi}(a,b)}$ are both bounded, continuous functions on a compact domain (the interval $[x_1 x_2]$), we need only show that the following limit exists:

$$\lim_{a \to b} \frac{\left|\sqrt{D_{s\phi}(q,a)} - \sqrt{D_{s\phi}(q,b)}\right|}{\sqrt{D_{s\phi}(a,b)}}. \tag{A.2}$$

First consider $a < b < q$, and we assume $\lim_{b \to a}$. For ease of computation, we replace ϕ' by ψ, to be restored at the last step. We will use the following Taylor expansions repeatedly in our derivation: $b = a + h$, $\psi(b) = \psi(a+h) = \psi(a) + h\psi'(a) + E(h^2)$, and $\sqrt{1+h} = 1 + h/2 + E(h^2)$. Here $E(h^x)$ denotes a tail of a Taylor expansion around a (or equivalently a Maclaurin expansion in h) where the lowest order term is h^x. Since we will be handling multiple Taylor expansions in what follows, we will use subscripts of the form E_1, E_2, etc. to distinguish the tails of different series.

$$\frac{\sqrt{D_{s\phi}(a,q)}-\sqrt{D_{s\phi}(b,q)}}{\sqrt{D_{s\phi}(a,b)}}=\frac{\left(\sqrt{(q-a)(\psi(q)-\psi(a))}-\sqrt{(q-b)(\psi(q)-\psi(b))}\right)}{\sqrt{(b-a)(\psi(b)-\psi(a))}}. \tag{A.3}$$

Computing the denominator, using the expansion that $\psi(b)=\psi(a+h)=\psi(a)+h\psi'(a)+E_1(h^2)$, we get:

$$\begin{aligned}
&\sqrt{(b-a)(\psi(b)-\psi(a))}\\
=&\sqrt{h(\psi(a+h)-\psi(a))}\\
=&\sqrt{h(\psi(a)+h\psi'(a)+E_1(h^2)-\psi(a))}\\
=&\sqrt{h(h\psi'(a)+E_1(h^2))}\\
=&\sqrt{h(h\psi'(a)+hE_2(h))}\\
=&\sqrt{h^2(\psi'(a)+E_2(h))}\\
=&h\sqrt{\psi'(a)+E_2(h)},
\end{aligned}$$

where in the third last step we set $hE_2(h)=E_1(h^2)$.

We now address the numerator, and begin by taking the same Taylor expansion.

$$\begin{aligned}
&\sqrt{(q-a)(\psi(q)-\psi(a))}-\sqrt{(q-b)(\psi(q)-\psi(b))}\\
=&\sqrt{(q-a)(\psi(q)-\psi(a))}-\sqrt{(q-a-h)(\psi(q)-\psi(a)-h\psi'(a)-E_1(h^2))}\\
=&\sqrt{(q-a)(\psi(q)-\psi(a))}\\
&\quad-\sqrt{(q-a)\left(1-\frac{h}{q-a}\right)(\psi(q)-\psi(a))\left(1-\frac{h\psi'(a)+E_1(h^2)}{\psi(q)-\psi(a)}\right)}\\
=&\sqrt{(\psi(q)-\psi(a))(q-a)}\left(1-\sqrt{1-\frac{h}{q-a}}\sqrt{1-\frac{h\psi'(a)+E_1(h^2)}{\psi(q)-\psi(a)}}\right).
\end{aligned}$$

We now take the McLaurin expansion of the square roots and note for the second such expansion we gain more higher order terms of h which we merge with E_1 to obtain a $E_3(h^2)$.

$$\begin{aligned}
&\sqrt{(q-a)(\psi(q)-\psi(a))}-\sqrt{(q-b)(\psi(q)-\psi(b))}\\
=&\sqrt{(\psi(q)-\psi(a))(q-a)}\\
&\quad\times\left(1-\left(1-\frac{h}{2(q-a)}+E_4(h^2)\right)\left(1-\frac{h\psi'(a)}{2(\psi(q)-\psi(a))}+E_3(h^2)\right)\right)\\
=&\sqrt{(\psi(q)-\psi(a))(q-a)}\left(\frac{h}{2(q-a)}+h\frac{\psi'(a)}{2(\psi(q)-\psi(a))}+E_5(h^2)\right)\\
=&h\sqrt{(\psi(q)-\psi(a))(q-a)}\left(\frac{1}{2(q-a)}+\frac{\psi'(a)}{2(\psi(q)-\psi(a))}+E_6(h)\right),
\end{aligned}$$

where the $E_5(h^2)$ is again obtained in the second last step from merging products involving one of $E_3(h^2)$ or $E_4(h^2)$, as well as of the two terms involving h with each other. And in the last step, we set $E_5(h^2) = hE_6(h)$.

Now combine numerator and denominator back in equation A.3 and observe a factor of h cancels out.

$$\begin{aligned}
&\frac{\sqrt{D_{s\phi}(a,q)}-\sqrt{D_{s\phi}(b,q)}}{\sqrt{D_{s\phi}(a,b)}}\\
=&\frac{h\sqrt{(\psi(q)-\psi(a))(q-a)}\left(\frac{1}{2(q-a)}+\frac{\psi'(a)}{2(\psi(q)-\psi(a))}+E_6(h)\right)}{h\sqrt{\psi'(a)+E_2(h)}}\\
=&\sqrt{\frac{(\psi(q)-\psi(a))(q-a)}{\psi'(a)+E_2(h)}}\left(\frac{1}{2(q-a)}+\frac{\psi'(a)}{2(\psi(q)-\psi(a))}+E_6(h)\right).
\end{aligned}$$

Note now that since ϕ is strictly convex, neither the numerator nor denominator of this expression approach 0 as $\lim_{h\to 0}$ (or equivalently, $\lim_{b\to a}$). So we can safely drop the higher order terms in the limit to obtain:

$$\begin{aligned}
&\lim_{h\to 0}\frac{\sqrt{D_{s\phi}(a,q)}-\sqrt{D_{s\phi}(b,q)}}{\sqrt{D_{s\phi}(a,b)}}\\
=&\sqrt{\frac{(\psi(q)-\psi(a))(q-a)}{\psi'(a)}}\left(\frac{1}{2(q-a)}+\frac{\psi'(a)}{2(\psi(q)-\psi(a))}\right)\\
=&\frac{1}{2}\sqrt{\frac{(\psi(q)-\psi(a))(q-a)}{\psi'(a)}}\left(\frac{1}{\sqrt{q-a}\sqrt{q-a}}+\frac{\sqrt{\psi'(a)}\sqrt{\psi'(a)}}{\sqrt{\psi(q)-\psi(a)}\sqrt{\psi(q)-\psi(a)}}\right)\\
=&\frac{1}{2}\left(\sqrt{\frac{\psi(q)-\psi(a)}{\psi'(a)(q-a)}}+\sqrt{\frac{\psi'(a)(q-a)}{\psi(q)-\psi(a)}}\right).
\end{aligned}$$

Substituting back $\phi'(x)$ for $\psi(x)$, we see that limit A.2 exists, provided ϕ is strictly convex:

$$\frac{1}{2}\left(\sqrt{\frac{\phi'(q)-\phi'(a)}{\phi''(a)(q-a)}}+\sqrt{\frac{\phi''(a)(q-a)}{\phi'(q)-\phi'(a)}}\right). \tag{A.4}$$

The analysis follows symmetrically for case 3, where $q < b < a$. ■

Lemma A.2 *Given any interval $I = [x_1 x_2]$ on the real line, there exists a finite μ such that $\sqrt{D_\phi}$ is right-sided μ-defective with respect to I.We require all order derivatives of ϕ to be defined and bounded over the closure of I, and ϕ'' to be bounded away from zero.*

Proof. Consider any three points $a, b, q \in I$. We will prove that there exists finite μ such that:

$$\left|\sqrt{D_\phi(a,q)}-\sqrt{D_\phi(b,q)}\right| < \mu\sqrt{D_\phi(b,a)}. \tag{A.5}$$

Here there are now six cases to consider: $a < q < b$, $b < q < a$, $a < b < q$, $b < a < q$, $q < b < a$, and $q < a < b$.

Cases 1 and 2: Here $a < q < b$. By monotonicity we have that:

$$\left|\sqrt{D_\phi(a,q)}-\sqrt{D_\phi(b,q)}\right| < \sqrt{D_\phi(a,b)}+\sqrt{D_\phi(b,a)}. \tag{A.6}$$

But by lemma 2.2.5, we have that $\sqrt{D_\phi(a,b)} < c\sqrt{D_\phi(b,a)}$ for some parameter c_0 defined over I. This implies that $\left|\sqrt{D_\phi(a,q)}-\sqrt{D_\phi(b,q)}\right| / \sqrt{D_\phi(b,a)} < c_0+1$, i.e, it is bounded over I. A similar analysis works for Case 2 where $b < q < a$.

Cases 3 and 4: For these two cases, $a < b < q$ and $b < a < q$, note that since $\sqrt{D_\phi(q,a)} - \sqrt{D_\phi(q,b)}$ and $\sqrt{D_\phi(b,a)}$ are both bounded, continuous functions on a compact domain (the interval $[x_1 x_2]$), we need only show that the following limit exists:

$$\lim_{a\to b}\frac{\left|\sqrt{D_\phi(a,q)}-\sqrt{D_\phi(b,q)}\right|}{\sqrt{D_\phi(b,a)}}. \tag{A.7}$$

First consider $a < b < q$, and we assume $\lim_{b\to a}$. For ease of computation, we replace ϕ' by ψ, to be restored at the last step. We will use the following Taylor expansions repeatedly in our derivation: $b = a+h$, $\phi(b) = \phi(a+h) = \phi(a)+h\phi'(a)+$

$E(h^2)$, $\phi(b) = \phi(a) + h\psi(a) + \frac{h^2\psi'(a)}{2} + E(h^3)$ and $\sqrt{1+h} = 1 + h/2 + E(h^2)$. Here $E(h^x)$ denotes a tail of a Taylor expansion where the lowest order term is h^x. Since we will be handling multiple Taylor expansions in what follows, we will use subscripts of the form E_1, E_2, etc. to distinguish the tails of different series.

$$\begin{aligned}
&\lim_{a\to b} \frac{\sqrt{D_\phi(a,q)} - \sqrt{D_\phi(b,q)}}{\sqrt{D_\phi(b,a)}} \\
&= \lim_{a\to b} \frac{\sqrt{\phi(a)-\phi(q)-\psi(q)(a-q)} - \sqrt{\phi(b)-\phi(q)-\psi(q)(b-q)}}{\sqrt{\phi(b)-\phi(a)-\psi(a)(b-a)}}.
\end{aligned} \tag{A.8}$$

Computing the denominator by replacing $b-a$ with h and taking the Taylor expansion of $\phi(b)$:

$$\begin{aligned}
&\sqrt{\phi(b)-\phi(a)-\psi(a)(b-a)} \\
&\quad = \sqrt{\left(\phi(a)+h\psi(a)+\frac{h^2\psi'(a)}{2}+E_1(h^3)\right)-\phi(a)-\psi(a)h} \\
&\quad = \sqrt{\frac{h^2\psi'(a)}{2}+E_1(h^3)} \\
&\quad = h\sqrt{\frac{\psi'(a)}{2}+E_2(h^2)},
\end{aligned}$$

where in the last step, we let $E_1(h^3) = hE_2(h^2)$. We now address the numerator:

$$\begin{aligned}
&\left(\sqrt{\phi(a)-\phi(q)-\psi(q)(a-q)} - \sqrt{\phi(b)-\phi(q)-\psi(q)(b-q)}\right) \\
&\quad = \sqrt{\phi(a)-\phi(q)-\psi(q)(a-q)} - \sqrt{\phi(b)-\phi(q)-\psi(q)(b-a+a-q)} \\
&\quad = \sqrt{\phi(a)-\phi(q)-\psi(q)(a-q)} - \sqrt{\phi(b)-\phi(q)-\psi(q)(h+a-q)} \\
&\quad = \sqrt{\phi(a)-\phi(q)-\psi(q)(a-q)} \\
&\qquad - \sqrt{\phi(a)+h\psi(a)+E_3(h^2)-\phi(q)-\psi(q)(h+a-q)},
\end{aligned}$$

where in the last step we took the Taylor Expansion of $\phi(b)$. Collecting terms of h and continuing, we obtain:

$$\begin{aligned}
&\sqrt{\phi(a)-\phi(q)-\psi(q)(a-q)} \\
&\quad -\sqrt{\phi(a)-\phi(q)-\psi(q)(a-q)+h(\psi(a)-\psi(q))+E_3(h^2)} \\
&\quad = \sqrt{D_\phi(a,q)}-\sqrt{D_\phi(a,q)\left(1+\frac{h(\psi(a)-\psi(q))+E_3(h^2)}{D_\phi(a,q)}\right)} \\
&\quad = \sqrt{D_\phi(a,q)}\left(1-\sqrt{1-\frac{h(\psi(q)-\psi(a))-E_3(h^2)}{D_\phi(a,q)}}\right) \\
&\quad = \sqrt{D_\phi(a,q)}\left(1-\left(1-\frac{h(\psi(q)-\psi(a))}{2D_\phi(a,q)}+E_4(h^2)\right)\right) \\
&\quad = \frac{h\left(\psi(q)-\psi(a)-E_4(h^2)\right)}{2\sqrt{D_\phi(a,q)}},
\end{aligned}$$

where we note in the above that the new error term of $E_4(h^2)$ was produced by combining $E_3(h^2)$ with the error term produced by taking the Maclaurin expansion of the square root.

Now combine numerator and denominator back in equation A.8 and cancel a factor of h accordingly, we get:

$$\begin{aligned}
&\frac{\sqrt{\phi(a)-\phi(q)-\psi(q)(a-q)}-\sqrt{\phi(b)-\phi(q)-\psi(q)(b-q)}}{\sqrt{\phi(b)-\phi(a)-\psi(a)(b-a)}} \\
&= \left(\frac{h\left(\psi(q)-\psi(a)-E_4(h^2)\right)}{2\sqrt{D_\phi(a,q)}}\right) / \left(h\sqrt{\frac{\psi'(a)}{2}+E_2(h^2)}\right) \\
&= \left(\frac{\left(\psi(q)-\psi(a)-E_4(h^2)\right)}{2\sqrt{D_\phi(a,q)}}\right) / \left(\sqrt{\frac{\psi'(a)}{2}+E_2(h^2)}\right).
\end{aligned}$$

Now if we take $\lim_{h\to 0}$ or equivalent $\lim_{a\to b}$, neither the numerator nor denominator of this new expression become 0 and indeed we may drop the higher order terms of h safely. Noting that $D_\phi(a,q)=\frac{1}{2}(\psi'(x))(q-a)^2$, for some $x\in[ab]$.

$$\begin{aligned}
\lim_{a\to b}\frac{\sqrt{D_\phi(a,q)}-\sqrt{D_\phi(b,q)}}{\sqrt{D_\phi(b,a)}} &= \frac{\frac{(\psi(q)-\psi(a))}{2\sqrt{D_\phi(a,q)}}}{\sqrt{\frac{\psi'(a)}{2}}} \\
&= \frac{(\psi(q)-\psi(a))}{q-a}\frac{\sqrt{\psi'(a)}}{\sqrt{\psi'(x)}}.
\end{aligned}$$

Substituting back $\phi'(x)$ for $\psi(x)$, we see that limit A.7 exists, provided ϕ is strictly convex:

$$\frac{(\phi'(q)-\phi'(a))}{q-a}\frac{\sqrt{\phi''(a)}}{\sqrt{\phi''(x)}}. \tag{A.9}$$

The analysis follows symmetrically for case 4, by noting that $\lim_{a\to b}\frac{\sqrt{D_\phi(a,b)}}{\sqrt{D_\phi(b,a)}} = 1$ and that $\sqrt{D_\phi(a,q)} - \sqrt{D_\phi(b,q)} = -(\sqrt{D_\phi(b,q)} - \sqrt{D_\phi(a,q)})$, i.e we may suitably interchange a and b.

Cases 5 and 6: Here $q < a < b$ or $q < b < a$. Looking more carefully at the analysis for cases 3 and 4, note that the ordering $q < a < b$ vs $a < b < q$ does not affect the magnitude of the expression for limit A.7, only the sign. Hence we can use the same analysis to prove μ-defectiveness for cases 5 and 6.

■

Corollary A.1 *Given any interval $I = [x_1 x_2]$ on the real line, there exists a finite μ such that $\sqrt{D_\phi}$ is left-sided μ-defective with respect to I.*

Proof. Follows from similar computation. ■

A.1 Discussion of empirical values of μ

We calculate now the values of μ observed for a selection of Bregman divergences points spread over a range of intervals, namely [0.10.9], [0.010.99] and [0.0010.999]. Note that each of the values below is for the square root of the relevant divergence and that for the Itakura Saito, Kullback-Liebler and Symmetrized Kullback-Liebler, 0 is a boundary point where distances approach infinity. Interestingly, lemma 2.2.7 implies that whatever bounds for μ hold for points spread on an interval $I \in \mathbb{R}$ also hold for points in the box $\prod_{i=1}^{d} I^d \in \mathbb{R}^d$. We observe that for reasonable spreads of points, while μ is not necessarily always small, it is also not a galactic constant as well.

Name	Interval Range	Value of μ
Itakura-Saito	[0.1 0.9]	2.35
	[0.01 0.99]	7.17
	[0.001 0.999]	22.42
Kullback-Liebler	[0.1 0.9]	1.65
	[0.01 0.99]	3.67
	[0.001 0.999]	9.18
Symmetrized Kullback-Liebler	[0.1 0.9]	1.22
	[0.01 0.99]	2.42
	[0.001 0.999]	6.05
Exponential	[0.1 0.9]	1.14
	[0.001 0.999]	1.18
	[0.001 100]	9.95

REFERENCES

[1] ABDULLAH, A., MOELLER, J., AND VENKATASUBRAMANIAN, S. Approximate Bregman near neighbors in sublinear time: Beyond the triangle inequality. In *Proceedings of the 2012 Symposium on Computational Geometry* (2012), ACM, pp. 31–40.

[2] ACKERMANN, M., AND BLÖMER, J. Bregman clustering for separable instances. In *Algorithm Theory - SWAT 2010*, H. Kaplan, Ed., vol. 6139 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2010, pp. 212–223.

[3] ACKERMANN, M. R., AND BLÖMER, J. Coresets and approximate clustering for Bregman divergences. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms* (Philadelphia, PA, USA, 2009), SODA '09, Society for Industrial and Applied Mathematics, pp. 1088–1097.

[4] ACKERMANN, M. R., BLÖMER, J., AND SOHLER, C. Clustering for metric and nonmetric distance measures. *ACM Transactions on Algorithms (TALG) 6*, 4 (2010), 59.

[5] AFSHANI, P., ARGE, L., AND LARSEN, K. D. Orthogonal range reporting in three and higher dimensions. *Foundations of Computer Science, Annual IEEE Symposium on 0* (2009), 149–158.

[6] AHLBERG, D., BROMAN, E., GRIFFITHS, S., AND MORRIS, R. Noise sensitivity in continuum percolation. *arXiv preprint arXiv:1108.0310* (2011).

[7] AHN, K. J., GUHA, S., AND MCGREGOR, A. Graph sketches: sparsification, spanners, and subgraphs. In *Proceedings of the 31st Symposium on Principles of Database Systems* (2012), ACM, pp. 5–14.

[8] AIGER, D., KAPLAN, H., AND SHARIR, M. Reporting neighbors in high-dimensional euclidean spaces. In *SODA* (2013), pp. 784–803.

[9] AILON, N., AND CHAZELLE, B. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing(STOC)* (2006), ACM, pp. 557–563.

[10] ALON, N. Problems and results in extremal combinatorics I. *Discrete Mathematics 273* (2003), 31–53.

[11] ALON, N., MATIAS, Y., AND SZEGEDY, M. The space complexity of approximating the frequency moments. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing* (1996), ACM, pp. 20–29.

[12] AMARI, S., AND NAGAOKA, H. *Methods of Information Geometry*, vol. 191 of *Translations of Mathematical Monographs*. Oxford University Press, 2000.

[13] ANDONI, A., CHARIKAR, M. S., NEIMAN, O., AND NGUYEN, H. L. Near linear lower bound for dimension reduction in ℓ_1. In *Proceedings of the Fifty-second Annual IEEE Symposium on Foundations of Computer Science (FOCS)* (2011), IEEE, pp. 315–323.

[14] ANDONI, A., AND INDYK, P. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proceedings of the Forty-seventh Annual IEEE Symposium on Foundations of Computer Science(FOCS)* (2006), IEEE, pp. 459–468.

[15] ANDONI, A., INDYK, P., AND KRAUTHGAMER, R. Overcoming the ℓ_1 non-embeddability barrier: algorithms for product metrics. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms* (2009), Society for Industrial and Applied Mathematics, pp. 865–874.

[16] ANDONI, A., INDYK, P., NGUYEN, H. L., AND RAZENSHTEYN, I. Beyond locality-sensitive hashing. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms* (2014), SIAM, pp. 1018–1028.

[17] ANDONI, A., INDYK, P., AND PĂTRAŞCU, M. On the optimality of the dimensionality reduction method. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS)* (2006), pp. 449–458.

[18] ANDONI, A., KRAUTHGAMER, R., AND RAZENSHTEYN, I. Sketching and embedding are equivalent for norms. *arXiv preprint arXiv:1411.2577* (2014).

[19] ANDONI, A., AND RAZENSHTEYN, I. Optimal data-dependent hashing for approximate near neighbors. *arXiv preprint arXiv:1501.01062* (2015).

[20] ARORA, S., AND KANNAN, R. Learning a mixture of gaussians. *Proceedings of the Symposium on Theory of Computing (STOC)* (2001).

[21] ARYA, S., AND MALAMATOS, T. Linear-size approximate voronoi diagrams. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2002), pp. 147–155.

[22] ARYA, S., MALAMATOS, T., AND MOUNT, D. M. Space-time tradeoffs for approximate nearest neighbor searching. *Journal of the ACM (JACM) 57*, 1 (2009), 1.

[23] ARYA, S., MOUNT, D. M., NETANYAHU, N. S., SILVERMAN, R., AND WU, A. Y. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM 45* (November 1998), 891–923.

[24] AURENHAMMER, F. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR) 23*, 3 (1991), 345–405.

[25] AVRON, H., NGUYEN, H., AND WOODRUFF, D. Subspace embeddings for the polynomial kernel. In *NIPS* (2014), pp. 2258–2266.

[26] BAI, L., AND HANCOCK, E. R. Graph kernels from the Jensen-Shannon divergence. *Journal of Mathematical Imaging and Vision 47*, 1-2 (2013), 60–69.

[27] BANERJEE, A., MERUGU, S., DHILLON, I. S., AND GHOSH, J. Clustering with bregman divergences. *J. Mach. Learn. Res. 6* (December 2005), 1705–1749.

[28] BARKOL, O., AND RABANI, Y. Tighter bounds for nearest neighbor search and related problems in the cell probe model. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing* (2000), ACM, pp. 388–396.

[29] BARTAL, Y. On approximating arbitrary metrices by tree metrics. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing* (1998), ACM, pp. 161–168.

[30] BENJAMINI, I., KALAI, G., AND SCHRAMM, O. Noise sensitivity of Boolean functions and applications to percolation. *Publications Mathématiques de l'Institut des Hautes Études Scientifiques 90*, 1 (1999), 5–43.

[31] BERAN, R. Minimum hellinger distance estimates for parametric models. *The Annals of Statistics* (1977), 445–463.

[32] BEYGELZIMER, A., KAKADE, S., AND LANGFORD, J. Cover trees for nearest neighbor. In *Proceedings of the 23rd International Conference on Machine Learning* (New York, NY, USA, 2006), ICML '06, ACM, pp. 97–104.

[33] BIAU, G., DEVROYE, L., AND LUGOSI, G. On the performance of clustering in Hilbert spaces. *IEEE Transactions on Information Theory 54*, 2 (2008), 781–790.

[34] BOISSONNAT, J.-D., NIELSEN, F., AND NOCK, R. Bregman voronoi diagrams. *Discrete and Computational Geometry 44* (2010), 281–307. 10.1007/s00454-010-9256-1.

[35] BORODIN, A., OSTROVSKY, R., AND RABANI, Y. Lower bounds for high dimensional nearest neighbor search and related problems. In *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 1999), STOC '99, ACM, pp. 312–321.

[36] BOUTSIDIS, C., ZOUZIAS, A., AND DRINEAS, P. Random projections for k-means clustering. In *Advances in Neural Information Processing Systems* (2010), pp. 298–306.

[37] BREGMAN, L. M. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics 7* (1967), 200–217.

[38] BRINKMAN, B., AND CHARIKAR, M. On the impossibility of dimension reduction in l_1. *Journal of the ACM (JACM) 52*, 5 (2005), 766–788.

[39] CAYTON, L. Fast nearest neighbor retrieval for bregman divergences. In *Proceedings of the 25th International Conference on Machine Learning* (New York, NY, USA, 2008), ICML '08, ACM, pp. 112–119.

[40] CAYTON, L. *Bregman proximity search.* PhD thesis, University of California, San Diego, 2009.

[41] CHAKRABARTI, A., AND REGEV, O. An optimal randomised cell probe lower bound for approximate nearest neighbour searching. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on* (2004), IEEE, pp. 473–482.

[42] CHAKRABARTY, D., AND SESHADHRI, C. A o(n) monotonicity tester for Boolean functions over the hypercube. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing* (2013), ACM, pp. 411–418.

[43] CHAUDHURI, K., AND MCGREGOR, A. Finding metric structure in information theoretic clustering. In *Conference on Learning Theory(COLT)* (2008), vol. 8, Citeseer, p. 10.

[44] CHITNIS, R., CORMODE, G., ESFANDIARI, H., HAJIAGHAYI, M., MCGREGOR, A., MONEMIZADEH, M., AND VOROTNIKOVA, S. Kernelization via sampling with applications to dynamic graph streams. *arXiv preprint arXiv:1505.01731* (2015).

[45] CLARKSON, K. An algorithm for approximate closest-point queries. *Proceedings of the Tenth Annual ACM Symposium on Computational Geometry* (1994), 160–164.

[46] CLARKSON, K. L. A randomized algorithm for closest-point queries. *SIAM Journal on Computing 17*, 4 (1988), 830–847.

[47] CLARKSON, K. L. Nearest-neighbor searching and metric space dimensions. *Nearest-neighbor Methods for Learning and Vision: Theory and Practice* (2006), 15–59.

[48] COHEN, M., ELDER, S., MUSCO, C., MUSCO, C., AND PERSU, M. Dimensionality reduction for k-means clustering and low rank approximation. *arXiv preprint arXiv:1410.6801* (2014).

[49] COLE, R., AND GOTTLIEB, L.-A. Searching dynamic point sets in spaces with bounded doubling dimension. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing* (2006), ACM, pp. 574–583.

[50] COLE, R., AND GOTTLIEB, L.-A. Searching dynamic point sets in spaces with bounded doubling dimension. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 2006), STOC '06, ACM, pp. 574–583.

[51] COLLINS, M., SCHAPIRE, R., AND SINGER, Y. Logistic regression, adaboost and bregman distances. *Machine Learning 48*, 1 (2002), 253–285.

[52] COVER, T. M., AND HART, P. E. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on 13*, 1 (1967), 21–27.

[53] COVER, T. M., AND THOMAS, J. A. *Elements of Information Theory*. Wiley-Interscience, New York, NY, USA, 1991.

[54] CSISZÁR, I. Information-type measures of difference of probability distributions and indirect observations. *Studia Scientiarum Mathematicarum Hungarica 2* (1967), 299–318.

[55] CSISZÁR, I. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability 3* (1975), 146–158.

[56] DASGUPTA, S. Learning mixtures of gaussians. In *40th Annual Symposium on Foundations of Computer Science* (1999), IEEE Computer Society, pp. 634–.

[57] DASGUPTA, S., AND FREUND, Y. Random projection trees and low dimensional manifolds. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing* (2008), ACM, pp. 537–546.

[58] DATAR, M., IMMORLICA, N., INDYK, P., AND MIRROKNI, V. S. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry* (2004), ACM, pp. 253–262.

[59] DAVIS, C., AND KAHAN, W. M. The rotation of eigenvectors by a perturbation. III. *SIAM J. Numer. Anal. 7* (1970), 1–46.

[60] DENOEUX, T. A k-nearest neighbor classification rule based on dempster-shafer theory. *Systems, Man and Cybernetics, IEEE Transactions on 25*, 5 (1995), 804–813.

[61] DHILLON, I. S., MALLELA, S., AND KUMAR, R. A divisive information theoretic feature clustering algorithm for text classification. *The Journal of Machine Learning Research 3* (2003), 1265–1287.

[62] DIACONIS, P., SALOFF-COSTE, L., ET AL. Logarithmic sobolev inequalities for finite markov chains. *The Annals of Applied Probability 6*, 3 (1996), 695–750.

[63] EIRON, N., AND MCCURLEY, K. S. Analysis of anchor text for web search. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval* (2003), ACM, pp. 459–460.

[64] EPPSTEIN, D., GOODRICH, M. T., AND SUN, J. Z. The skip quadtree: a simple dynamic data structure for multidimensional data. In *Proceedings of the Twenty-first Annual Symposium on Computational Geometry* (New York, NY, USA, 2005), SCG '05, ACM, pp. 296–305.

[65] FAKCHAROENPHOL, J., RAO, S., AND TALWAR, K. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing* (2003), ACM, pp. 448–455.

[66] FARAGO, A., LINDER, T., AND LUGOSI, G. Fast nearest-neighbor search in dissimilarity spaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 15*, 9 (sep 1993), 957 –962.

[67] FRADKIN, D., AND MADIGAN, D. Experiments with random projections for machine learning. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2003), ACM, pp. 517–522.

[68] FRIEDGUT, E. Boolean functions with low average sensitivity depend on few coordinates. *Combinatorica 18*, 1 (1998), 27–35.

[69] FUGLEDE, B., AND TOPSØE, F. Jensen-Shannon divergence and Hilbert space embedding. In *Proceedings of the 2004 IEEE International Symposium on Information Theory(ISIT)* (2004), pp. 31–31.

[70] GOLDBERGER, J., GORDON, S., AND GREENSPAN, H. An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on* (2003), IEEE, pp. 487–493.

[71] GRAY, R., BUZO, A., GRAY JR, A., AND MATSUYAMA, Y. Distortion measures for speech processing. *Acoustics, Speech and Signal Processing, IEEE Transactions on 28*, 4 (Aug 1980), 367–376.

[72] GRITZMANN, P., AND KLEE, V. Computational complexity of inner and outer j-radii of polytopes in finite-dimensional normed spaces. *Mathematical Programming 59*, 1-3 (1993), 163–213.

[73] GUHA, S., INDYK, P., AND MCGREGOR, A. Sketching information divergences. In *Learning Theory*. Springer, 2007, pp. 424–438.

[74] GUHA, S., MCGREGOR, A., AND VENKATASUBRAMANIAN, S. Streaming and sublinear approximation of entropy and information distances. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms(SODA)* (2006), ACM, pp. 733–742.

[75] HAR-PELED, S. A replacement for voronoi diagrams of near linear size. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science* (Washington, DC, USA, 2001), FOCS '01, IEEE Computer Society, pp. 94–105.

[76] HAR-PELED, S., AND KUMAR, N. Approximate nearest neighbor search for low-dimensional queries. *SIAM J. Comput. 42*, 1 (2013), 138–159. Previously in SODA'11.

[77] HAR-PELED, S., AND MAZUMDAR, S. Fast algorithms for computing the smallest k-enclosing circle. *Algorithmica 41* (2005), 147–157. 10.1007/s00453-004-1123-0.

[78] HAR-PELED, S., AND MENDEL, M. Fast construction of nets in low dimensional metrics, and their applications. In *Proceedings of the Twenty-first Annual Symposium on Computational Geometry* (New York, NY, USA, 2005), SCG '05, ACM, pp. 150–158.

[79] HAR-PELED, S., AND MENDEL, M. Fast construction of nets in low-dimensional metrics and their applications. *SIAM Journal on Computing 35*, 5 (2006), 1148–1184.

[80] HAR-PELED, S., AND VARADARAJAN, K. R. High-dimensional shape fitting in linear time. *Discrete & Computational Geometry 32*, 2 (2004), 269–288.

[81] HERSHEY, J. R., OLSEN, P., ET AL. Approximating the kullback leibler divergence between gaussian mixture models. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on* (2007), vol. 4, IEEE, pp. IV–317.

[82] HUANG, X., LI, S. Z., AND WANG, Y. Jensen-Shannon boosting learning for object recognition. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR)* (2005), vol. 2 of *CVPR '05*, pp. 144–149.

[83] INDYK, P. Approximate nearest neighbor algorithms for fréchet distance via product metrics. In *Proceedings of the Eighteenth Annual Symposium on Computational Geometry* (2002), ACM, pp. 102–106.

[84] INDYK, P., AND MOTWANI, R. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 1998), STOC '98, ACM, pp. 604–613.

[85] INDYK, P., AND NAOR, A. Nearest-neighbor-preserving embeddings. *ACM Transactions on Algorithms (TALG) 3*, 3 (2007), 31.

[86] INDYK, P., AND WOODRUFF, D. Tight lower bounds for the distinct elements problem. *Proceedings of the Symposium on Foundations of Computer Science (FOCS)* (2003), 283–290.

[87] JOHNSON, W., AND SCHECHTMAN, G. Embedding ℓ_p^m into ℓ_1^n. *Acta Mathematica 149* (1982), 71–85.

[88] JOHNSON, W. B., AND LINDENSTRAUSS, J. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics 26*, 189-206 (1984), 1.

[89] KANNAN, R., AND VEMPALA, S. Spectral algorithms. *Found. Trends Theor. Comput. Sci. 4*, 3&#8211;4 (Mar. 2009), 157–288.

[90] KARGER, D., AND RUHL, M. Finding nearest neighbors in growth-restricted metrics. *Proceedings of the Symposium on Theory of Computing (STOC)* (2002).

[91] KELLER, N. A simple reduction from a biased measure on the discrete cube to the uniform measure. *arXiv preprint arXiv:1001.1167* (2010).

[92] KHOT, S., MINZNER, D., AND SAFRA, M. On monotonicity testing and Boolean isoperimetric type theorems. *Electronic Colloquium on Computational Complexity*, TR15-011 (2015). http://eccc.hpi-web.de/report/2015/011/.

[93] KINDLER, G. *PCP, Property Testing and Juntas.* PhD thesis, Tel-Aviv University, 2002.

[94] KRAUTHGAMER, R., AND LEE, J. The black-box complexity of nearest neighbor search. In *Automata, Languages and Programming*, J. Daz, J. Karhumki, A. Lepist, and D. Sannella, Eds., vol. 3142 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2004, pp. 153–178.

[95] KRAUTHGAMER, R., AND LEE, J. R. Navigating nets: simple algorithms for proximity search. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (Philadelphia, PA, USA, 2004), SODA '04, Society for Industrial and Applied Mathematics, pp. 798–807.

[96] KYNG, R. J., PHILLIPS, J. M., AND VENKATASUBRAMANIAN, S. Johnson-Lindenstrauss dimensionality reduction on the simplex. In *20th Fall Workshop on Computational Geometry* (2010).

[97] LARSEN, K. G. Higher cell probe lower bounds for evaluating polynomials. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on* (2012), IEEE, pp. 293–301.

[98] LATAŁA, R. Some estimates of norms of random matrices. *Proceedings of the American Mathematical Society 133*, 5 (2005), 1273–1282.

[99] LAURENT, B., AND MASSARAT, P. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics 28* (1998), 1303–1338.

[100] LEE, J. R., MENDEL, M., AND NAOR, A. Metric structures in l1: dimension, snowflakes, and average distortion. *European Journal of Combinatorics 26*, 8 (2005), 1180–1190.

[101] LEE, J. R., AND NAOR, A. Embedding the diamond graph in L_p and dimension reduction in L_1. *Geometric & Functional Analysis GAFA 14*, 4 (2004), 745–747.

[102] LI, Y., NGUYEN, H. L., AND WOODRUFF, D. P. Turnstile streaming algorithms might as well be linear sketches. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing(STOC)* (2014), ACM, pp. 174–183.

[103] LIU, D. A strong lower bound for approximate nearest neighbor searching. *Information Processing Letters 92*, 1 (2004), 23–29.

[104] MAHALANOBIS, P. C. On the generalised distance in statistics. *Proc. National Institute of Sciences in India 2*, 1 (1936), 49–55.

[105] MAHMOUDI, M., AND SAPIRO, G. Three-dimensional point cloud recognition via distributions of geometric distances. *Graphical Models 71*, 1 (2009), 22–31.

[106] MANTHEY, B., AND RÖGLIN, H. Worst-case and smoothed analysis of k-means clustering with bregman divergences. In *Algorithms and Computation*, Y. Dong, D.-Z. Du, and O. Ibarra, Eds., vol. 5878 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2009, pp. 1024–1033.

[107] MANTHEY, B., AND RÖGLIN, H. Worst-case and smoothed analysis of k-means clustering with bregman divergences. In *Algorithms and Computation*. Springer, 2009, pp. 1024–1033.

[108] MARTÍN, A., LÓPEZ-ROSA, S., ANGULO, J., AND ANTOLÍN, J. Jensen–shannon and kullback–leibler divergences as quantifiers of relativistic effects in neutral atoms. *Chemical Physics Letters 635* (2015), 75–79.

[109] MCNAMES, J. A fast nearest-neighbor algorithm based on a principal axis search tree. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 23*, 9 (2001), 964–976.

[110] MEISER, S. Point location in arrangements of hyperplanes. *Information and Computation 106* (1993), 286–303.

[111] MOITRA, A., AND VALIANT, G. Settling the polynomial learnability of mixtures of gaussians. In *51st Annual IEEE Symposium on Foundations of Computer Science* (2010), IEEE, pp. 93–102.

[112] MOTWANI, R., NAOR, A., AND PANIGRAHY, R. Lower bounds on locality sensitive hashing. *SIAM Journal on Discrete Mathematics 21*, 4 (2007), 930–935.

[113] MUJA, M., AND LOWE, D. G. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP (1)* (2009), pp. 331–340.

[114] NAGHSHVAR, M., JAVIDI, T., AND WIGGER, M. Extrinsic jensen–shannon divergence: Applications to variable-length coding. *Information Theory, IEEE Transactions on 61*, 4 (2015), 2148–2164.

[115] NGUYÊN, H. L. Approximate nearest neighbor search in ℓ_p. *arXiv preprint arXiv:1306.3601* (2013).

[116] NIELSEN, F., AND BOLTZ, S. The burbea-rao and bhattacharyya centroids. *CoRR abs/1004.5049* (2010).

[117] NIELSEN, F., AND NOCK, R. On the smallest enclosing information disk. *Information Processing Letters 105*, 3 (2008), 93 – 97.

[118] NIELSEN, F., AND NOCK, R. Sided and symmetrized bregman centroids. *IEEE Trans. Inf. Theor. 55* (June 2009), 2882–2904.

[119] NIELSEN, F., PIRO, P., AND BARLAUD, M. Bregman vantage point trees for efficient nearest neighbor queries. In *Proceedings of the 2009 IEEE International Conference on Multimedia and Expo* (Piscataway, NJ, USA, 2009), ICME'09, IEEE Press, pp. 878–881.

[120] NIELSEN, F., PIRO, P., AND BARLAUD, M. Tailored bregman ball trees for effective nearest neighbors. In *In European Workshop on Computational Geometry* (2009).

[121] NOCK, R., LUOSTO, P., AND KIVINEN, J. Mixed bregman clustering with approximation guarantees. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 2008, pp. 154–169.

[122] NOCK, R., AND NIELSEN, F. Fitting the smallest enclosing bregman ball. In *Machine Learning: ECML 2005*. Springer, 2005, pp. 649–656.

[123] *Frontiers in Massive Data Analysis*. The National Academies Press, 2013.

[124] O'DONNELL, R. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[125] O'DONNELL, R., WU, Y., AND ZHOU, Y. Optimal lower bounds for locality-sensitive hashing (except when q is tiny). *ACM Trans. Comput. Theory 6*, 1 (Mar. 2014), 5:1–5:13.

[126] OLESZKIEWICZ, K. On a nonsymmetric version of the khinchine-kahane inequality. In *Stochastic Inequalities and Applications*. Springer, 2003, pp. 157–168.

[127] PANIGRAHY, R. Entropy-based nearest neighbor algorithm in high dimensions. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2006).

[128] PANIGRAHY, R., TALWAR, K., AND WIEDER, U. A geometric approach to lower bounds for approximate near-neighbor search and partial match. In *Foundations of Computer Science, 2008. FOCS '08. IEEE 49th Annual IEEE Symposium on* (oct. 2008), pp. 414 –423.

[129] PANIGRAHY, R., TALWAR, K., AND WIEDER, U. Lower bounds on near neighbor search via metric expansion. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on* (oct. 2010), pp. 805 –814.

[130] PĂTRAŞCU, M. Unifying the landscape of cell-probe lower bounds. *SIAM Journal on Computing 40*, 3 (2011), 827–847.

[131] PĂTRAŞCU, M., AND THORUP, M. Time-space trade-offs for predecessor search. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing* (2006), ACM, pp. 232–240.

[132] PETER, A. M., AND RANGARAJAN, A. Maximum likelihood wavelet density estimation with applications to image and shape matching. *IEEE Transactions on Image Processing 17*, 4 (2008), 458–468.

[133] RABINOVICH, Y. On average distortion of embedding metrics into the line and into l 1. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing* (2003), ACM, pp. 456–462.

[134] RAHIMI, A., AND RECHT, B. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems* (2007), Neural Information Processing Systems, pp. 1177–1184.

[135] REGEV, O. Entropy-based bounds on dimension reduction in l 1. *Israel Journal of Mathematics 195*, 2 (2013), 825–832.

[136] RUDELSON, M., AND VERSHYNIN, R. Sampling from large matrices: an approach through geometric functional analysis. *Journal of the ACM (JACM) 54*, 4 (2007), 21.

[137] RUDELSON, M., AND VERSHYNIN, R. Non-asymptotic theory of random matrices: extreme singular values. In *Proceedings of the International Congress of Mathematicians. Volume III* (New Delhi, 2010), Hindustan Book Agency, pp. 1576–1602.

[138] SALAKHUTDINOV, R., AND HINTON, G. Semantic hashing. *International Journal of Approximate Reasoning 50*, 7 (2009), 969–978.

[139] SARIEL-HAR-PELED. *Geometric Approximation Algorithms*. AMS, 2011. http://goo.gl/pLiEO.

[140] SILPA-ANAN, C., AND HARTLEY, R. Optimised kd-trees for fast image descriptor matching. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), IEEE, pp. 1–8.

[141] SILVA, J., AND NARAYANAN, S. Average divergence distance as a statistical discrimination measure for hidden markov models. *Audio, Speech, and Language Processing, IEEE Transactions on 14*, 3 (2006), 890–906.

[142] SPIELMAN, D. A., AND TENG, S.-H. Smoothed analysis: an attempt to explain the behavior of algorithms in practice. *Commun. ACM 52*, 10 (Oct. 2009), 76–84.

[143] SPROULL, R. Refinements to nearest-neighbor searching in k-dimensional trees. *Algorithmica 6* (1991), 579–589.

[144] SRA, S., JEGELKA, S., AND BANERJEE, A. Approximation algorithms for Bregman clustering, co-clustering and tensor clustering. Tech. rep., Technical Report 177, MPI for Biological Cybernetics, 2008.

[145] STEWART, G. W. On the early history of the singular value decomposition. *SIAM review 35*, 4 (1993), 551–566.

[146] TALAGRAND, M. On russo's approximate zero-one law. *The Annals of Probability* (1994), 1576–1587.

[147] TANG, K., PALURI, M., FEI-FEI, L., FERGUS, R., AND BOURDEV, L. Improving image classification with location context. *arXiv preprint arXiv:1505.03873* (2015).

[148] TOPSØE, F. Some inequalities for information divergence and related measures of discrimination. *IEEE Transactions on Information Theory 46*, 4 (2000), 1602–1609.

[149] VARADARAJAN, K. R., VENKATESH, S., AND ZHANG, J. On approximating the radii of point sets in high dimensions. In *43rd Annual IEEE Symposium on Foundations of Computer Science* (2002), IEEE, pp. 561–569.

[150] VEDALDI, A., AND ZISSERMAN, A. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI) 34*, 3 (2012), 480–492.

[151] VEMPALA, S., AND WANG, G. A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences 68*, 4 (2004), 841–860. Previously in FOCS'02.

[152] VERMA, N., KPOTUFE, S., AND DASGUPTA, S. Which spatial partition trees are adaptive to intrinsic dimension? In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (2009), AUAI Press, pp. 565–574.

[153] WANG, Y., AND YIN, Y. Certificates in data structures. *arXiv preprint arXiv:1404.5743* (2014).

[154] WEDIN, P.-Å. Perturbation bounds in connection with singular value decomposition. *BIT Numerical Mathematics 12*, 1 (1972), 99–111.

[155] WEISS, Y., TORRALBA, A., AND FERGUS, R. Spectral hashing. In *Advances in neural information processing systems* (2008), pp. 1753–1760.

[156] WOODRUFF, D. Optimal space lower bounds for all frequency moments. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2004).

[157] YAGNIK, J., STRELOW, D., ROSS, D. A., AND LIN, R.-S. The power of comparative reasoning. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (2011), IEEE, pp. 2431–2438.

[158] YI, X., CARAMANIS, C., AND PRICE, E. Binary embedding: Fundamental limits and fast algorithm. *arXiv preprint arXiv:1502.05746* (2015).

[159] ZHANG, Z., OOI, B. C., PARTHASARATHY, S., AND TUNG, A. K. H. Similarity search on bregman divergence: towards non-metric indexing. *Proc. VLDB Endow. 2* (August 2009), 13–24.

DISSEMINATION OF THIS WORK

• **Approximate Bregman near neighbors in sublinear time: Beyond the triangle inequality**.

Amirali Abdullah, John Moeller and Suresh Venkatasubramanian.

©ACM, Proceedings of Symposium on Computational Geometry (SOCG), June 2012.

• **A directed isoperimetric inequality with application to Bregman near neighbor lower bounds**.

Amirali Abdullah and Suresh Venkatasubramanian.

©ACM, Proceedings of Symposium on Theory of Computing (STOC), June 2015

• **Spectral Approaches to Nearest Neighbor Search**.

Amirali Abdullah, Alexandr Andoni, Ravindran Kannan and Robert Krauthgamer.

©IEEE, Proceedings of Symposium on Foundations of Computer Science (FOCS), October 2014.

• **Sketching, Embedding, and Dimensionality Reduction for Information Spaces**.

Amirali Abdullah, Ravi Kumar, Andrew McGregor, Sergei Vassilvitskii and Suresh Venkatasubramanian.

Under submission.