METRIC SPACE EMBEDDINGS INTO ℓ_1: AN OPTIMIZATION APPROACH

William John Brinkman II

A DISSERTATION
PRESENTED TO THE FACULTY
OF PRINCETON UNIVERSITY
IN CANDIDACY FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE
BY THE DEPARTMENT OF
COMPUTER SCIENCE

November 2004

Abstract

The technique of metric space embeddings into Euclidean space has been used extensively to give powerful approximation algorithms for a variety of problems. Much of the success of this technique has been due to the famous lemma of Johnson and Lindenstrauss which says that any n points in Euclidean space can be well approximated by vectors of length roughly $O(\log n)$ via random projections.

Embeddings into Euclidean space, however, are known to be somewhat limited: For example, there are planar graph metrics which require $\Omega(\sqrt{\log n})$ distortion in Euclidean space. Finite metric spaces in $\mathbb{R}^d$ under the ℓ_1 norm arise in practice as well, and it is known that for embedding this is a strictly richer class of metrics than the finite Euclidean metrics. This has motivated an interest in ℓ_1 as a possible host space for difficult metrics, particularly edit metrics and planar graph metrics.

We have developed an approach to the study of ℓ_1 embeddings based on mathematical optimization. We show how these techniques, through duality, lead naturally to *both* lower bounds and algorithms. We demonstrate this approach by proving that no analogue of the Johnson-Lindenstrauss lemma exists for ℓ_1. This answers the open question recently popularized by both Piotr Indyk and Nathan Linial.

This work is available online at `http://www.derandomized.org/`.

Acknowledgments

While writing this dissertation I have come to reflect on the choices and opportunities that brought me to this point. This dissertation communicates the work that I have done toward my degree, but in many ways this presents an incomplete picture. At every step in my life I have been surrounded by good people who have given me aid, knowledge, guidance and love. Without their contributions I could not have succeeded, and I would like to take a moment to thank them.

I want to thank the many teachers who have imparted their wisdom to me. Harriet Bell, Dick Starks and Joe Hawblitzel in particular influenced my understanding of the positive impact that a teacher can make on a student's life. I must especially thank the faculty of the University of Washington for giving me the opportunity to teach as an undergraduate. At the University of Washington I was supervised by Alan Borning, Martin Dickey, Richard Ladner, and Martin Tompa. At Princeton I have taught under Doug Clark, Robert Sedgewick and Kevin Wayne. Working with such a wide variety of very talented teachers has been both an honor and a tremendously valuable experience.

Richard Ladner and Martin Tompa were also the first to encourage me to work on independent research.

I would also like to thank everyone at Cerner Corporation: Joe Geris, Susan

Williams, Charlie Whitcraft and the HNA Millennium PowerChart team. Working as a programmer full time during the summers of 1996-1999 gave me a lot of insight into the application of algorithms research to real world problems.

I also owe many debts of gratitude to the faculty of Princeton's Computer Science Department. As my first-year advisor and co-author Sanjeev Arora not only started me on a productive research career, but also taught me much of what I know about writing and presenting. Along with Sanjeev, I would like to thank Amit Sahai and Robert Sedgewick for their service on my generals committee. My dissertation committee, Sanjeev Arora, Bernard Chazelle, Rob Schapire and Robert Sedgewick, have been steadfast in their support. I have also greatly appreciated their guidance in preparing this dissertation and my defense. Many of the main tools I use day to day were imparted to me by Sanjeev, Bernard, Amit, Bob Tarjan and Andy Yao through their excellent courses.

Our department administrators, Melissa Lawson and Mitra Kelly, have also played key parts in my success at Princeton. Melissa's hard work on my behalf in administrative matters has left me free to focus on my work. Mitra's cheerful approach to organizing programs and her eager help during my job search have been a great blessing.

Outside of work, my friends Elbert Chiang, Phil Gressman, Jim McCullough and Joan and David Linsenmeier have been a constant source of support. Matt Parmeter's unflagging friendship from afar has been a particular source of comfort. I want to thank everyone from Westerly Road Church and Washington Crossing Methodist Church for helping make Princeton feel like home.

I am thankful to my parents, Bill and Judy Brinkman, for always believing I would succeed, and giving me every opportunity to do so. They gave me their trust and respect, and gave me the freedom to do what I thought was right. I have been blessed with a wonderful family, and their impact cannot be over-stated.

I am especially grateful to my fiancé Stacy Nakamura for giving me the love and support I need to be successful. She has been my best friend and my most loyal fan.

Finally, I wish to thank my advisor, Moses Charikar. No other person has had a greater impact on my research career. He has been completely selfless in his support, and has gone out of his way to help me advance my career. His guidance in research, writing, speaking and career planning has been exceedingly valuable to me, and I appreciate it greatly.

Bo Brinkman
April 2004

I gratefully acknowledge the grants and awards that have supported this research: NSF ITR grant CCR-0205594, DOE award DE-FG02-02ER25540, NSF CAREER award CCR-0237113, an Alfred P. Sloan Fellowship (awarded to Moses Charikar), a Princeton University Research Board Tuition Award, a David and Lucile Packard Foundation Fellowship (awarded to Sanjeev Arora), NSF grant CCR-098180 and a Francis Upton Graduate Fellowship.

Contents

List of Tables

List of Figures

Chapter 1

Foundations

The primary purpose of this chapter is to introduce and motivate our study of metric space embeddings into ℓ_1. We will broadly define the types of problems we are interested in and demonstrate some of the most common proof techniques used in this area. At the same time we will also survey the most important and directly related research results, both classical and recent, in order to provide the reader with background and resources for further study. The chapter concludes by introducing our main theorems and outlining the remainder of the work.

1.1 Proximity and geometric problems

Many of the most fundamental problems of computer science involve sets of data points along with some measure of similarity or dissimilarity between them. Clustering, nearest-neighbor searching, computing the diameter of a point set, and web searching all fall into this general framework.

The efficient solution of these problems often depends critically on the properties of the space on which the problem is defined. For example, consider the problem of searching a database of points for the nearest neighbor of a given query point x. We are given the database of n points with oracle access to the distance function on the points, and we may pre-process the points to aid our search. If the items are points on a line, then the solution is simple. Sort the given n items in $O(n \log n)$ time and store the result in an array. Given an item x, searching for its nearest neighbor using binary search takes $O(\log n)$ time.

On the other hand, consider nearest-neighbor searching when the distance function $\Delta : X \times X \to \{0, 1\}$ is any arbitrary function. In this case each time we are given a new x we must make n calls to the distance oracle because we do not yet have any information about x's relationship to the points in the database.

In solving these types of problems the goal is usually to exploit the special structure of our problem space to get an efficient algorithm.

1.2 Metric space embeddings

Most commonly studied distance functions are metrics.

Definition 1.2.1 (Metric). *For a given set X, a metric is a function $\Delta : X \times X \to [0, \infty)$ such that, for all $u, v, w \in X$:*

1. $\Delta(u, v) = \Delta(v, u)$ *(Symmetry)*
2. $\Delta(u, w) + \Delta(w, v) \geq \Delta(u, v)$ *(Triangle inequality)*
3. $\Delta(u, v) = 0$ *if and only if* $u = v$

These restrictions on the distance function align very closely with our intuitive ideas of distance. Most of our results will deal with finite metric spaces, which we will denote $M = (X, \Delta)$, where X is a finite set and $\Delta : X \times X \to [0, \infty)$ is a metric. In some cases we will relax the last restriction and allow $\Delta(u, v) = 0$ even when $u \neq v$. In this case Δ is called a *semimetric*.[1]

Next we will define a natural notion of reduction between metric spaces which can be used to design algorithms.

Definition 1.2.2 (Isometric embedding of a metric space). *Given two metric spaces $M_1 = (X_1, \Delta_1)$ and $M_2 = (X_2, \Delta_2)$, an embedding of M_1 into M_2 is an injective map $f : X_1 \to X_2$ such that $\forall u, v \in X_1,\ \Delta_1(u, v) = \Delta_2(f(u), f(v))$.*

Consider a problem P which we want to solve on points taken from a metric space M_1. If we already have an algorithm that solves P in another metric space M_2, the simplest approach may be to try to find an algorithm that embeds M_1 into M_2 efficiently.

1.2.1 Motivation

As mentioned above, one main motivation for the technique of metric space embeddings has been the need to solve proximity problems on databases of points with a defined similarity (or distance) measure. For example, many computational genomics applications require searching or clustering a database of DNA or protein sequences. Usually the similarity of two sequences is defined by some string edit metric. Another

[1] Various sources refer to these functions as *pseudometrics*. We have chosen to follow the terminology of Deza and Laurent [20].

common example is web searching, where one defines some measure of similarity between web pages. In this setting a web search is a nearest neighbor problem, while services like Google's "find similar pages" can be viewed as a web page clustering problem. In digital libraries we would like to be able to search for images similar to a given image or songs similar to a given sample of music, or to automatically group media items into categories via clustering. All of these problems have a very similar description, but the underlying metrics seem to be very different.

The key to solving these problems efficiently is to exploit some known structure in the underlying metric. Unfortunately many interesting metrics are not well studied, and so useful structural theorems may not be known. The study of metric space embeddings can be viewed as one approach to studying the structure of a given metric space: If one can find an embedding of a metric space M_1 into a well studied metric space M_2, this allows one to immediately re-use the structural results that have already been proved for M_2.

Of course, from a more pragmatic point of view, one need not even directly understand the geometric implications of the relationship between M_1 and M_2. Often it is sufficient to find the embedding into M_2 and then simply use an algorithm that is already known to work well in M_2. This has led to the widespread use of the technique of metric space embeddings for the design of algorithms.

One important special case of this approach is to use metric embeddings to succinctly represent problem data. For example, consider the problem of finding approximate near neighbors in a high dimensional Euclidean space. Many nearest-neighbor algorithms suffer from the "curse of dimensionality" in that they have an exponential dependence on dimension. One way to circumvent this problem is to try to embed the high-dimensional Euclidean space into a low-dimensional one while only introducing a small error or failure probability. If the number of dimensions needed can be significantly reduced then we may be able to give efficient approximation algorithms with a much weaker dependence on dimension.

The technique of metric space embeddings has, however, had an impact far beyond algorithms for proximity problems. To paraphrase R. Ravi [66], there are at least three main ways that embeddings have been used to give approximation algorithms:

1. **Metric data.** Often we are given a problem with data that either comes directly from a metric, or can be embedded into a metric in a pre-processing step. The goal is to embed the data into an "easy" metric and use already existing algorithms. Most of our discussion of proximity problems falls into this area.

2. **Metric relaxations.** In some cases our problem can be formulated as an optimization problem, and then relaxed so that the resulting problem is a problem of finding good metric space embeddings. In fact, rounding algorithms for linear program and semidefinite program relaxations have resulted in some new approximation algorithms (e.g. Charikar [14]).

3. **Problems on metrics.** There are some problems where the goal is to find a metric with certain properties. For example, given an arbitrary set of DNA sequences, find a tree metric that closely matches the string edit metric over those sequences. This type of problem arises in the context of trying to automatically learn phylogenetic trees from experimental data.

We will see a number of results from each of these classes. For example, there is

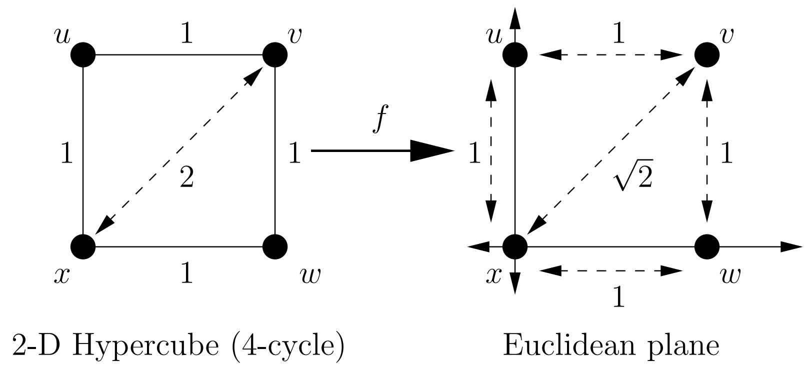

Figure 1: Distortion in embeddings

a strong relationship between embeddings and multi-commodity flow problems that leads to approximation algorithms for the sparsest cut problem. We will also see some interesting hardness of approximation results for a variety of problems on special case metrics which may be proved by embedding a "hard" metric space into the metric space being explored.

In Sections 1.5 and 1.7 we will survey results in these areas in more detail. Unfortunately, however, it is often the case that no isometric embedding from one metric space M_1 into another space M_2 exists. In the next section we will define the concept of an approximate metric space embedding. This will lead to the meat of our topic, a powerful and rich approach to approximation algorithms via metric space embeddings.

1.2.2 Approximate embeddings

Consider embedding the two dimensional hypercube (under Hamming distance) into the Euclidean plane as in Figure 1. Let Δ_H be the Hamming distance function, while $||x||_2$ denotes the Euclidean length of x. Then $\Delta_H(u, w) = \Delta_H(v, x) = 2$ while

$\Delta_H(u,v) = \Delta_H(v,w) = \Delta_H(w,x) = \Delta_H(x,u) = 1$. Under Euclidean distance we should suspect that if $||u-v||_2 = ||v-w||_2 = ||w-x||_2 = ||x-u||_2 = 1$, then either $||u-w||_2 \leq \sqrt{2}$ or $||v-x||_2 \leq \sqrt{2}$. In other words we expect that any embedding of the two dimensional Hamming cube into a Euclidean space would have some distance expanded or contracted by a factor at least $\sqrt{2}$.

In order to rigorously state and prove this fact (in Section 1.2.3), we introduce a few more definitions. First we will define approximate metric space embeddings, in which each pairwise distance between points is allowed to expand or contract by up to some factor D called the distortion.

Definition 1.2.3 (Distortion of an approximate metric space embedding). *Consider two metric spaces $M_1 = (X_1, \Delta_1)$ and $M_2 = (X_2, \Delta_2)$ and an injective map $f : X_1 \to X_2$. The embedding f is said to have distortion $D = \inf D'$ for which*

$$\exists r > 0, \ \textit{s.t.}\ \forall u, v \in X_1, \ \frac{r}{D'}\Delta_1(u,v) \leq \Delta_2(f(u), f(v)) \leq r\Delta_1(u,v).$$

*We will call any embedding with distortion at most D a D-*embedding*.*

This definition (adapted from [56]) allows the embedding to scale distances arbitrarily via r, so we need not make the distinction between distances that expand versus those that contract. Many metrics, in particular the ℓ_p norms for $p \geq 1$, allow arbitrary scaling, so we will often ignore the r in later discussions of distortion.

Definition 1.2.4 (The ℓ_p norm of $\mathbb{R}^d$). *For a point $x \in \mathbb{R}^d$ and $p \in [1, \infty)$, the ℓ_p norm is defined to be (where x_i denotes the ith coordinate of x):*

$$||x||_p = \left(\sum_{i=1}^{d} |x_i|^p\right)^{1/p}$$

The ℓ_p distance between two vectors u and v is $||u-v||_p$, and defines a metric. The metric space $(\ell_p, \mathbb{R}^d)$ is often denoted ℓ_p^d.

The ℓ_p distance functions are well studied: ℓ_2 is just the Euclidean distance, while the ℓ_∞-norm of a vector x is defined to be $\max_i(x_i)$. The ℓ_1 norm, also known as the Manhattan or "taxicab" metric, will be of particular interest in this work. We will use ℓ_p to denote the ℓ_p distance itself or an arbitrary dimensional real vector space equipped with the ℓ_p distance. The meaning should be clear from context. The minimum distortion with which some metric M can be embedded into ℓ_p is denoted $c_p(M)$.

1.2.3 Proving distortion lower bounds

Now let us return to the question of embedding the two dimensional hypercube into ℓ_2. How can we, in general, give a lower bound on the distortion needed to embed a metric M_1 into another metric space M_2? We present the method of [56]. In order to make this idea more comprehensible to a general audience we also supply our own elementary proof of the correctness of this technique.

Consider, for a particular metric M_1, a ratio of two different linear combinations of distances

$$R_{M_1}(p) = \left(\frac{\sum\limits_{u,v \in X_1} \alpha_{uv} \Delta_1(u,v)^p}{\sum\limits_{u,v \in X_1} \beta_{uv} \Delta_1(u,v)^p} \right)^{1/p},$$

with $\alpha_{uv}, \beta_{uv} \geq 0$. We will choose the values of α_{uv} and β_{uv}, and then compare this to the same ratio (for the same p) under any embedding f into the target host metric:

$$R_{M_2}(p) = \left(\frac{\sum\limits_{u,v \in X_1} \alpha_{uv} \Delta_2(f(u), f(v))^p}{\sum\limits_{u,v \in X_1} \beta_{uv} \Delta_2(f(u), f(v))^p} \right)^{1/p}.$$

Theorem 1.2.5. *If there exists $p > 0$, $\alpha_{uv}, \beta_{uv} \geq 0$ such that for any $f : X_1 \to X_2$, $R_{M_1}(p) \geq D' R_{M_2}(p)$, then any embedding of M_1 into M_2 requires distortion at least D'.*

Proof. Taking both sides to the power p and cross-multiplying, we see that

$$\begin{aligned} &\left(\sum_{u,v \in X_1} \alpha_{uv} \Delta_1(u, v)^p \right) \left(\sum_{u,v \in X_1} \beta_{uv} \Delta_2(f(u), f(v))^p \right) \\ &\geq D'^p \left(\sum_{u,v \in X_1} \alpha_{uv} \Delta_2(f(u), f(v))^p \right) \left(\sum_{u,v \in X_1} \beta_{uv} \Delta_1(u, v)^p \right). \end{aligned}$$

Rewriting this as

$$\begin{aligned} &\sum_{u,v \in X_1} \sum_{w,x \in X_1} \alpha_{uv} \beta_{wx} \Delta_1(u, v)^p \Delta_2(f(w), f(x))^p \\ &\geq D'^p \sum_{u,v \in X_1} \sum_{w,x \in X_1} \alpha_{uv} \beta_{wx} \Delta_2(f(u), f(v))^p \Delta_1(w, x)^p, \end{aligned}$$

we see that there exists some particular u, v, w, x such that

$$\alpha_{uv} \beta_{wx} \Delta_1(u, v)^p \Delta_2(f(w), f(x))^p \geq D'^p \alpha_{uv} \beta_{wx} \Delta_2(f(u), f(v))^p \Delta_1(w, x)^p.$$

This fact follows from the convexity of linear combinations. Simplification yields

$$\frac{1}{D'} \frac{\Delta_1(u, v)}{\Delta_2(f(u), f(v))} \geq \frac{\Delta_1(w, x)}{\Delta_2(f(w), f(x))}.$$

Now consider any embedding with distortion $D < D'$ and scale factor r. Combined with the definition of a distortion D embedding, we see that

$$1 \geq \frac{r}{D}\frac{\Delta_1(u,v)}{\Delta_2(f(u),f(v))} > \frac{r}{D'}\frac{\Delta_1(u,v)}{\Delta_2(f(u),f(v))} \geq r\frac{\Delta_1(w,x)}{\Delta_2(f(w),f(x))} \geq 1.$$

This is a contradiction, so the distortion of any embedding is at least D'. $\square$

Notice that Theorem 1.2.5 is symmetric with respect to R_{M_1} and R_{M_2}: If $\forall f : X_1 \to X_2,\ R_{M_1} \geq DR_{M_2}$, then $\forall f : X_1 \to X_2,\ R_{M_2} \geq DR_{M_1}$. This can be proved by switching the roles of α_{uv} and β_{uv}.

Also note that this technique is universal for embedding finite metric spaces into ℓ_p spaces. As Matoušek [56] describes:

Claim 1.2.6. *Let M_1 be a finite metric space and let $D \geq 1$ be the smallest number such that M_1 can be D-embedded into ℓ_p, $p \in [1,\infty)$. Then there are weights $\alpha_{uv}, \beta_{uv} \geq 0$ such that $R_{M_1}(p) \geq DR_{M_2}(p)$ for any metric M_2 resulting from an embedding of M_1 into ℓ_p.*

The proof for the ℓ_2 case is given in [56], while the precise statement and proof for the general ℓ_p case is left as an exercise. This means that there is always a "simple" proof giving the exact lower bound for embedding a finite metric into an ℓ_p space, though note that *finding* the right α_{uv} and β_{uv} can be NP-complete (for $p = 1$, for example).

Now to complete our proof of distortion for embedding the 2-hypercube into ℓ_2^2 we need one more simple lemma from Matoušek [56].

Lemma 1.2.7 (Short diagonals lemma [56]). *Let x, y, z, w be arbitrary points in* *a Euclidean space. Then:*

$$||x-z||_2^2 + ||y-w||_2^2 \leq ||x-y||_2^2 + ||y-z||_2^2 + ||z-w||_2^2 + ||w-x||_2^2.$$

Proof. We present Matoušek's proof from [56] for completeness. Notice that

$$||x-y||_2^2 = \sum_{i=0}^{d} (x_i - y_i)^2.$$

(The number of dimensions can be assumed to be 3 since we are in Euclidean space.) Since the contributions of each dimension can be separated, simply proving the lemma for a line will suffice: We can then add together one such inequality for each dimension. Therefore we calculate

$$-(x-z)^2-(y-w)^2+(x-y)^2+(y-z)^2+(z-w)^2+(w-x)^2 = (x-y+z-w)^2 \geq 0.$$

$\square$

Theorem 1.2.8. *Any embedding of the two dimensional hypercube into ℓ_2 requires distortion at least $\sqrt{2}$.*

Proof. Consider

$$\alpha_{uv} = \begin{cases} 1 & \text{iff } (u,v) \text{ is a diagonal} \\ 0 & \text{otherwise} \end{cases}$$

$$\beta_{uv} = \begin{cases} 0 & \text{iff } (u,v) \text{ is a diagonal} \\ 1 & \text{otherwise} \end{cases}$$

If M_1 is the two dimensional hypercube and $M_2 = \ell_2$, then it is easy to see that, choosing $p = 2$,

$$R_{M_1}(2) = \sqrt{\frac{2^2 + 2^2}{1^2 + 1^2 + 1^2 + 1^2}} = \sqrt{2}.$$

On the other hand, by the short diagonals lemma,

$$||u - w||_2^2 + ||v - x||_2^2 \leq ||u - v||_2^2 + ||v - w||_2^2 + ||w - x||_2^2 + ||x - u||_2^2,$$

which implies that

$$R_{M_2}(2) = \sqrt{\frac{||u - w||_2^2 + ||v - x||_2^2}{||u - v||_2^2 + ||v - w||_2^2 + ||w - x||_2^2 + ||x - u||_2^2}} \leq 1.$$

Hence $R_{M_1} \geq \sqrt{2} R_{M_2}$, and the distortion for embedding the two dimensional hypercube into ℓ_2 is at least $\sqrt{2}$. The embedding given in Figure 1 is therefore optimal. $\square$

Note that this proof does not make use of the fact that we are embedding into two dimensions, and indeed holds for any Euclidean space.

How did we choose the α_{uv} and β_{uv}? Recall that we want our inequality to show that $R_{M_1} \geq D R_{M_2}$. That means that, under the embedding, we want the items in the numerator (those with $\alpha_{uv} \neq 0$) to increase relative to the items in the denominator, so that R_{M_2} is much smaller than R_{M_1}. Intuitively, if we have some idea which distances "want to shrink" under the embedding, these are the distances that should have $\alpha_{uv} > 0$. Similarly, distances that we expect to expand should have large β_{uv} but $\alpha_{uv} = 0$. This is a very rough idea, however, and precisely choosing optimal α_{uv} and β_{uv} is often the key to proving a good bound.

1.3 Optimization

In this work we will make extensive use of optimization techniques. Very broadly, an optimization problem is a problem in n variables where one tries to minimize (or maximize) some objective function subject to some constraint functions. We will introduce two types of optimization problems, linear programs and semidefinite programs, that have intimate relationships with ℓ_1 and ℓ_2 embeddings.

1.3.1 Linear programming

A linear program is an optimization problem in which we minimize a linear objective function subject to linear constraints. Formally, given an $m \times n$ matrix A, an m-vector b and an n-vector c, we want to minimize $c^T x$ (See Table 1). In other words,

minimize:	$c^T x$	
subject to:		
Ax	$\geq$	b
x	$\geq$	0

Table 1: Canonical form of a linear program

A and b together define a set of m linear constraints on n variables, while c gives the linear objective function on n variables. The *dual* of this linear program is a related maximization problem (See Table 2).

Theorem 1.3.1 (Gale, Kuhn and Tucker [27]). *Let OPT_P denote the solution of the primal LP and OPT_D denote the solution of the dual LP. Then if both the original LP and its dual are feasible, $OPT_P = OPT_D$.*

maximize:	$b^T y$	
subject to:		
$A^T y$	$\leq$	c
y	$\geq$	0

Table 2: Canonical form of a dual linear program

This theorem is often referred to as the "Strong Duality" theorem for linear programs because it shows that there is no gap between the solutions of a linear program and its dual. Also note that for *any* y' which is a feasible solution to the dual linear program, $OPT_P \geq b^T y'$. This fact can be used to prove lower bounds. If our problem can be formulated as a linear program, providing any feasible solution to the dual problem proves a lower bound. In addition to using this proof technique in Section 3.1, we will also mention some other relationships between linear programming and ℓ_1 in Section 1.5.3. Linear programs can be solved in polynomial time [44, 42], while such a result is not known for many other natural optimization problems.

1.3.2 Semidefinite programming

We will also have some interest in another type of optimization problem called a semidefinite program. Let $\mathcal{S}$ be the set of $n \times n$ real symmetric matrices. We say that a matrix is positive semidefinite if $\forall x, x^T S x \geq 0$, and this fact is denoted $S \succeq \mathcal{O}$ (where $\mathcal{O}$ is the zero matrix). Now given m constraint matrices $A_1, \ldots, A_m \in \mathcal{S}$, an objective matrix $A_0 \in \mathcal{S}$ and a length m constraint vector c (where $\bullet$ denotes the inner product) we want to minimize $A_0 \bullet X$ (See Table 3). Again, semidefinite programs have a natural dual (See Table 4). In this case the weak duality condition holds, namely that for any feasible x and X, $c^T x \leq A_0 \bullet X$. The conditions for a zero duality gap are

minimize:	$A_0 \bullet X$
subject to:	$A_i \bullet X = c_i, (1 \leq i \leq m)$ $X \in \mathcal{S}, X \succeq \mathcal{O}$

Table 3: Canonical form of a semidefinite program

maximize:	$\sum_{i=1}^{m} c_i x_i$
subject to:	$A_0 = \sum_{i=1}^{m} A_i x_i + Y$ $Y \in \mathcal{S}, Y \succeq \mathcal{O}$

Table 4: Canonical semidefinite program dual

more technical (see Ramana et al. [64]), but for proving lower bounds we will find weak duality to be the most useful. In Section 1.5.3 we will show how distortion-optimal ℓ_2 embeddings may be found via SDPs. Again, it is a key point that the solution of an SDP can be approximated up to an additive factor ϵ in polynomial time (with the running time depending on ϵ). For more information about algorithms for solving SDPs, see Alizadeh [2] and Grötschel et al. [31]. SDPs have found application in both approximation algorithms and combinatorics: For example, see Goemans' and Williamson's seminal paper [30] which used SDP relaxations to approximate MAX CUT, and the work of Lovász and Schrijver [52, 53] on combinatorial optimization.

1.4 Graphs and graph metrics

Many metrics of interest arise from the shortest-path metrics of graphs, so we will need a few graph theoretic ideas before we proceed. For a general reference on graph theory, see the excellent book by Diestel [21].

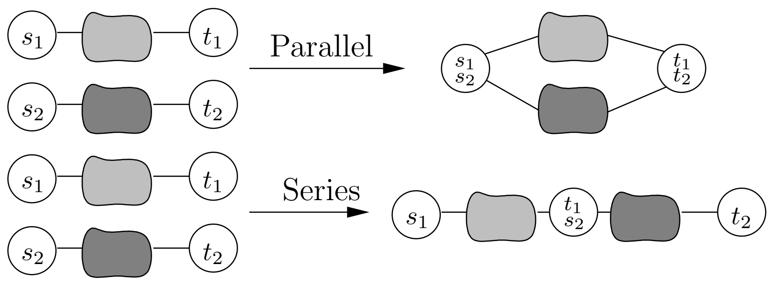

Figure 2: Series-parallel graph construction

Given an undirected graph G with non-negative edge lengths, we will consider the metric of G, where V is the vertex set of G, to be $\Delta_G(v_i, v_j)$ = the length of the shortest path from v_i to v_j in G. We will say that a metric $M = (X, \Delta)$ is supported on an undirected graph G if there exists some weighting of the edges of the graph and $f : X \to V$ such that:

$$\forall u, v \in X,\ \Delta_G(f(u), f(v)) = \Delta(u, v).$$

1.4.1 Some key classes of graphs

We need to introduce or recall the definitions of some important classes of graphs.

Definition 1.4.1 (Tree). *A graph with no cycles.*

Definition 1.4.2 (Outerplanar graph). *Any graph which may be embedded in the plane such that no edges cross and all vertices lie on the outside face.*

Definition 1.4.3 (Series-parallel graph). *Define these recursively (See Figure 2):*

- **Base case:** *An edge (s, t) is a series-parallel graph with source s and target t.*

- **Recursive case:** *Given two series-parallel graphs G_1 and G_2 the following two graphs are also series-parallel:*
  - **Series:** *Set $s = s_1$, $t = t_2$ and $s_2 = t_1$ (that is, merge t_1 and s_2 into a single vertex).*
  - **Parallel:** *Set $s = s_1 = s_2$ and $t = t_1 = t_2$.*

Definition 1.4.4 (Planar graph). *Any graph which may be embedded in the plane such that no edges cross.*

It is known that trees $\subset$ outerplanar graphs $\subset$ series-parallel graphs $\subset$ planar graphs. All of these containments are proper.[2]

Definition 1.4.5 (k-outerplanar graph (taken from [16])). *An outerplanar graph is 1-outerplanar. A graph is k-outerplanar if it can be embedded in the plane so that no edges cross, and when the vertices on the outer face are deleted the result is a $(k-1)$-outerplanar graph.*

This provides a nice class of graphs which interpolate between outerplanar graphs and planar graphs. Notice that in a planar graph with more than 2 points the outside face must contain at least 3 points, so $k \leq \lfloor \frac{n+2}{3} \rfloor$. Chekuri et al. [16] also point out that the class of n-vertex k-outerplanar graphs contains the $k \times \frac{n}{k}$ planar grid.

Definition 1.4.6 (Graph minor). *Given two graphs H and G, G has an H-minor if there is some sequence of edge deletions and edge contractions in G that result in a graph G' isomorphic to H.*

[2] These facts are a simple exercise. Hint: $K_{2,3}$ is series-parallel but not outerplanar; K_4 is planar but not series-parallel.

We say that a graph is H-free if it contains no H-minor. It is known that the class of K_4-free graphs is exactly the class whose blocks are series-parallel [21, p. 185]. The class of $K_{2,3}$-free graphs is exactly the class whose blocks are either outerplanar or isomorphic to K_4 [21, p. 91].

1.5 Previous work

In defining approximate embeddings we demonstrated that there are settings in which an exact embedding of a metric space into a particular normed space does not exist. In order to develop approximation algorithms, however, we do not need isometric embeddings, but only reasonable approximate embeddings. For this reason the properties of approximate embeddings of metric spaces into normed spaces, especially ℓ_2, have been extensively studied. Here we present the key results regarding the distortion of metric embeddings into normed spaces.

1.5.1 Distortion results for ℓ_∞

The case of ℓ_∞ is particularly straight-forward when the number of dimensions used is not constrained.

Theorem 1.5.1 (Fréchet). *Every finite metric $M = (X, \Delta)$ with $|X| = n$ can be isometrically embedded into ℓ_∞^n.*

Proof. (adapted from [39]) Where $X = \{x_1, \dots, x_n\}$, define $f : X \to \ell_\infty^n$ by $f(x_i)_j = \Delta(x_i, x_j)$. This works because $f(x_j)_j = 0$, so $|f(x_i)_j - f(x_j)_j| = \Delta(x_i, x_j)$. Furthermore, for all k, $|f(x_i)_k - f(x_j)_k| \leq |f(x_i)_j - f(x_j)_j|$ because $|\Delta(x_i, x_k) - \Delta(x_j, x_k)| \leq \Delta(x_i, x_j)$ as a result of the fact that Δ obeys the triangle inequality. $\square$

In this sense ℓ_∞ is the most powerful normed space.

1.5.2 Distortion results for ℓ_p, $p < \infty$

We begin with the classical result of Bourgain:

Theorem 1.5.2 (Bourgain [11]). *Any finite metric (X, Δ) can be embedded into ℓ_2^d for some finite d with distortion $O(\log(|X|))$.*

Linial et al. observe [50] that, for any m, ℓ_2^m may be embedded into ℓ_p^{2m} with $1 \le p \le 2$ with constant distortion. This implies that Bourgain's theorem actually holds for all $1 \le p \le 2$. Linial et al. [50] also proved that this bound on distortion is asymptotically tight.

Theorem 1.5.3 ([50]). *For all $1 \le p \le 2$, embedding the shortest-path metric of an n-vertex constant-degree expander graph into an ℓ_p space requires distortion $\Omega(\log(n))$.*

Matoušek extended these results (See [56, p. 345] and [55]) to all ℓ_p:

Theorem 1.5.4. *The minimum distortion required to embed all n-point metric spaces into ℓ_p is $\Theta(\frac{\log n}{p})$.*

Another interesting lower bound is provided by Linial et al. [51]:

Theorem 1.5.5 ([51]). *If G is any k-regular graph with $k \ge 3$ and girth g, then every embedding of G into ℓ_2 has distortion $\Omega(\sqrt{g})$.*

For many problems $\Omega(\log(n))$ distortion is actually far too much. For example, consider a proximity problem defined on a given expander graph G with unit edge weights. It is well known that the diameter of an expander graph with n vertices is $O(\log(n))$. Theorem 1.5.3 implies that if we try to solve such a problem via D-embeddings into $\ell_p, 1 \leq p \leq 2$, we may simply map the vertices of G onto the vertices of the n-point simplex. This still gives $O(\log(n))$ distortion, which is optimal up to constant factors, but at the same time we have removed all the distance information from the metric. Simply embedding into ℓ_p and optimizing for worst-case distortion will not always be useful.

On the other hand, there are numerous special classes of metrics which are of interest and do not have expander properties or high girth. We will present two more lower bounds for embedding commonly studied graphs into ℓ_2, and then move on to some positive results. First, Theorem 1.2.8 can be extended to any m-dimensional hypercube.

Theorem 1.5.6 (Enflo [22]). *Any embedding of the m-dimensional hypercube into ℓ_2 has distortion at least $\sqrt{m}$.*

Proof and discussion can be found in [56], and is attributed to Enflo [22]. Now let us observe that the hypercube on n points is an ℓ_1 metric. This implies a nice corollary:

Corollary 1.5.7. *There are sets of n points in ℓ_1 which require $\sqrt{\log(n)}$ distortion to embed into ℓ_2.*

It is also known that there are even series-parallel graph metrics which embed poorly into ℓ_2.

Theorem 1.5.8 (Newman and Rabinovich [59]). *There is an infinite family of n-vertex series-parallel graphs that cannot be embedded into ℓ_2 with distortion smaller than $\Omega(\sqrt{\log n})$.*

Recall that series-parallel graphs are a (strict) subset of the planar graphs. This lower bound therefore matches the earlier upper bound given by Rao.

Theorem 1.5.9 (Rao [65]). *Any metric supported on a planar graph with n vertices can be embedded in ℓ_2 (and therefore into ℓ_1) with distortion at most $O(\sqrt{\log(n)})$. These results hold for* any *family of graphs which exclude a fixed minor.*

Recall that we say a metric M is supported on a graph G if there is some non-negative weighting of the edges of G for which the resulting shortest-path metric is M.

Theorem 1.5.10 ([32, 20, 17]). *The class of graphs G for which*

$$\max_{M \text{ supp on } G} c_1(M) = 1$$

is exactly the class of graphs which exclude $K_{2,3}$ as a minor.

Conjecture 1.5.11 ([32, 36]). *Given a small fixed graph H, the class of H-minor free graphs embeds into ℓ_1 with constant distortion.*

This class of graphs contains the planar graphs. Any $o(\sqrt{\log n})$ distortion embedding for planar graphs into ℓ_1 would be an exciting result.

1.5.3 Finding optimal embeddings via optimization

Here we will consider the question of how to find an embedding into ℓ_1 or ℓ_2 with optimal distortion.

Definition 1.5.12 (Cut semimetric). *A metric $M = (X, \Delta)$ is a cut semimetric if there exist $X_1 \cup X_2 = X$ such that $X_1 \cap X_2 = \emptyset$ and*

$$\Delta(u,v) = \begin{cases} 0 & \text{if } u,v \in X_1 \\ 0 & \text{if } u,v \in X_2 \\ 1 & \text{otherwise} \end{cases} .$$

In other words, each point is on one side or the other of some cut, and two points have distance one iff they are on opposite sides of the cut. It is easy to see (see [20]) that every ℓ_1 metric is a member of the *cut cone*: Every ℓ_1 metric is a linear combination of cut semimetrics with non-negative coefficients. In fact, the other direction is true as well:

Theorem 1.5.13 (Assouad [7]). *A finite metric Δ embeds in ℓ_1 iff it can be written as a linear combination of cut semimetrics with non-negative coefficients. (In other words, iff it belongs to the cone of cut semimetrics.)*

It is possible to solve for this linear combination using a linear program, but since there are n points, there are 2^{n-1} possible cuts. This leads to a linear program with 2^{n-1} variables. Solving such a problem in general is likely to be infeasible even for relatively small n. In fact, deciding whether or not a given metric is isometrically embeddable into ℓ_1 is known to be NP-complete [39, 43].

Open Problem 1.5.14 ([57]). *Can the optimal distortion for embedding a given metric into ℓ_1 be approximated up to a constant factor in polynomial time? Or, alternatively, can it be shown that approximating the optimal distortion is NP-hard?*

The case for ℓ_2 is a bit nicer:

Theorem 1.5.15 (Linial et al. [50]). *There is a deterministic polynomial-time algorithm that for every constant ϵ embeds $M = (X, \Delta)$ in ℓ_2 with distortion at most $c_2(M) + \epsilon$.*

Proof. ([50]) Let the rows of a matrix B be the images of the points of M under a distortion D embedding into ℓ_2. Let $A = BB^T$. A is positive semidefinite, and for all $u \neq v$:

$$\frac{1}{D^2}\Delta(u,v)^2 \leq a_{uu} - 2a_{uv} + a_{vv} \leq \Delta(u,v)^2.$$

If we set minimizing D as our objective, then this defines a semidefinite program, which can be ϵ-approximated in polynomial time by the ellipsoid method (see Section 1.3.2). A matrix B corresponding to the embedding may then be obtained from the Cholesky factorization of A. □

This method is actually very appealing because finding an optimal embedding of an n-point metric only requires $O(n^2)$ variables and $O(n^2)$ constraints. Semidefinite program solvers, however, are much slower than linear program solvers.

1.5.4 Volume respecting embeddings

For ℓ_2, Feige has defined a useful generalization of D-embeddings called volume respecting embeddings [24]. For any set of points $|P| = k$ in ℓ_2, let $Evol(P)$ denote the volume of the $(k-1)$-dimensional simplex corresponding to P. Note that if the points lie completely in some $(k-2)$-dimensional subspace of ℓ_2 then $Evol(P) = 0$.

Definition 1.5.16 ($Vol(X)$ of a metric M [24]). *Given a metric $M = (X, \Delta)$, $|X| = k$*

$$Vol(X) \stackrel{def}{=} \max_{f \in F} Evol(f(X)),$$

where F is the set of all contractions $f : X \to \ell_2^{k-1}$.

A *contraction* of M_1 is an embedding f into a metric M_2 such that

$$\forall u, v \in X_1,\ \Delta_2(f(u), f(v)) \le \Delta_1(u, v).$$

Note that the embeddings must be restricted to be contractions, otherwise $Vol(X)$ would be infinite in general.

Definition 1.5.17 (Distortion of a contraction [24]). *Given a set X of k vertices from a metric M, the distortion of the contraction f on X is defined to be*

$$\eta(f, X) = \left(\frac{Vol(X)}{Evol(f(X))}\right)^{1/(k-1)}.$$

Note that if $Evol(X) = 0$, we define $\eta(f, X) = \infty$.

Definition 1.5.18 ((k, D)-volume respecting embedding [24]). *An embedding $f : X \to \ell_2^d$ is (k, D)-volume respecting if it is a contraction and*

$$D \ge \max_{P \in X : |P| = k} \eta(f, P).$$

Now we may give the main theorem of Feige [24].

Theorem 1.5.19 (Existence of volume respecting embeddings [24]). *For any connected graph G (equivalently, for any metric with all distances finite) and any $d > \log^2(n)$ there is a randomized embedding into ℓ_2^d which with high probability is $(O(\sqrt{d}/\log(n)), O(\log(n) d^{\frac{1}{4}}))$-volume respecting.*

Feige's embedding is essentially a variation on the embedding of Bourgain [11]. Rao [65] gave improved tradeoffs for some important special cases:

Theorem 1.5.20 (Rao [65]). *Metrics arising from shortest path metrics of planar graphs have* $(k, O(\sqrt{\log(n)}))$*-volume respecting embeddings.*

Euclidean metrics have $(k, O(\sqrt{\log(k)\log(L)})$*-volume respecting embeddings, where* L *is the ratio of the largest distance to the smallest distance in the metric.*

Very recently Krauthgamer et al. [46] have given a very nice improvement of these results.

Theorem 1.5.21 (Krauthgamer et al. [46]). *Any* n*-point metric space* (X, Δ) *can be embedded in Hilbert space with distortion* $O(\sqrt{\log(\lambda_X)\log(n)})$*, where* λ_X *is the doubling constant of* X*.*

They also give $(k, \log(n))$*-volume respecting embeddings for general metrics for every* $1 \leq k \leq n$*, which is the best possible.*

Feige developed these embeddings in order to give an approximation algorithm for graph bandwidth. Detailed discussion of volume-respecting embeddings and their applications is beyond the scope of this document, but readers may look to [24] and [34] for more information.

1.5.5 Probabilistic embeddings into dominating trees

One very tempting approach to algorithmic problems on graphs is to try to embed the metric of a graph H into another graph G. The goal is to pick the graph G from some family of graphs which have nice properties in order to simplify algorithm design. Unfortunately Rabinovich and Raz [63] showed that for many natural problems the distortion incurred by such an embedding is high.

Lemma 1.5.22 (Rabinovich and Raz [63]). *Even if* H *is a simple unweighted connected graph, and* G *is allowed to be any arbitrary weighted graph with the same* *number of vertices but strictly less edges, then the distortion of embedding H into G is at least $g/3 - 1$, where g is the girth of H.*

Consider embedding the n-cycle into any tree on n vertices. This lemma shows that such an embedding incurs distortion $\Omega(n)$. Amazingly, however, this has not been the end of the story for embeddings into trees.

We need two more definitions.

Definition 1.5.23 (Dominating metric). *A metric $M_2 = (X_2, \Delta_2)$ dominates a metric $M_1 = (X_1, \Delta_1)$ if there exists some mapping $f : X_1 \to X_2$ such that:*

$$\forall u, v \in X_1,\ \Delta_2(f(u), f(v)) \geq \Delta_1(u, v).$$

Definition 1.5.24 (Probabilistic embedding). *A probabilistic embedding of a metric $M = (X, \Delta)$ is a probability distribution $\mathcal{S}$ over a set $\{(M_i, f_i : X \to X_i)\}$ of metric spaces M_i with an attached embedding f_i from M to M_i. Such an embedding is said to have distortion D if:*

$$\exists r > 0 \ s.t.\ \forall u, v \in X,\ \frac{r}{D}\Delta(u, v) \leq \mathop{E}_{(M_i, f_i) \in \mathcal{S}}[\Delta_i(f_i(u), f_i(v))] \leq r\Delta(u, v)$$

In other words we are willing to allow the distortion to be high in any particular embedding $M \to M_i$, but the expected distortion should be at most D.

Theorem 1.5.25 (Fakcharoenphol et al. [23, 9]). *Any finite metric on n points can be probabilistically embedded into a set of dominating tree metrics with expected distortion $D = \Theta(\log(n))$.*

This theorem is the result of a considerable amount of work, starting with Alon et al. [3] and their algorithm for the online k-server problem which used a distortion $2^{O(\sqrt{\log(n)\log\log(n)})}$ embedding. In his highly influential paper [9] Bartal reduced this distortion to $O(\log^2(n))$ and gave numerous applications, as well as the $\Omega(\log(n))$ lower bound which is achieved for expander graphs. Fakcharoenphol et al. [23] recently reduced the distortion to $O(\log(n))$, essentially closing the problem for general metrics, and solving the second open question of Indyk [36]. Note that we have not discussed why we require the tree metrics to dominate the original metric. This is a useful property needed for a number of applications, see [9] for more details.

1.6 Dimension reduction

One of our main motivations for studying metric space embeddings has been to find embeddings of difficult metrics into well-studied metrics, such as the ℓ_p-normed spaces. We will now look at another use for embeddings when we are already working with problems in an ℓ_p space.

Given any set X of n points in $\mathbb{R}^d$ under the ℓ_p norm, what is the smallest k (depending on n, p, d and ϵ) such that any X can be embedded into ℓ_p^k with distortion at most $(1+\epsilon)$? This is the fundamental question of dimension reduction in the ℓ_p spaces.

One of the main motivations for studying dimension reduction is the "curse of dimensionality." Many of the computational geometry problems we are interested in, such as nearest neighbor, diameter and many clustering problems, have efficient solutions under the ℓ_p norms when the dimension is a fixed constant. Unfortunately these algorithms seem to require either time or space exponential in the dimension d. If we can reduce the dimensionality of our set of points significantly, at the cost of some distortion, we might be able to design approximation algorithms for geometric problems that do not suffer from the curse of dimensionality. For more information about this technique, see Kushilevitz et al. [47] and the forthcoming chapter by Indyk [38].

1.6.1 Dimension reduction in ℓ_2

First we will introduce one of the most powerful tools for working with metric space embeddings into ℓ_2, the lemma of Johnson and Lindenstrauss [41]. Let us start by observing that, under ℓ_2, the n-point regular simplex requires $n-1$ dimensions to represent exactly. Approximate embeddings, however, have a very different behavior in ℓ_2.

Lemma 1.6.1 (Johnson-Lindenstrauss lemma [41]). *Any set X of n points from ℓ_2 can be embedded into ℓ_2^d with $d = O(\log(n)/\epsilon^2)$ for any $\epsilon > 0$ with distortion at most $(1+\epsilon)$.*

The proof of the original result of Johnson and Lindenstrauss was subsequently simplified by a number of later works: Frankl and Maehara [26], Indyk and Motwani [40], Dasgupta and Gupta [19], Arriaga and Vempala [6] and Achlioptas [1].

We will discuss many applications of this theorem in Section 1.7. Here let us focus on a few other related questions. The reader who is unfamiliar with this work might find this lemma somewhat surprising: For example, how can the n-point simplex with unit edge lengths possibly be well approximated with a small number of dimensions? We think it aids intuition to think about the embedding of Achlioptas [1] and see how it applies to this case. Achlioptas suggests the following operation: Given n vectors of length d represented as an $n \times d$ matrix A, pick a random $d \times k$ matrix R (with

$k = c\log(n)/\epsilon^2$):

$$r_{ij} = \begin{cases} -1/\sqrt{2k} & \text{with probability } \frac{1}{2} \\ +1/\sqrt{2k} & \text{with probability } \frac{1}{2} \end{cases}.$$

Then, with positive probability, AR is an $n \times k$ matrix which corresponds to an embedding of A into ℓ_2^k with distortion at most $(1+\epsilon)$. Now notice that we can represent the n points of the simplex with the vectors

$$\{x : 0^i 10^{n-i-1}, 0 \leq i < n\}.$$

Under Achlioptas' embedding this gives an independent random $\{-1/\sqrt{2k}, 1/\sqrt{2k}\}$ vector of length only $k = c\log(n)/\epsilon^2$ for each point in the simplex. For any such pair of vectors x, y, we expect $|x_i - y_i| = 0$ with probability $\frac{1}{2}$ and $|x_i - y_i| = \frac{2}{\sqrt{2k}}$ with probability $\frac{1}{2}$. Therefore $E[|x_i - y_i|^2] = \frac{1}{k}$, and $E[\sum_{i=0}^{k} |x_i - y_i|^2] = 1$, so each distance between a pair of points is expected to be 1. The key part of the proof in [1] that this embedding works is to show that under such a random projection the distance between any particular pair of points is tightly concentrated around its mean.

The main practical implication of the Johnson-Lindenstrauss lemma is that, if one is working on approximation algorithms for points in ℓ_2, the size of the vectors involved can be reduced to about $O(\log n)$. In these circumstances an exponential dependence on dimension is not out of the question. The storage space needed for n vectors is reduced to about $n\log(n)$ instead of n^2. This has also made possible a variety of very interesting *sub-linear space* algorithms, particularly in the area of streaming algorithms. Again, we will return to these topics in Section 1.7.

1.6.2 Dimension reduction in ℓ_∞

We have seen a randomized polynomial time dimension reduction algorithm for ℓ_2. Is general dimension reduction possible under other ℓ_p norms? Surprisingly little is known about this question.

For ℓ_∞, the work of Bourgain [11] and Matoušek [56, 54] implies that for distortion $D < 3$, $\Omega(n)$ dimensions are required. On the other hand Matoušek also showed closely related positive results:

Theorem 1.6.2 (Matoušek [56, 54]). *Let $D = 2q - 1 \geq 3$ be an odd integer and let (X, Δ) be an n-point metric space. Then there is a distortion D embedding of X into ℓ_∞^d with*

$$d = O(qn^{1/q} \ln n).$$

Krauthgamer et al. [46] have recently shown an embedding for metrics supported on planar graphs into $\ell_\infty^{O(\log(n))}$, which is optimal in terms of the number of dimensions.

1.6.3 Dimension reduction in ℓ_1

Previously very little was known about dimension reduction in ℓ_1. Talagrand [68], following a breakthrough of Schechtman [67] and the refinements of Bourgain, Lindenstrauss and Milman [12] showed:

Theorem 1.6.3 (Talagrand [68]). *Every n-point ℓ_1 metric can be embedded into ℓ_1^d with distortion $(1+\epsilon)$ and $d = Cn\log(n)/\epsilon^2$, where C is some universal constant.*

In contrast to this result, the best lower bound known previously was $\Omega(\log(n))$, which can be shown for the $\log_2(n)$-dimensional hypercube:

Theorem 1.6.4. *There exists some constant $\epsilon_0 > 0$ such that for $0 < \epsilon \leq \epsilon_0$, the n points $\{0,1\}^{\log_2(n)}$ (the $\log_2(n)$-dimensional hypercube) require $\Omega(\log_2(n))$ dimensions to embed into ℓ_1 with distortion $(1+\epsilon)$.*

To prove this we will use a simple lemma:

Lemma 1.6.5. *Consider any finite metric $M_1 = (X \in \mathbb{R}^m, \ell_1)$. Then $M_2 = (X, \ell_2)$ is an embedding of M_1 into ℓ_2^m with distortion at most $\sqrt{m}$.*

Proof. Consider any vector $x = (x_1, \ldots, x_m)$.

$$\sqrt{\sum_m |x_i|^2} \leq \sum_m |x_i| \leq \sqrt{m}\sqrt{\sum_m |x_i|^2}.$$

□

The result of Enflo [22] (Theorem 1.5.6) that the hypercube requires distortion $\Omega(\sqrt{\log_2(n)})$ in ℓ_2 implies the theorem. For if we had a $(1+\epsilon)$ distortion embedding into ℓ_1^m this would imply a $(1+\epsilon)\sqrt{m}$-distortion embedding into ℓ_2. Hence, a $(1+\epsilon)$ distortion embedding of the hypercube with dimension $m = o(\log_2(n))$ is not possible. Note that the constant hidden by the Ω in Enflo's result is at least 1.

There has been substantial interest in trying to prove an analogue of the Johnson-Lindenstrauss lemma for ℓ_1 (for example, see Indyk [36], Linial [49] and the Haifa open problems list [57] maintained by Matoušek). One line of attack has been to consider the dimensionality needed for certain special case metrics when embedded into ℓ_1. Chepoi and Fichet [17] have shown that all circular decomposable (Kalmanson) metrics embed into ℓ_1 isometrically, implying that all metrics supported on outer-planar graphs embed isometrically in ℓ_1. Gupta et al. [32] point out that this in turn

	Distortion, any d	Distortion D, dim d
Trees	1	$D = (1+\epsilon)$, $d = O(\frac{1}{\epsilon^2}\log^3(n))$ [15]
Circular decomposable	1 [17]	$D = (3+\epsilon)$, $d = O(\frac{1}{\epsilon^2}\log^2(n))$ [15]
$K_{2,3}$	1 [17, 32]	$D = O(1)$, $d = O(\log^2(n))$ [15]
Series-parallel graphs	$O(1)$ [32]	

Table 5: Distortion and dimension results for metrics on graphs

implies Theorem 1.5.10, because it is known that $K_{2,3}$ (with all edges weighted one) does not embed isometrically in ℓ_1 [20]. They then go on to study a slightly richer class of graphs, the series-parallel graphs.

Theorem 1.6.6 (Gupta et al. [32]). *Every metric supported on a series-parallel graph (or on a graph with a constant number of non-tree edges $|E(G)| - |V(G)| + 1$) can be embedded into ℓ_1 with constant distortion.*

Charikar and Sahai [15] initiated a study of the trade-off between distortion and dimension for these graphs with constant distortion ℓ_1 embeddings. Table 1.6.3 lists their results along with the results of Gupta et al. and Chepoi and Fichet.

Another direction has been to weaken the requirements of the embedding instead of restricting the original metric. Indyk [35] has shown a "weak" dimension reduction theorem for ℓ_p with $1 \le p \le 2$.

Lemma 1.6.7 ([36, 35]). *For any $1 \le p \le 2$, $1 > \epsilon, \delta > 0$ there is a randomized embedding from ℓ_p^d into $\ell_p^{d'}$ with distortion $(1+\epsilon)$ and $d' = (\log(1/\delta))^{O(1/\epsilon)}$ such that:*

- *The probability that a particular distance contracts by a factor worse than $(1-\epsilon)$ is less than δ*

- *The probability that a particular distance expands by a factor worse than* $(1+\epsilon)$ *is at most* $\frac{1}{1+\epsilon}$.

Indyk notes that this differs from the Johnson-Lindenstrauss lemma in two key ways. One is that the dependence on $(1/\epsilon)$ is now exponential, not polynomial. Secondly, the probability that some distance is expanded a large amount is only bounded away from 1 by a tiny amount. He calls this an "asymmetric" embedding because it prevents contraction, but allows distances to be expanded significantly. In fact, for the ℓ_1 version of this embedding the expectation (over f) of $||f(u) - f(v)||_1$ for any pair of points u, v such that $||u - v||_1 > 0$ is unbounded. Fortunately, though the mean is unbounded, the "left" tail of the distribution can be shown to be small, and this property is sufficient for many applications.

On the other hand, let $f()_i$ denote the ith dimension of the embedded point under Indyk's embedding, and let the embedding have $d' = c\log(1/\delta)/\epsilon^2$ dimensions (where c is some constant we pick later). Define a new distance function $\Delta'(f(u), f(v)) = \text{median}_i|f(u)_i - f(v)_i|$. He shows that

$$(1-\epsilon)||u-v||_1 \leq \Delta'(f(u), f(v)) \leq (1+\epsilon)||u-v||_1$$

with probability at least $1-\delta$. Unfortunately this does not give an embedding into a metric space because the median operation does not give a metric. In particular notice that the three vectors $(0,0,0)$, $(0,0,c)$ and $(0,c,c)$ break the triangle inequality under this component-wise median function (for any $c \neq 0$).

The main technical tools of this work, the p-stable distributions, are of independent interest, and Indyk [35] discusses many aspects of their use, both technical and practical. Several works on small space algorithms have been able to make use of these techniques in spite of the fact that variables from the 1-stable Cauchy distribution have unbounded mean (see [35, 36, 29]).

1.6.4 Dimension reduction in other ℓ_p norms

Charikar and Sahai [15] show the following lower bound for dimension reduction for ℓ_p with $4 \leq p \leq c\log(n)$ for some small constant c.

Theorem 1.6.8 ([15]). *For $4 \leq p \leq c\log(n)$, there exists a set of $2n$ points in ℓ_p^n such that any $(1+\epsilon)$ distortion embedding into ℓ_p^d requires the number of dimensions d to be at least*

$$\begin{cases} \Omega\left(\frac{4^p \log(n)}{p(1+\epsilon)^{2p}}\right) & \epsilon \geq \frac{1}{p} \\ \Omega\left(\frac{4^p \log(n)}{\epsilon^2 p^2 (p+\log(1/\epsilon))}\right) & \epsilon \leq \frac{1}{p} \end{cases}$$

In other words, the number of dimensions needed to approximate a metric of n points in ℓ_p goes up roughly exponentially in p.

1.6.5 Dimension reduction for Hamming metrics

Kushilevitz et al. [47] have also shown a dimension reduction of a different type that maps a Hamming space into a low dimensional Hamming space. Indyk [36] defines a randomized threshold (r_1, r_2, r_1', r_2')-embedding with failure probability p to be an embedding $f : X_1 \to X_2$ such that for any pair of points $u, v \in X_1$ the two conditions

- if $\Delta_1(u,v) \leq r_1$ then $\Delta_2(f(u), f(v)) \leq r_1'$
- if $\Delta_1(u,v) \geq r_2$ then $\Delta_2(f(u), f(v)) \geq r_2'$

are satisfied with probability at least $1-p$. In this setting, Indyk formulates the embedding of Kushilevitz et al. as follows:

Lemma 1.6.9 (Kushilevitz et al. [47, 36]). *For any* $r \in [1, d]$ *and* $\epsilon > 0$ *there exists* $r' \geq 0$ *such that there is a randomized* $(r, (1+\epsilon)r, r', r'+1)$*-embedding from a* d*-dimensional Hamming metric into a* d'*-dimensional Hamming metric with failure probability* p *and* $d' = O(\log(1/p)/\epsilon^2)$.

Essentially this algorithm can distinguish between distances that are less than r and distances that are greater than $(1+\epsilon)r - 1$. This does not, a priori, give an algorithm for calculating distances between points. It seems for this purpose you would need to repeat the embedding for many different values of r. Further discussion may be found in [47, 34, 36].

1.7 Applications of embeddings

Metric embeddings have provided a very nice framework in which to study a large number of algorithmic questions, particularly in approximation algorithms. Here we survey a number of the most interesting results. Much of this discussion is adapted from Indyk's tutorial [36], and we refer the reader to this work for a more broad survey.

1.7.1 Sparsest cuts and multi-commodity flows

One of the first explicit uses of embeddings in the design of approximation algorithms was by Linial et al. [50] and Aumann and Rabani [8].[3] Both of these papers explored the connection between multi-commodity maxflows, sparsest cuts, and ℓ_1 embeddings.

[3] We have chosen to follow the exposition of [50] because we already draw on this paper extensively in this section.

First let us define a multi-commodity maxflow. Let G be an undirected graph with capacities C_{ij} for each edge (i, j). There are k source-sink pairs (s_l, t_l) which want to ship some quantity $d_l \geq 0$ (the demand) of a commodity from s_l to t_l. This is called a multi-commodity flow because each source-sink pair has a different commodity, so all k flows are independent of the others except for the fact that they must share the capacities on the edges. The *maxflow* λ is the largest λ for which every flow may send at least λd_l.

Let $Cap(S)$ be the total capacity of the edges crossing some cut S, and $Dem(S)$ be the total amount of demand from flows with the source and sink on opposite sides of the cut. Then, for every cut S in the graph, $\lambda \leq \frac{Cap(S)}{Dem(S)}$. In fact, the optimal λ is exactly

$$\lambda = \min \frac{\sum\limits_{i \neq j} C_{ij} \Delta(i, j)}{\sum\limits_{l=1}^{k} d_l \Delta(s_l, t_l)},$$

where the minimum is over all metrics Δ supported on G.

The sparsest-cut problem is to find a set S which minimizes $\frac{Cap(S)}{Dem(S)}$, and is known to be NP-complete. The novel contribution of Linial et al. and Aumann and Rabani was, using Bourgain's result [11], to argue that one can embed the metric Δ into ℓ_1 with distortion at most $O(\log k)$. They then use the embedding to find a cut which can be proved to be at most a factor $O(\log(k))$ worse than the sparsest cut. This gives an $O(\log k)$-approximation algorithm for the sparsest-cut problem.

In proving this they showed that the max-flow min-cut gap for a graph G is bounded by the least distortion with which the related metric Δ can be embedded into ℓ_1. In the terminology of Theorem 1.5.10 from Gupta et al.:

Theorem 1.7.1 ([32, 50]). *For any graph G, the worst possible min-cut max-flow* *gap γ attained by a multi-commodity max-flow problem on G is exactly*

$$\gamma = \max_{M \text{ supp on } G} c_1(M).$$

One direction of this theorem appeared in [50] and the theorem was stated and the other direction proved in [32]. The results of Gupta et al. [32] and Chepoi and Fichet [17] (see Table 1.6.3) therefore imply $O(1)$ approximations for sparsest-cut problems on trees, circular decomposable metrics and series-parallel graphs. The result of Rao [65] (Theorem 1.5.9) implies an $O(\sqrt{\log(n)})$ approximation algorithm for sparsest-cut problems on planar graphs.

A positive resolution to Conjecture 1.5.11 would also imply that planar graphs embed into ℓ_1 with constant distortion, and therefore the min-cut max-flow gap for planar graph metrics would be $O(1)$. Also note that Klein et al. [45] have already shown that the min-cut max-flow gap for the case of *uniform* demands on a planar graph is $O(1)$. Aside from [17] and [32], Chekuri et al. [16] have contributed to progress toward this conjecture by showing that, for any fixed constant k, all k-outerplanar graphs embed into ℓ_1 with constant distortion.

1.7.2 $(1, 2) - B$ metrics and applications

Metric embeddings have even found applications in proving hardness of approximation. Here we consider one particular problem, $MinTSP$ in an $\ell_p^{\log(n)}$ space.

A metric is called a $(1, 2)-B$ metric if every distance in the metric is either 1 or 2, and for any u the number of v for which $\Delta(u, v) = 1$ is at most B. Papadimitriou and Yannakakis [61] show that there is a constant $B_0 > 0$ such that computing $MinTSP$ on $(1, 2)-B_0$ metrics is $Max-SNP$ hard. Trevisan [69] has used this result combined with metric space embeddings to show that $MinTSP$ is $Max - SNP$ hard even in $\ell_p^{c\log(n)}$.

Theorem 1.7.2 ($(1,2)-B_0$ metrics "embed" into Hamming space [69]). *Any $(1,2)-B_0$ metric can be embedded into an $O(\log(n))$-dimensional Hamming space so that if $\Delta(u,v)=1$ and $\Delta(s,t)=2$, $\Delta(u,v)\approx\frac{B_0}{B_0+1}\Delta(s,t)$. (The precise statement of the theorem is somewhat technical, see [69].)*

He then shows that such an embedding may be used to distinguish between TSP tours of length n and length $r_0 n$ for some $r_0 > 1$. Since this second problem is $Max - SNP$ hard on $(1,2)-B_0$ metrics, $MinTSP$ on Hamming spaces with $\log(n)$ dimensions must also be $Max - SNP$ hard. Finally, observe that for $u, v \in \{0,1\}^k$, where Δ_H is the Hamming metric, $||u-v||_p = (\Delta_H(u,v))^{1/p}$. Trevisan's result therefore implies $Max - SNP$ hardness for $MinTSP$ in any $\ell_p^{\log(n)}$ space.

Guruswami and Indyk [33] recently showed that $(1,2)-B$ metrics may also be isometrically embedded into ℓ_∞ with $O(B\log(n))$ dimensions by a different technique. They then use this embedding to show hardness of approximation results for TSP, k-median and min-sum k-clustering, even in low dimensional ℓ_p spaces. The reader should refer to [33] for a precise statement of their results.

1.7.3 Probabilistic embeddings into dominating trees

Though the topic at hand is the study of embeddings into normed spaces, especially ℓ_1, we should also briefly mention some of the applications of probabilistic embeddings into trees (see Section 1.5.5). Often designing an optimal algorithm for a tree is significantly easier than designing such an algorithm for a general metric. For that matter, there are many problems which, though NP-complete in general, are poly-time solvable on trees.

The first use of these probabilistic embeddings into trees was by Alon et al. [3] to give an online algorithm for the k-server problem. Recall that online algorithms are often randomized, and are typically analyzed by comparing the expected value of the algorithm's solution $A(P)$ with the optimal (offline) solution $OPT(P)$. The algorithm designer's goal (for a minimization problem) is to minimize the worst case (over problem instances P) ratio of $\frac{E[A(P)]}{OPT(P)}$. Since we are working with expected values, we can often take our problem on metric M, choose a random embedding (M_i, f_i) according to a good probabilistic embedding into trees, and then solve the problem on the tree M_i. After Bartal improved the embedding of Alon et al., he and his co-authors [10] were able to show the first known polylog(n)-competitive online algorithm for the metrical task system problem. These techniques have also led to many new approximation algorithms. See [36] for a list and citations.

1.8 Applications of dimension reduction

The question of whether or not dimension reduction exists for ℓ_1 will be the focus of much of this work. The technique of dimension reduction, particularly in ℓ_2, has been an extremely fruitful tool for algorithm designers. In order to motivate our interest in ℓ_1 dimension reduction we will survey some of the surprising and powerful algorithms which result from the Johnson-Lindenstrauss lemma.

1.8.1 Geometric proximity problems

We now return to the proximity problems which we mentioned as one of our main motivations at the beginning of the chapter. Let us formally define a proximity problem in the following way: A proximity problem is a problem on some set of points X, the solution of which depends only on the distances between points, $\Delta(u, v)$, and not on the actual "locations" of the points.

One of the most studied proximity problems has been the approximate near neighbor problem.

Definition 1.8.1 (Approximate Near Neighbors, $c-ANN$). *Given a set $|P| = n$ of points from some metric space M and an approximation factor c, create a data structure that can answer the following type of query:*

Given a new point x from M, if the distance from x to its nearest neighbor in P is d, return any point in P which is within a distance cd of x.

The goal is to bound both the pre-processing time and the query time needed for this procedure. Indyk and Motwani [40] and Kushilevitz et al. [47] showed the first algorithms for $(1+\epsilon) - ANN$ under ℓ_1 and ℓ_2 which achieved polynomial pre-processing time along with query time that is polynomial in $\log(n)$ and d.[4] Previous algorithms had suffered from the "curse of dimensionality," that either running time or pre-processing time was *exponential* in d (in many cases, $O(1/\epsilon^d)$). Both of these papers use random projections to reduce dimension, and in particular use the Johnson-Lindenstrauss lemma for the ℓ_2 case.

In [34] Indyk went on to give a variety of algorithms for other proximity problems.

[4] Note however that the pre-processing time depends exponentially on $1/\epsilon$, so this algorithm is unlikely to be useful in practice.

Theorem 1.8.2 (Indyk's proximity problems [34]).

- *There is a Las Vegas randomized algorithm for* $(1+\epsilon)$ *near neighbors under* ℓ_1^d *with storage and query time polynomial in* $\log(n)$*,* d *and* $1/\epsilon$*.*

- *There is a deterministic algorithm for* $(1+\epsilon)$ *near neighbors for* $\epsilon > 2$ *under* ℓ_1^d *with similar bounds.*

- *Both of these results hold for* ℓ_2^d *with an additional approximation factor of* $\sqrt{3}$*.*

- *There is an algorithm for* $(1+\epsilon)$ *approximating the diameter of a set of points in Hamming space with running time* $O(n^{2-O(\epsilon)})$*.*

All of these algorithms rely on a dimension reduction algorithm for Hamming metrics which reduces dimension to $O(\log(n))$ while preserving the gap between "long" distances and "short" distances.

Indyk [37] has also given an asymmetric embedding of ℓ_2 into ℓ_∞ which results in a $(1+\epsilon)$ approximation for diameter in ℓ_2 in time $\tilde{O}(n^{1+1/(1+\epsilon)^2})$. One interesting result of this work is that he shows that any ℓ_2 metric can be embedded in $\ell_\infty^{n^\alpha}$ $(1 > \alpha > 0)$ with distortion $\sqrt{2/\alpha + 1}$, which improves upon the result of Matoušek, Theorem 1.6.2, for the special case of ℓ_2 metrics.

1.8.2 Learning

There are a number of geometric learning problems which have been solved using dimension reduction. The first type are continuous clustering problems:

Definition 1.8.3 (Continuous clustering (see [36])). *Given a set of points* $X \in \mathbb{R}^d$*, find* k *points* $\{c_1, \dots, c_k\} \in \mathbb{R}^d$ *that minimize some objective function.*

- **k-medians**: $\sum_{x \in X} \min_i ||x - c_i||_2$
- **k-center**: $\max_{x \in X} \min_i ||x - c_i||_2$

The key difficulty in this definition is that the points c_i need not be points from the set X, but may be any arbitrary point in the space $\mathbb{R}^d$. Dasgupta [18] and Arora and Kannan [5] have shown that, if the set of points $X \in \mathbb{R}^d$ is randomly generated by some set of Gaussian distributions, then the original Gaussians can be approximately learned in time polynomial in d and the number of different Gaussians in the mixture, k. Several of the algorithms proposed by these authors first project points into a random subspace of size polylog(k). Since a linear projection of a Gaussian is still a Gaussian, the authors can identify candidate Gaussians efficiently in the low dimensional space, and then try to lift these results back to $\mathbb{R}^d$.

Ostrovsky and Rabani [60] gave a similar polynomial time approximation scheme for the 2-center clustering problem under ℓ_1, ℓ_2, $(\ell_2)^2$ and Hamming distance. They first embed their set of points into the Hamming cube, and then they use the dimension reduction theorem for the Hamming cube from their paper with Kushilevitz [47]. The key idea is that in a Hamming cube with only $O(\log(n))$ dimensions one can actually test *every* possible pair of centers in time poly(n, d). This result has recently been improved by Fernandez de la Vega et al. [25] using different techniques.

Arriaga and Vempala [6] have also used dimension reduction to improve the sample complexity of the Perceptron Algorithm [58] when learning *robust* concepts. Roughly speaking an l-robust concept is a concept where the probability of getting a sample point that is within a distance l of the boundary is zero. Note here that we assume all points lie within the n-dimensional unit ball, and that $l \leq 1$. Arriaga and Vempala showed that robust concepts can still be learned when projected into a small number of dimensions (usually poly$(\log(n), 1/l)$). This leads to algorithms based on the Perceptron Algorithm which use roughly poly$(1/l)$ samples and run in time poly$(n, 1/l)$. They give algorithms for learning half-spaces, intersections of half-spaces and polynomial surfaces. Learning a half-space in $\mathbb{R}^n$ via the Perceptron Algorithm uses a number of samples linear in n, so this is a significant improvement.

1.8.3 Proximity-preserving labeling

Another very closely related topic is the problem of distance labeling or proximity-preserving labeling in graphs. This problem, introduced by Peleg [62] asks the following:

Definition 1.8.4 ((D, d) proximity preserving labeling scheme). *First, for a given class of graphs $\mathcal{G}$ give a function f which for any graph $G = (V, E) \in \mathcal{G}$ maps V into $\{0, 1\}^d$. This provides a binary label for each vertex.*

Second, give a function $\Delta : \{0, 1\}^d \times \{0, 1\}^d \to [0, \infty)$ which is poly-time computable such that:

$$\exists r > 0, \ s.t. \ \forall u, v \in V, \ \frac{r}{D}\Delta_G(u, v) \leq \Delta(f(u), f(v)) \leq r\Delta_G(u, v).$$

Δ_G is defined to be the shortest path distance on G. The labeling has distortion at most D. A labeling scheme with d bits per label and distortion at most D is called a (D, d) proximity preserving labeling scheme *for the graph class $\mathcal{G}$.*

The goal is to minimize d, the number of bits needed for each label. Notice that this differs from our discussion of embeddings into normed spaces because here we must count all our bits, not just the number of reals used. We will also allow the distance function Δ to be *any* poly-time computable function. There is still a very interesting relationship between these labelings and metric space embeddings.

Theorem 1.8.5 ([62, 36, 56]). *There exists an* $(O(\log(n)), O(\log^3(n)))$ *labeling scheme for general graphs and a* $(1, O(\log^2(n)))$ *labeling scheme for trees.*

The connection to low-dimensional ℓ_p embeddings here is that the first part of this result may be derived from Matoušek's Theorem 1.6.2. The second half of the theorem can be proved using the proof of Linial et al. [50] that trees embed isometrically into $\ell_\infty^{O(\log(n))}$. It is interesting that Peleg's [62] original proofs of these facts did not explicitly use ℓ_∞ embeddings. In his subsequent work with Gavoille et al. [28] it was showed that:

Theorem 1.8.6 (Gavoille at al. [28]). *For* $(1, d)$ *labeling schemes (that is, exact labeling schemes):*

- *For general graphs* $d = O(n)$
- *For trees,* $d = \Theta(\log^2 n)$
- *For planar graphs,* $d = O(\sqrt{n}\log(n))$ *and* $d = \Omega(n^{1/3})$

For (D, d) *labeling schemes:*

- *For general graphs and* $D < 2$, $d = \Omega(n)$
- *For general graphs and* $D < 3$ *the* total length *of all labels is* $\Omega(n^2)$.

It is not clear whether the relationship between labelings and ℓ_∞ embeddings will hold for more interesting examples. Krauthgamer et al. [46] have recently shown that any metric supported on an n-vertex planar graph can be embedded into $\ell_\infty^{O(\log(n))}$

with constant distortion. The above bound for labeling schemes for planar graphs, however, only applies to isometric labelings. The proof proceeds by constructing a large class of graphs which must have different sets of labels, and this gives a lower bound on the number of bits needed to represent such a graph.

In general it is not clear whether these lower bounds can be applied to dimension reduction questions. Since labelings are embeddings into a bounded number of bits, they may not have the full power of ℓ_p at their disposal. Badoiu and Indyk [36] indicate, however, that dimension lower bounds for labeling schemes on graphs with integral distances hold for ℓ_∞.

1.9 Our results

We provide a major milestone in the study of the dimensionality of ℓ_1-embeddable metrics.

Theorem 1.9.1 (Some finite ℓ_1 metrics require many dimensions [13]). *There exists an infinite family of n-point sets in ℓ_1 which require $d = n^{\Omega(1/D^2)}$ dimensions for any embedding into ℓ_1^d with distortion at most a constant D. When $D = (1+\epsilon)$ with ϵ close enough to zero, this bound is $n^{\frac{1}{2}-O(\epsilon\log(1/\epsilon))}$.*

The key point is that for any constant distortion the number of dimensions required is polynomial in n, a stark contrast to the case of ℓ_2 (see above, Section 1.6.1). In Section 3.2 we provide an explicit set of n points in $\ell_1^{O(\sqrt{n})}$ which requires high dimension. This answers the third open question of Indyk [36] in the negative.

Our point set is actually derived from the shortest-path metrics of a family of series-parallel graphs. In order to prove Theorem 1.9.1, we will first prove that some families of series-parallel graphs require high dimension in ℓ_1.

Theorem 1.9.2 (Series-parallel graph metrics require high ℓ_1 dimension [13]). *There exists an infinite family of series-parallel graphs G_n on n vertices such that any embedding of G_n into ℓ_1 with distortion at most a constant D requires $d = n^{\Omega(1/D^2)}$ dimensions. Again, when $D = (1+\epsilon)$ with ϵ close enough to zero, this is asymptotically $n^{\frac{1}{2}-O(\epsilon\log(1/\epsilon))}$.*

1.10 Organization

In Chapter 2 we will introduce the notion of the stretch of an embedding, and discuss how it may be used to prove upper and lower bounds on ℓ_1 dimension. In Section 3.1 we will prove Theorem 1.9.2. Then in Section 3.2 we prove Theorem 1.9.1 by giving an explicit infinite family of $\ell_1^{O(\sqrt{n})}$ metrics which require high dimension in ℓ_1. In Chapter 4 we will discuss in detail the use of optimization techniques in proving our bounds, and show some connections to other problems.

Chapter 2

Stretch-limited Embeddings

We wish to prove, in Section 3.1, that there are some classes of ℓ_1-embeddable metrics which require a high number of dimensions if the incurred distortion is to be small. How can we try to prove such a result? Consider the technique given in Section 1.2.3. In this approach we consider two ratios of the form

$$R_{M_1}(p) = \left(\frac{\sum_{u,v \in X_1} \alpha_{uv} \Delta_1(u,v)^p}{\sum_{u,v \in X_1} \beta_{uv} \Delta_1(u,v)^p} \right)^{1/p}.$$

Consider the ℓ_1 distance in d dimensions, $\sum_{i=1}^{d} |u_i - v_i|$. Increasing the number of dimensions to $d+1$ corresponds to adding a new variable u_{d+1} for each point u in the metric.

In fact, we may think of an embedding of M into ℓ_1^d as a sum of d embeddings of M into the real line. Adding a new dimension corresponds to allowing ourselves an extra line embedding in this sum.

Essentially we want to optimize the α_{uv} and β_{uv} to show a large gap between R_{M_1} and R_{M_2}, *in terms of* d, the number of variables allowed per point u. We do not know of any way to reason about this type of problem directly.

We need to find some property of ℓ_1 metrics which can stand in as a *proxy* for dimensionality. For this purpose Charikar and Sahai [15] introduced the notion of a *stretch-limited* embedding. We will first give our definition of a stretch-limited embedding, and then prove that it is a good proxy for ℓ_1 dimensionality. Our presentation is somewhat different from that of [15].

Definition 2.0.1 (Stretch-limited embedding). *A stretch-limited embedding $\sigma = (f_\sigma, \Delta_\sigma)$ of a metric $M = (X, \Delta)$ consists of a mapping $f_\sigma : X \to \mathbb{R}^t$* along with *a distance function Δ_σ. We will denote the ith dimension of u under f_σ as $f_\sigma(u)_i$, and we can view each $f_\sigma(\cdot)_i$ as a mapping onto the real line. Weights $\{w_1, w_2, \ldots, w_t\}$ are assigned to each line embedding $f_\sigma(\cdot)_i$ such that $\sum_{i=1}^t w_i = 1$, and Δ_σ is defined to be the weighted average of distances under the $f_\sigma(\cdot)_i$:*

$$\Delta_\sigma(f_\sigma(u), f_\sigma(v)) \stackrel{def}{=} \sum_{i=1}^{t} w_i |f_\sigma(u)_i - f_\sigma(v)_i|.$$

The distortion of this embedding is defined to be $D = \inf D'$ such that:

$$\exists r > 0,\ \forall u, v \in X,\ \frac{r}{D'}\Delta(u,v) \le \Delta_\sigma(f_\sigma(u), f_\sigma(v)) \le r\Delta(u,v).$$

If for all points u and v in the original metric, and $\forall i \in \{1, \ldots, t\}$,

$$|f_\sigma(u)_i - f_\sigma(v)_i| \le rs \cdot \Delta(u,v),$$

the embedding is said to be stretch s limited.

In other words, a stretch-s embedding is a convex combination of line metrics where distances in any line metric can not be more than a factor s larger than distances in the *original* metric. We will assume, without loss of generality, that $r = 1$, implying that a stretch-limited embedding is always a contraction. Scaling a stretch-limited embedding affects neither the distortion nor the stretch incurred.

Note that we may alternately define stretch (call it stretch-2) by replacing the stretch condition by: $|f_\sigma(u)_i - f_\sigma(v)_i| \leq rs \cdot \Delta_\sigma(f(u), f(v))$. This alternate definition makes the notion of stretch-2 a function of the *resulting* metric independent of the original metric. The difference between these two notions of stretch is exactly a factor D, the distortion of the embedding. The original definition of stretch seems easier to work with for proving lower bounds; any lower bounds we prove also hold for stretch-2 up to the factor D.

2.1 Stretch as a proxy for ℓ_1 dimension

Claim 2.1.1. *The existence of a D-distortion embedding of a metric $M = (X, \Delta)$ into ℓ_1^s implies the existence of a D-distortion stretch-limited embedding of M with stretch s.*

Proof. Consider a mapping g of M into ℓ_1^s. Let $g(\cdot)_i$ denote the ith dimension under g, which we think of as an embedding into a line. Let $\sigma = (f_\sigma, \Delta_\sigma)$ be the stretch limited embedding defined by $f_\sigma(u)_i = sg(u)_i$, and $w_i = \frac{1}{s}$. Then:

$$\begin{aligned}
\Delta_\sigma(f_\sigma(u), f_\sigma(v)) &= \sum_{i=1}^{s} \frac{1}{s} |f_\sigma(u)_i - f_\sigma(v)_i| \\
&= \sum_{i=1}^{s} \frac{1}{s} \left(s|g(u)_i - g(v)_i|\right) \\
&= ||g(u) - g(v)||_1.
\end{aligned}$$

Since distances are identical under σ and g, their distortions must be equal. $\square$

Claim 2.1.2. *The existence of a D-distortion stretch-limited embedding of a metric $M = (X, \Delta)$ with stretch s implies the existence of a $D(1+\epsilon)$-distortion embedding of M into $\ell_1^{O(sD\log(n)/\epsilon^2)}$.*

Proof. Consider a stretch-s embedding $\sigma = (f_\sigma, \Delta_\sigma)$ as a probability distribution on line metrics, where each line metric $f_\sigma(\cdot)_i$ has probability w_i. For each of the m dimensions we will take a random line metric $f_\sigma(\cdot)_l$ from this distribution, and let the value of x in this dimension be $f_\sigma(x)_l/m$.

Consider the distance of a particular pair of points u and v in a random $f_\sigma(\cdot)_i$ where i is picked with probability w_i. For convenience of notation we will call this distribution $\mathcal{I}$, and I denotes a random index chosen from this distribution. The expected distance is exactly the distance between u and v in the stretch-limited embedding which in turn is in the range $[\Delta(u,v)/D, \Delta(u,v)]$. The stretch condition imposes a bound on the value of $|f_\sigma(u)_i - f_\sigma(v)_i|$, namely that

$$\forall i: \ |f_\sigma(u)_i - f_\sigma(v)_i| \le s\Delta(u,v).$$

Now we can use standard Chernoff-Hoeffding bounds:

$$Pr\left[\left(\sum_{j=1}^{m} |f_\sigma(u)_{I_j} - f_\sigma(v)_{I_j}|\right)\frac{1}{s\Delta(u,v)} \ge (1+\epsilon)\frac{m}{s\Delta(u,v)} E_{I\in\mathcal{I}}[|f_\sigma(u)_I - f_\sigma(v)_I|]\right]$$

$$< \left(\frac{e^\epsilon}{(1+\epsilon)^{(1+\epsilon)}}\right)^{\frac{m\Delta_\sigma(u,v)}{s\Delta(u,v)}}$$

and

$$Pr\left[\left(\sum_{j=1}^{m}|f_\sigma(u)_{I_j}-f_\sigma(v)_{I_j}|\right)\frac{1}{s\Delta(u,v)}\leq(1-\epsilon)\frac{m}{s\Delta(u,v)}E_{I\in\mathcal{I}}[|f_\sigma(u)_I-f_\sigma(v)_I|]\right]$$

$$<\left(\frac{e^{-\epsilon}}{(1-\epsilon)^{(1-\epsilon)}}\right)^{\frac{m\Delta_\sigma(u,v)}{s\Delta(u,v)}},$$

where the I_j are i.i.d. random variables chosen from $\mathcal{I}$. Noticing that $\frac{\Delta_\sigma(u,v)}{\Delta(u,v)}\geq\frac{1}{D}$, we set $m=8sD\log(n)/\epsilon^2$. For this choice of m the probability that, for a particular pairwise distance, the average over m samples is not within $(1\pm\epsilon)$ of its expectation is at most $1/n^2$ (for $\epsilon>0$ small enough). Since there are $\binom{n}{2}$ pairs of points, $P[\text{relative error is at most } \epsilon]\geq 1/2$. Hence there exists an embedding in ℓ_1^m with $m=8sD\log(n)/\epsilon^2$ and distortion at most $D(1+\epsilon)$.[1] □

Note that stretch-s embeddings are more general than ℓ_1 embeddings with s dimensions: Not only will we allow them to have arbitrary dimension, but we allow them to use any convex combination of line metrics, not simply an average. By the results above, stretch is a good proxy for dimension in ℓ_1 embeddings.

2.2 Applications and questions

Limited-stretch embeddings were developed by Charikar and Sahai [15] to prove both positive and negative dimension reduction results.

Theorem 2.2.1 (No ℓ_1 dimension reduction by linear embeddings [15]). *We call an embedding $f=\{f(\cdot)_1,f(\cdot)_2,\ldots,f(\cdot)_d\}$ of ℓ_1^n into ℓ_1^d linear if each $f(\cdot)_i$ is a*

[1] Note that if we used the stretch-2 definition discussed earlier, the claim can be strengthened to guarantee an embedding into $\ell_1^{O(s\log n/\epsilon^2)}$.

linear function on the components of the input point:

$$f(x)_i = \sum_{j=1}^{n} \lambda_{ij} x_j.$$

For every sufficiently large n there exists a set of $O(n)$ points in ℓ_1^n which for any linear embedding into ℓ_1^d incurs distortion $\Omega(\sqrt{n/d})$.

Bourgain's Theorem 1.5.2, along with the Johnson-Lindenstrauss Lemma 1.6.1, implies an $O(\log(n))$ dimension $O(\log(n))$ distortion embedding of any n point metric into ℓ_1. Bourgain's embedding seems to be inherently non-linear, and the application of Theorem 2.2.1 to Bourgain's embedding confirms this fact. This also brings to light the fact that linear embeddings are a highly restricted class of embeddings, and that this theorem does not disprove the existence of exponential dimension reduction for ℓ_1.

It is also possible to use the concept of stretch-limited embeddings to construct low dimensional embeddings into ℓ_1, incurring only about a factor $D\log(n)$ extra dimensions. In fact, this is the idea that is implicit behind Charikar and Sahai's cycle flattenings that are used in embedding $K_{2,3}$-free graphs into low dimensional ℓ_1. They used an embedding of $K_{2,3}$-free graphs into a convex combination of two trees, but observed that traditional Bartal-like [9, 23] probabilistic embeddings into trees cannot be used. Bartal's construction uses edge deletions, an intrinsically high-stretch operation. The explicit link between stretch and dimension should greatly simplify the design of low-dimensional ℓ_1 embeddings, though we do not yet know of any other results in this direction.

Chapter 3

Some metrics require high dimension in ℓ_1

In this chapter we will prove that there are some ℓ_1 metric spaces which require polynomially many dimensions if only constant distortion is allowed.

3.1 Series-parallel graphs require high dimension in ℓ_1

We will first prove Theorem 1.9.2, that there are families of series-parallel graphs that require $n^{\Omega(1/D^2)}$ dimensions to embed in ℓ_1 with distortion at most D.

3.1.1 Proof overview

In Section 1.2.3 we showed a general approach for proving distortion lower bounds. In order to prove lower bounds for dimension reduction in ℓ_1, we adapt this technique.

As we observed in the previous chapter, low dimensional embeddings seem tricky to reason about. Instead we focus on low-stretch embeddings, exploiting the connection between stretch-limited embeddings and embeddings in low dimensions. Our goal will be to prove a lower bound on the stretch s needed to achieve a given distortion D.

Reducing stretch-limited embeddings to stretch-limited line embeddings

Recall from Section 2 that a stretch-limited embedding can be assumed to be non-expansive without loss of generality. As a result, a stretch-limited embedding with distortion D must satisfy the property that no distance expands and also no distance contracts by more than a factor D. Let us write down these constraints as we would give them for an optimization problem:

$$\forall u, v \in X,\ \Delta_\sigma(f_\sigma(u), f_\sigma(v)) \leq ||u - v||_1 \text{ (Non-expansion constraint } NE_{uv})$$

and

$$\forall u, v \in X,\ \frac{1}{D}||u - v||_1 \leq \Delta_\sigma(f_\sigma(u), f_\sigma(v)) \text{ (Low distortion constraint } LD_{uv}).$$

Now consider linear combination of these constraints of the form

$$\forall \gamma_{uv}, \lambda_{uv} \geq 0,\ \sum_{u,v \in X} \left(\gamma_{uv} NE_{uv} + \lambda_{uv} LD_{uv}\right).$$

This is a class of linear constraints, and if all of the NE_{uv} and LD_{uv} are satisfied, so is any linear inequality L of this form. Recall that a stretch-limited embedding is a convex combination of line metrics. From the convexity of stretch-limited embeddings we can see that:

Observation 3.1.1. *If L is satisfied by some stretch-limited embedding σ, then L must be satisfied by at least one of the line embeddings $f_\sigma(\cdot)_i$.*

Our task now is to, given some family of series-parallel graph metrics, derive a single inequality L by finding values for the the γ_{uv} and λ_{uv}. The intuition is that this L should be hard to satisfy, and we pick the γ_{uv} and λ_{uv} to get the best bound possible. We will then show that any dimension which satisfies L incurs high stretch, and therefore that no stretch-limited embedding with low stretch exists.

Charikar and Sahai [15] used this technique to prove Theorem 2.2.1, as stated in the previous chapter. In that case, the restriction to linear embeddings and a careful choice of the inequality on pairwise distances made it possible to prove a lower bound on the stretch s required. How can one prove lower bounds on the stretch for arbitrary (i.e. non-linear) line embeddings? Our innovation is to express the problem of minimizing stretch so as to satisfy the inequality L as a linear program. In general, finding such an LP formulation might be very difficult. However, we are able to obtain an LP that minimizes stretch for a carefully chosen family of points in ℓ_1 and a particular set of linear inequalities on pairwise distances. Having obtained the LP formulation, we consider the dual LP and exhibit a dual feasible solution. This establishes a lower bound on the stretch.

Our dual solution was in fact extrapolated from embeddings generated by the CPLEX LP solver for large instances of our LP. This allowed us to discover combinatorial structure in our problem that we could leverage for our proof. We will ignore the details of this process in giving the proof, saving a more detailed exposition for

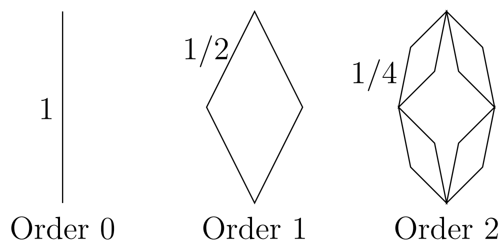

Figure 3: Recursive diamond graphs of different orders

Section 4.1. We will also temporarily ignore the problem of how to derive the "correct" values of γ_{uv} and λ_{uv}. This is, however, crucial to obtaining a good lower bound on stretch, and we will discuss our solution to this problem in Section 4.2. Again, optimization techniques and LP duality come to the rescue. For now we will present our proof by directly obtaining the single "hard" constraint L.

3.1.2 The recursive diamond graph

In order to prove our results, we will focus on one particular family of series-parallel graphs which we call the recursive diamond graphs. These are the graphs that Newman and Rabinovich [59] previously used to establish an $\Omega(\sqrt{\log n})$ lower bound for embedding planar graphs into ℓ_2 (Theorem 1.5.8). The order 0 recursive diamond graph is a single edge, with length one. In order to make the order k graph from the order $k-1$ graph, replace each edge of length $1/2^{k-1}$ with a four-edge diamond, with edges of length $1/2^k$ (See Figure 3). This is a family of series-parallel graphs with 4^k edges and $\frac{2}{3}4^k + \frac{4}{3}$ vertices. Furthermore, Theorem 1.6.6 of Gupta et al. [32] shows that this graph can be embedded into ℓ_1 with constant distortion[1] (with many

[1] The recursive diamond graph is in fact a series-parallel "bundle." Gupta et al., in the full version of [32] (to appear in Combinatorica), show that such graphs can be embedded into ℓ_1 with

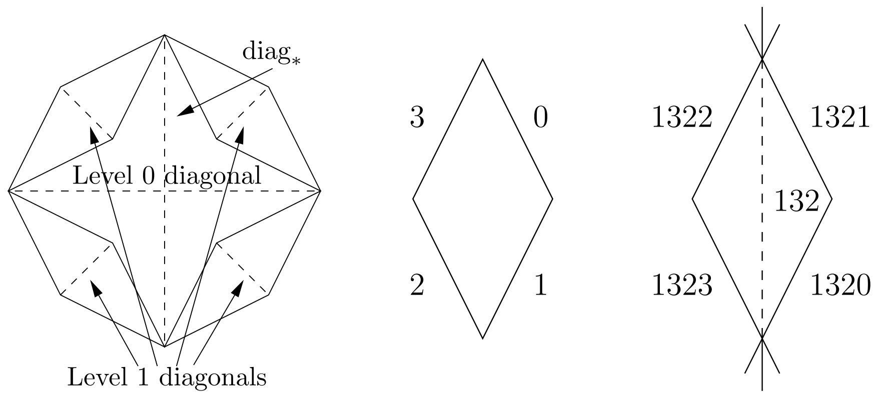

Figure 4: Diamond graph terminology and labels for edges

dimensions).

We will need some terminology in order to talk about the graph (see Figure 4). We will use n to refer to the number of vertices in a given diamond graph, and k to refer to the order (number of levels) of the graph. Each vertex has some k such that it is present in the order k graph, but not in the order $k-1$ graph. We will refer to any vertex as a *level* k vertex if it first appears in the order k graph. When an edge is replaced with a diamond, the two new vertices that are created will be called *siblings*, and we will refer to the pair of siblings as the *diagonal*[2] of this diamond. We will say that it is a level k diagonal if the vertices concerned are level k vertices. Finally, there is a natural parent-child relationship between diamonds of different levels: A diamond is a child of the diamond whose edge it replaces. An ancestor of a diamond is defined in the obvious way, and the ancestors of an edge are the diamonds of each order in which the edge participates.

---

distortion 2. We present an elementary distortion 2 embedding of the recursive diamond graph in Section 3.2.1.

[2]These are called *anti-edges* in [59].

Every edge in the graph is labeled by a string in $\{0,1,2,3\}^k$. A particular diamond has four edges, which we will number 0, 1, 2 and 3 (again, see Figure 4). The label for the ith edge of a diamond is obtained by concatenating i to the label of its parent edge. We label diamonds with the label of the parent edge. Also, we use this same label to label the diagonal edge. For a label x, edge(x) denotes the edge labeled by x and diag(x) denotes the diagonal whose label is x. This leaves the original edge of the graph unlabeled. We will treat it as being "diagonal like" and refer to it as diag$_*$. We will return to the matter of exactly specifying the labeling in a later section: For now it is sufficient to notice that the 0 edge is always opposite the 2 edge and the 1 edge is always opposite the 3 edge.

3.1.3 The recursive diamond graph requires high ℓ_1 dimension

We may now state our first result.

Theorem 3.1.2. *A recursive diamond graph on n vertices requires $n^{\Omega(1/D^2)}$ dimensions to embed in ℓ_1 with distortion at most D.*

We focus on the n-vertex diamond graph (which has k levels). Consider $\mathcal{E}$, the set of all edges in the graph, and $\mathcal{D}$, the set of all diagonals. We will bound D by showing that the edges tend to expand while the diagonals tend to contract (refer back to Sections 1.2.3 and 3.1.1).

The D-distortion constraint

We first develop our key constraint on edge and diagonal lengths imposed by D. Recall that we have labeled the edges and diagonals. We refer to the length of edge(x) as

e'_x and the length of diag(y) as d_y.[3]

We assumed w.l.o.g. that our embedding σ is non-expansive and has distortion at most D. The non-expansive property of σ implies that

$$-\sum_{x\in\{0,1,2,3\}^k} e'_x/2^k \geq -1.$$

There are 4^k edges of length $\frac{1}{2^k}$, so this says that the average length of an edge must be at most $\frac{1}{2^k}$. The D-distortion property implies that

$$D\left(d_* + \sum_{i=0}^{k-1}\sum_{y\in\{0,1,2,3\}^i} d_y/2^i\right) \geq k+1.$$

At a level i there are 4^i diagonals of length 2^i. This bound gives a weighted average of the diagonal lengths such that the total contribution of each level to the average is the same, and says that this weighted average cannot be more than a factor D smaller in the embedding than for the original metric. We combine these constraints to get a single constraint (referred to as the distortion constraint) that should be hard to satisfy:

$$\forall\gamma \geq 0,\ D\left(d_* + \sum_{i=0}^{k-1}\sum_{y\in\{0,1,2,3\}^i} d_y/2^i\right) - \gamma \sum_{x\in\{0,1,2,3\}^k} e'_x/2^k \geq k+1-\gamma.$$

We will eventually optimize γ in order to make this bound as strong as possible.

If this is true for a convex combination of line metrics, then it must be true for at least one of those line metrics. We will show a lower bound on the stretch s which

[3]We use e'_x here because we later define a variable e_x which is the *signed* length of an edge. The reader may safely ignore this distinction for now.

must be incurred by a line metric which satisfies this constraint given values of n, k and D.

Let ρ be a line metric from σ which might satisfy this constraint. In order to simplify notation, let us re-interpret the meaning of e'_x to be the length of edge(x) in ρ and d_x to be the length of diag(x) in ρ. From this point on we will work only in the candidate line ρ, and all distances are in ρ unless otherwise noted.

Constraints on edges and diagonals

Before we can write down our LP we will need a few more constraints. There is a very strong relationship between the length of an edge and the lengths of the diagonals of the edge's ancestor diamonds, and this will give us a second set of constraints.

We first precisely specify the labeling scheme for the edges of the recursive diamond graph. The labeling scheme we choose will depend on the particular ρ we are considering, but we will describe how to choose a labeling which satisfies our needs for *any* given ρ.

For each edge(x), we will designate one end point as the *head* and the other as the *tail* (denoted by head(edge(x)) and tail(edge(x)) respectively). Similarly, the end points of diag(x) are labeled as the *top* end point (denoted by top(diag(x))) and the *bottom* end point (denoted by bot(diag(x))). This labeling is done such that $\rho(\text{top}(\text{diag}(x))) \geq \rho(\text{bot}(\text{diag}(x)))$ (ties are broken arbitrarily).[4] We will now derive, for every edge(x) in the graph, an expression for $\rho(\text{head}(\text{edge}(x))) - \rho(\text{tail}(\text{edge}(x)))$ in terms of the lengths of edge(x)'s parent diagonals.

The edges of the diamond connect the end points of the parent edge(x) to the end points of diag(x) (see Figure 5). The edge connecting head(edge(x)) to top(diag(x))

[4]This choice is what causes the labeling to be dependent on ρ.

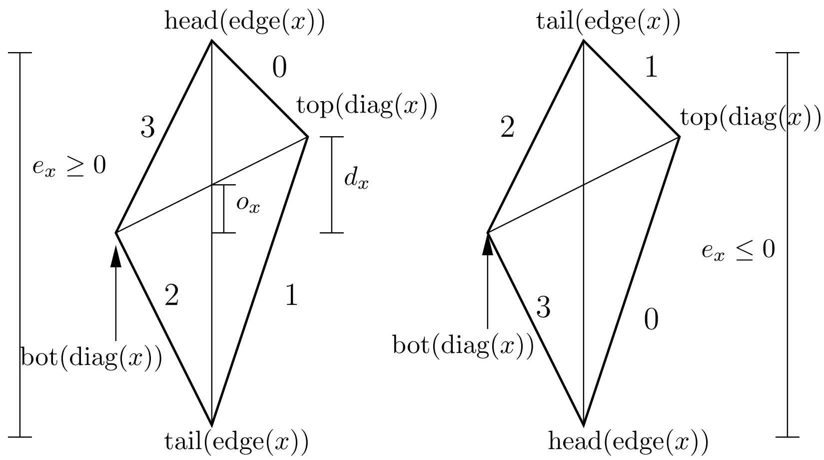

Figure 5: Single dimension of embedded diamond

is called the 0-edge. The edge connecting top(diag(x)) to tail(edge(x)) is called the 1-edge. The edge connecting bot(diag(x)) to tail(edge(x)) is called the 2-edge. The edge connecting head(edge(x)) to bot(diag(x)) is called the 3-edge. Further, head(edge(x)) is considered the *head* for the 0-edge and the 3-edge; for each of these edges, the end point of diag(x) incident on it is considered the *tail*. tail(edge(x)) is considered the *tail* for the 1-edge and the 2-edge; for each of these edges, the end point of diag(x) incident on it is considered the *head*.

We define the following:

$$\begin{aligned}
e_x &= \rho(\text{head}(\text{edge}(x))) - \rho(\text{tail}(\text{edge}(x))) \\
d_x &= \rho(\text{top}(\text{diag}(x))) - \rho(\text{bot}(\text{diag}(x))) \\
o_x &= \frac{\rho(\text{top}(\text{diag}(x))) + \rho(\text{bot}(\text{diag}(x)))}{2} - \frac{\rho(\text{head}(\text{edge}(x))) + \rho(\text{tail}(\text{edge}(x)))}{2}
\end{aligned}$$

Note that these definitions correspond exactly to our earlier definitions: $|e_x| = e'_x$ is the length of edge(x) (we allow e_x to be negative when head(edge(x)) < tail(edge(x))) in ρ and d_x is the length of diag(x) in ρ. We refer to o_x as the *offset* of the diamond labeled x in ρ.

Now, we can calculate e_{x0}, e_{x1}, e_{x2} and e_{x3} in terms of e_x, d_x and o_x as follows:

Lemma 3.1.3.

$$\begin{array}{ll} e_{x0} = \frac{e_x}{2} - \frac{d_x}{2} - o_x & e_{x1} = \frac{e_x}{2} + \frac{d_x}{2} + o_x \\ e_{x2} = \frac{e_x}{2} - \frac{d_x}{2} + o_x & e_{x3} = \frac{e_x}{2} + \frac{d_x}{2} - o_x \end{array}$$

Proof. We will show the calculation for e_{x0} only.

$$\begin{aligned} e_{x0} &= \rho(\text{head}(\text{edge}(x0))) - \rho(\text{tail}(\text{edge}(x0))) \\ &= \rho(\text{head}(\text{edge}(x))) - \rho(\text{top}(\text{diag}(x))) \\ &= \frac{\rho(\text{head}(\text{edge}(x))) - \rho(\text{tail}(\text{edge}(x)))}{2} + \frac{\rho(\text{head}(\text{edge}(x))) + \rho(\text{tail}(\text{edge}(x)))}{2} \\ &\quad - \frac{\rho(\text{top}(\text{diag}(x))) - \rho(\text{bot}(\text{diag}(x)))}{2} - \frac{\rho(\text{top}(\text{diag}(x))) + \rho(\text{bot}(\text{diag}(x)))}{2} \\ &= \frac{e_x}{2} - \frac{d_x}{2} \\ &\quad - \frac{\rho(\text{top}(\text{diag}(x))) + \rho(\text{bot}(\text{diag}(x)))}{2} + \frac{\rho(\text{head}(\text{edge}(x))) + \rho(\text{tail}(\text{edge}(x)))}{2} \\ &= \frac{e_x}{2} - \frac{d_x}{2} - o_x \end{aligned}$$

The proofs for e_{x1}, e_{x2} and e_{x3} are similar. □

Using this, one can obtain an expression for e_x in terms of the diagonal lengths and the offsets of the diamonds which are ancestors of edge(x). We use $y \sqsubset x$ to denote that y is a prefix of x, and the empty string is a prefix of every string. We also use $|x|$ to denote the length of the string x.

Lemma 3.1.4.

$$e_x = \frac{d_*}{2^{|x|}} + \sum_{y \sqsubset x} S(x_{|y|+1}) \frac{d_y}{2^{|x|-|y|}} + \sum_{y \sqsubset x} T(x_{|y|+1}) \frac{o_y}{2^{|x|-|y|-1}}$$

where $S(0) = S(2) = -1, S(1) = S(3) = +1$ *and* $T(0) = T(3) = -1, T(1) = T(2) = +1$.

Proof. We prove this by induction on $|x|$.

Base Case: Consider $|x| = 0$. In this case, $e_x = d_*$ and the statement is true.

Inductive Step: Suppose the statement is true for all x such that $|x| = i$. Now consider e_{x0}, where $|x| = i$. From Lemma 3.1.3, $e_{x0} = \frac{e_x}{2} - \frac{d_x}{2} - o_x$. Using the expression we have for e_x from the inductive hypothesis, we get:

$$\begin{aligned}
e_{x0} &= \frac{1}{2}\left(\frac{d_*}{2^i} + \sum_{y \sqsubset x} \frac{S(x_{|y|+1})d_y}{2^{i-|y|}} + \sum_{y \sqsubset x} \frac{T(x_{|y|+1})o_y}{2^{i-|y|-1}}\right) - \frac{d_x}{2} - o_x \\
&= \left(\frac{d_*}{2^{i+1}} + \sum_{y \sqsubset x} \frac{S(x_{|y|+1})d_y}{2^{i+1-|y|}} + \sum_{y \sqsubset x} \frac{T(x_{|y|+1})o_y}{2^{i+1-|y|-1}}\right) - \frac{d_x}{2} - o_x \\
&= \frac{d_*}{2^{i+1}} + \sum_{y \sqsubset x0} \frac{S(x_{|y|+1})d_y}{2^{i+1-|y|}} + \sum_{y \sqsubset x0} \frac{T(x_{|y|+1})o_y}{2^{i+1-|y|-1}}
\end{aligned}$$

Thus the statement holds for e_{x0} as well. Similarly, we can show that the statement holds for e_{x1}, e_{x2} and e_{x3}. By induction, the statement of the lemma is true. □

Grouping edges and diagonals

Before we continue, we would like to remove the dependence on o_y. We noticed in experiments that optimal embeddings had all the o_y set to zero, so we expect removing the o_y should not hurt our bounds much. In fact, we will place our edges and diagonals into groups, and write our constraints in terms of the average distances in these groups. The careful choice of our labeling will cause the o_y terms to cancel out.

In particular, we group edges into 2^k groups of 2^k edges each. Groups are identified with labels in $\{0,1\}^k$. For a group labeled by $z \in \{0,1\}^k$, edge(x) belongs to the group z if $x(mod\ 2) = z$. Here $x(mod\ 2)$ refers to the label obtained by performing a coordinate-wise $mod\ 2$ operation. Similarly, diagonals of level i are grouped into 2^i groups, identified with labels in $\{0,1\}^i$.

$$\begin{aligned} \overline{e_z} &= \frac{1}{2^k} \sum_{\{x : x(mod\ 2) = z\}} |e_x| \\ \overline{d_z} &= \frac{1}{2^i} \sum_{\{x : x(mod\ 2) = z\}} d_x \end{aligned}$$

In other words, $\overline{e_z}$ and $\overline{d_z}$ are the average lengths of their constituent edges and diagonals.

We can immediately rewrite our distortion constraint in terms of $\overline{e_z}$ and $\overline{d_z}$ without changing anything.

$$D \left(\overline{d_*} + \sum_{i=0}^{k-1} \sum_{y \in \{0,1\}^i} \overline{d_y} \right) - \gamma \sum_{x \in \{0,1\}^k} \overline{e_x} \geq k + 1 - \gamma$$

Claim 3.1.5. *For a group label* $z \in \{0,1\}^k$,

$$\frac{1}{2^k} \sum_{\{x:x(mod\ \ 2)=z\}} e_x \quad = \quad \frac{d_*}{2^k} + \sum_{y \sqsubset z} S(z_{|y|+1}) \frac{\overline{d_y}}{2^{k-|y|}}$$

Proof. Using Lemma 3.1.4, the value of the LHS is as follows:

$$\begin{aligned}
\text{LHS} \quad &= \quad \frac{1}{2^k} \sum_{\{x:x(mod\ \ 2)=z\}} \left(\frac{d_*}{2^k} + \sum_{y \sqsubset x} \frac{S(x_{|y|+1}) d_y}{2^{k-|y|}} + \sum_{y \sqsubset x} \frac{T(x_{|y|+1}) o_y}{2^{k-|y|-1}} \right) \\
&= \quad \frac{d_*}{2^k} + E_1(z) + E_2(z)
\end{aligned}$$

where

$$\begin{aligned}
E_1(z) \quad &= \quad \frac{1}{2^k} \sum_{\{x:x(mod\ \ 2)=z\}} \sum_{y \sqsubset x} \frac{S(x_{|y|+1}) d_y}{2^{k-|y|}} \\
E_2(z) \quad &= \quad \frac{1}{2^k} \sum_{\{x:x(mod\ \ 2)=z\}} \sum_{y \sqsubset x} \frac{T(x_{|y|+1}) o_y}{2^{k-|y|-1}}
\end{aligned}$$

We now simplify the two expressions $E_1(z)$ and $E_2(z)$. Note that $x, y \in \{0,1,2,3\}^j$ for some j while $y', z \in \{0,1\}^j$.

$$\begin{aligned}
E_1(z) \quad &= \quad \frac{1}{2^k} \sum_{\{x:x(mod\ \ 2)=z\}} \sum_{y \sqsubset x} \frac{S(x_{|y|+1}) d_y}{2^{k-|y|}} \\
&= \quad \sum_{y' \sqsubset z} \frac{1}{2^{|y'|}} \sum_{\{y:y(mod\ \ 2)=y'\}} \frac{S(z_{|y|+1}) d_y}{2^{k-|y|}} \\
&= \quad \sum_{y' \sqsubset z} \frac{S(z_{|y'|+1}) \overline{d_{y'}}}{2^{k-|y'|}}
\end{aligned}$$

$$\begin{aligned}
E_2(z) &= \frac{1}{2^k} \sum_{\{x:x(mod\ \ 2)=z\}} \sum_{y \sqsubset x} \frac{T(x_{|y|+1})o_y}{2^{k-|y|-1}} \\
&= \sum_{y' \sqsubset z} \frac{1}{2^{|y'|+1}} \sum_{\{y:y(mod\ \ 2)=y'\}} \frac{(T(z_{|y|+1}) + T(2+z_{|y|+1}))o_y}{2^{k-|y|-1}} \\
&= 0
\end{aligned}$$

The last equality follows from the fact that $T(i) + T(2+i) = 0$ for $i \in \{0,1\}$. Substituting the values of $E_1(z)$ and $E_2(z)$ in the expression we derived earlier proves the claim. □

Lemma 3.1.6. *For a group label $z \in \{0,1\}^k$,*

$$\begin{aligned}
\overline{e_z} &\geq \frac{d_*}{2^k} + \sum_{y \sqsubset z} S(z_{|y|+1}) \frac{\overline{d_y}}{2^{k-|y|}} \\
\overline{e_z} &\geq -\frac{d_*}{2^k} - \sum_{y \sqsubset x} S(z_{|y|+1}) \frac{\overline{d_y}}{2^{k-|y|}}
\end{aligned}$$

Proof.

$$\begin{aligned}
\overline{e_z} &= \frac{1}{2^k} \sum_{\{x:x(mod\ \ 2)=z\}} |e_x| \\
&\geq \frac{1}{2^k} \sum_{\{x:x(mod\ \ 2)=z\}} e_x
\end{aligned}$$

Using Claim 3.1.5, we get the first inequality we need to prove. Also,

$$\begin{aligned}
\overline{e_z} &= \frac{1}{2^k} \sum_{\{x:x(mod\ \ 2)=z\}} |e_x| \\
&\geq -\frac{1}{2^k} \sum_{\{x:x(mod\ \ 2)=z\}} e_x
\end{aligned}$$

Table 6: The main linear program

min s			
	$D\left(\overline{d_*}+\sum_{i=0}^{k-1}\sum_{y\in\{0,1\}^i}\overline{d_y}\right)-\gamma\sum_{z\in\{0,1\}^k}\overline{e_z}$	$\geq k+1-\gamma$	$[\mu]$
$\forall z\in\{0,1\}^k$	$\frac{s}{2^k}-\overline{e_z}$	≥ 0	$[p_z]$
$\forall z\in\{0,1\}^k$	$\overline{e_z}+\left(\frac{\overline{d_*}}{2^k}+\sum_{y\sqsubset z}\frac{S(z_{\lvert y\rvert+1})\overline{d_y}}{2^{k-\lvert y\rvert}}\right)$	≥ 0	$[\alpha_z]$
$\forall z\in\{0,1\}^k$	$\overline{e_z}-\left(\frac{\overline{d_*}}{2^k}+\sum_{y\sqsubset z}\frac{S(z_{\lvert y\rvert+1})\overline{d_y}}{2^{k-\lvert y\rvert}}\right)$	≥ 0	$[\beta_z]$

Again, using Claim 3.1.5 gives the second inequality. □

These inequalities result when we replace the e_x in Claim 3.1.5 with the $|e_x|$ from the definition of $\overline{e_z}$.

Linear program for minimizing stretch

We have already derived three of the four constraints that we will use in our linear program. All that remains is to provide a lower bound for stretch.

Consider the stretch incurred by edge(x) in the line ρ. For every edge$(x) = (u, v)$, $s \geq |e_x|/\Delta(u, v) \geq 2^k|e_x|$, where Δ is understood to be the distance function for the original metric space. Since $\max_{\{x:x(mod\ 2)=z\}}|e_x| \geq \overline{e_z}$, we conclude that $\forall z \in \{0,1\}^k$ $s \geq 2^k\overline{e_z}$.

Now we are ready to give our linear program (see Table 6). Note that we will optimize γ later, but that it is constant with respect to the variables of the LP. We provide the names of the dual variables in brackets for reference. We have carefully derived our constraints so that we can see that the solution to our LP is no larger than the minimum stretch needed to embed the recursive diamond graph into ℓ_1.

Table 7: The main dual linear program

max $(k+1-\gamma)\mu$			
$\forall z \in \{0,1\}^k$	$-\gamma\mu - p_z + \alpha_z + \beta_z$	≤ 0	$[\overline{e_z}]$
	$\sum_{z\in\{0,1\}^k} p_z$	$\leq 2^k$	$[s]$
$\forall y \in \bigcup_{i\in[0,k-1]} \{0,1\}^i$	$D\mu + \sum_{v\in\{0,1\}^{k-\lvert y\rvert}} \frac{S((yv)_{\lvert y\rvert+1})(\alpha_{yv}-\beta_{yv})}{2^{k-\lvert y\rvert}}$	≤ 0	$[\overline{d_y}]$
	$D\mu + \sum_{z\in\{0,1\}^k} \frac{\alpha_z-\beta_z}{2^k}$	≤ 0	$[\overline{d_*}]$

Table 8: The dual with μ factored out

max $(k+1-\gamma)\mu$			
$\forall z \in \{0,1\}^k$	$\mu\left(-\gamma - p^*_z + \alpha^*_z + \beta^*_z\right)$	≤ 0	$[\overline{e_z}]$
	$\mu\left(\sum_{z\in\{0,1\}^k} p^*_z\right)$	$\leq 2^k$	$[s]$
$\forall y \in \bigcup_{i\in[0,k-1]} \{0,1\}^i$	$\mu\left(D + \sum_{v\in\{0,1\}^{k-\lvert y\rvert}} \frac{S((yv)_{\lvert y\rvert+1})(\alpha^*_{yv}-\beta^*_{yv})}{2^{k-\lvert y\rvert}}\right)$	≤ 0	$[\overline{d_y}]$
	$\mu\left(D + \sum_{z\in\{0,1\}^k} \frac{\alpha^*_z-\beta^*_z}{2^k}\right)$	≤ 0	$[\overline{d_*}]$

Dual linear program for the lower bound on stretch

We have formulated an LP minimization problem whose optimum value is a lower bound on the minimum stretch for a D-distortion embedding. In order to prove our lower bound we give the dual of this LP and a feasible solution. We construct the dual in the normal way (see Table 7).

Next we give our solution for this LP. In fact, our solution is very simple. Every variable is just a constant multiple of μ: $p_x = p^*_x\mu$, $\alpha_x = \alpha^*_x\mu$ and $\beta_x = \beta^*_x\mu$. We will specify the values of these constants, and then maximize μ subject to the constraints of the dual in order to get our bound. For these purposes, we can rewrite the dual LP (see Table 8).

Table 9: The dual solution

$$\gamma = k - D$$

$$\alpha^*_x = \begin{cases} D(k-1-2\|x\|_1) & \text{if } \|x\|_1 \leq k/2-1 \\ 0 & \text{otherwise} \end{cases}$$

$$\beta^*_x = \begin{cases} D(2\|x\|_1-k+1) & \text{if } \|x\|_1 \geq k/2 \\ 0 & \text{otherwise} \end{cases}$$

$$p^*_x = \begin{cases} 2D(k/2-k/2D-\|x\|_1) & \text{if } \|x\|_1 \leq k/2-k/2D-1 \\ 2D(\|x\|_1-k/2-k/2D+1) & \text{if } \|x\|_1 \geq k/2+k/2D \\ 0 & \text{otherwise} \end{cases}$$

The dual solution

We now give our solution to the dual in Table 9. We use $\|x\|_1$ to denote the number of 1s in the 0-1 string x.

Claim 3.1.7. *The values of γ, α^*_z, β^*_z and p^*_z in Table 9 give a feasible solution for our dual LP.*

Proof. First check the $\overline{e_z}$ constraint, that $(-\gamma - p^*_z + \alpha^*_z + \beta^*_z) \leq 0$ for all z. We break this into three cases.

Case 1: $\|z\|_1 \leq k/2 - k/2D - 1$

$$\begin{aligned} -p^*_z + \alpha^*_z + \beta^*_z &= D(k-1-2\|z\|_1) - 2D(k/2-k/2D-\|z\|_1) \\ &= \gamma \end{aligned}$$

Case 2: $\|z\|_1 \geq k/2 + k/2D$

$$\begin{aligned} -p^*_z + \alpha^*_z + \beta^*_z &= D(2\|z\|_1-k+1) - 2D(\|z\|_1-k/2-k/2D+1) \\ &= \gamma \end{aligned}$$

Case 3: $k/2 - k/2D - 1 < \|z\|_1 < k/2 + k/2D$

$$\begin{aligned} p^*_z &= 0 \\ \alpha^*_z &\leq D(k-1-2(k/2-k/2D)) = k - D = \gamma \\ \beta^*_z &\leq D(2(k/2+k/2D-1)-k+1) = k - D = \gamma \end{aligned}$$

Since the ranges where α^*_x and β^*_x are positive do not overlap, this proves that the $\overline{e_z}$ constraint is satisfied.

Now let us skip to the $\overline{d_y}$ constraint. In order to prove this, we will use the following lemma:

Lemma 3.1.8.

$$\forall i \in [0, k-1], y \in \{0,1\}^i, v \in \{0,1\}^{k-i-1}$$
$$S((y0v)_{|y|+1})(\alpha^*_{y0v} - \beta^*_{y0v}) + S((y1v)_{|y|+1})(\alpha^*_{y1v} - \beta^*_{y1v}) = -2D$$

Proof. $S((y0v)_{|y|+1}) = -1$ and $S((y1v)_{|y|+1}) = +1$, so

$$\begin{aligned} S((y0v)_{|y|+1})(\alpha^*_{y0v} - \beta^*_{y0v}) + & \\ S((y1v)_{|y|+1})(\alpha^*_{y1v} - \beta^*_{y1v}) &= -\alpha^*_{y0v} + \beta^*_{y0v} + \alpha^*_{y1v} - \beta^*_{y1v}. \end{aligned}$$

Since $\|y0v\|_1 + 1 = \|y1v\|_1$, there are three cases. **Case 1:** $\|y0v\|_1, \|y1v\|_1 \leq k/2 - 1$

$$\begin{aligned} -\alpha^*_{y0v} + \beta^*_{y0v} + \alpha^*_{y1v} - \beta^*_{y1v} &= \\ -\alpha^*_i + \alpha^*_{i+1} &= \\ -D(k-1-2i) + D(k-1-2(i+1)) &= -2D \end{aligned}$$

Case 2: $\|y0v\|_1, \|y1v\|_1 \geq k/2$

$$\begin{aligned} -\alpha^*_{y0v} + \beta^*_{y0v} + \alpha^*_{y1v} - \beta^*_{y1v} &= \\ \beta^*_i - \beta^*_{i+1} &= \\ D(2i-k+1) - D(2(i+1)-k+1) &= -2D \end{aligned}$$

Case 3: $\|y0v\|_1 = k/2-1, \|y1v\|_1 = k/2$

$$\begin{aligned} -\alpha^*_{y0v} + \beta^*_{y0v} + \alpha^*_{y1v} - \beta^*_{y1v} &= \\ -\alpha^*_i - \beta^*_{i+1} &= \\ -D - D &= -2D \end{aligned}$$

□

Applying lemma 3.1.8, we conclude that

$$\begin{aligned} D + \sum_{v \in \{0,1\}^{k-|y|}} \frac{S((yv)_{|y|+1})(\alpha^*_{yv} - \beta^*_{yv})}{2^{k-|y|}} &= \\ D - \sum_{v' \in \{0,1\}^{k-|y|-1}} \frac{2D}{2^{k-|y|}} &= \\ D - \frac{(2^{k-|y|-1})(2D)}{2^{k-|y|}} &= 0 \end{aligned}$$

Hence the $\overline{d_y}$ constraints are all satisfied: In fact, they are all tight.

The case for the $\overline{d_*}$ constraint is even simpler because the sign for α^*_z is always positive and the sign for β^*_z is always negative. For every x with

$$\|x\|_1 = l \leq k/2 - 1,$$

pair x with y such that $(x \text{ xor } y) = 111\ldots1$ (in other words, y is the bitwise NOT of x). Note that

$$\|y\|_1 = k - l \geq k/2 + 1,$$

and that

$$\alpha^*_l - \beta^*_{k-l} = D(k-1-2l) - D(2(k-l) - k + 1) = -2D.$$

This accounts for all x except where $\|x\|_1 = k/2$. In this case α_* is 0, so we see that the $\overline{d_*}$ constraint is satisfied.

Finally, we return to the s constraint. Recall that our lower bound will be $(1+D)\mu$. This constraint is the only one which limits μ, and we will try to make μ as big as we can. Hence, $\mu = 2^k/\sum_z p^*_z$. Let us now bound p^*_z:

$$\begin{aligned}
\sum p^*_z &= 2D\left(\sum_{i=0}^{\frac{k}{2}-\frac{k}{2D}-1}\left(\frac{k}{2}-\frac{k}{2D}-i\right)\binom{k}{i} + \sum_{i=\frac{k}{2}+\frac{k}{2D}}^{k}\left(i-\frac{k}{2}-\frac{k}{2D}+1\right)\binom{k}{i}\right) \\
&= 2D\left(\sum_{i=\frac{k}{2}+\frac{k}{2D}+1}^{k}\left(i-\frac{k}{2}-\frac{k}{2D}\right)\binom{k}{i} + \sum_{i=\frac{k}{2}+\frac{k}{2D}}^{k}\left(i-\frac{k}{2}-\frac{k}{2D}+1\right)\binom{k}{i}\right) \\
&= 2D\left(\sum_{i=\frac{k}{2}+\frac{k}{2D}}^{k}\left(2\left(i-\frac{k}{2}-\frac{k}{2D}\right)+1\right)\binom{k}{i}\right)
\end{aligned}$$

Now observe that

$$\begin{aligned}\binom{k}{i+1} &= \frac{k-i}{i+1}\binom{k}{i} \leq \frac{k/2-k/2D}{k/2+k/2D}\binom{k}{i} \\ &= \frac{D-1}{D+1}\binom{k}{i},\end{aligned}$$

which implies that

$$\binom{k}{i+t} \leq \left(\frac{D-1}{D+1}\right)^t \binom{k}{i}.$$

It is simple to check that

$$\begin{aligned}\sum_{t=0}^{\infty}(2t+1)r^t &= 2\sum_{t=0}^{\infty}(t+1)r^t - \sum_{t=0}^{\infty} r^t \\ &= \frac{2}{(1-r)^2} - \frac{1}{1-r} = \frac{1+r}{(1-r)^2}.\end{aligned}$$

Substituting $r = \left(\frac{D-1}{D+1}\right)$, we can see that

$$\begin{aligned}\sum p_z^* &\leq 2D\binom{k}{\frac{k}{2}+\frac{k}{2D}} \sum_{t=0}^{k/2-k/2D} (2t+1)\left(\frac{D-1}{D+1}\right)^t \\ &\leq 2D\binom{k}{k/2+k/2D}\frac{D(D+1)}{2} \\ &= D^2(1+D)\binom{k}{k/2+k/2D}\end{aligned}$$

$$\begin{aligned}\text{Hence, }\ \mu &= \frac{2^k}{\sum_z p_z^*} \geq \frac{2^k}{D^2(1+D)\binom{k}{k/2+k/2D}} \\ \mathsf{LP}_{\mathsf{dual}} &= (1+D)\mu \geq \frac{2^k}{D^2\binom{k}{k/2+k/2D}}\end{aligned}$$

Using Stirling's approximation, we get a lower bound of $\Omega\left(\frac{1}{D^2}2^{k(1-H(\frac{1}{2}(1+\frac{1}{D})))}\right)$. Note that the number of points $n = \Theta(2^{2k})$. For large constant D, this bound becomes $n^{\Omega(1/D^2)}$. For $D = (1+\epsilon)$ where ϵ is small, the bound becomes $n^{\frac{1}{2}-O(\epsilon\log(1/\epsilon))}$. This concludes the proof of Theorem 3.1.2. $\square$

We should note that our bounds are presented to make clear that constant distortion embeddings require polynomial dimension. Once $D \approx \sqrt{\log(n)}$ our bound does not say anything interesting. On the other hand, our estimate for $\binom{k}{k/2+k/2D}$ is a strict overestimate, so our bound does indeed hold for any D. We can deduce a few simple corollaries from this.

Corollary 3.1.9. *For any distortion $D = O((\log(n))^{c/2})$ where $c < 1$, at least $2^{\Omega(\log(n)^{1-c})}$ dimensions are required for the recursive diamond graph.*

In other words, super-logarithmic dimension is needed even when almost $\sqrt{\log(n)}$ distortion is allowed.

On the other hand, the result of Rao (Theorem 1.5.9, [65]) implies a distortion $D = O(\sqrt{\log(n)})$ embedding with dimension $O(\log(n))$ for any planar graph. This in turn implies that the dependence on $1/D^2$ in the exponent cannot be improved (say to $1/D$) for planar graphs.

Corollary 3.1.10. *The lower bound on dimension in terms of distortion cannot be improved to $n^{\Omega(1/D^{2-\delta})}$ for any $\delta > 0$ for any family of planar metrics.*

This is easy to see because any bound of this form would imply that embedding planar graphs with distortion $\sqrt{\log(n)}$ would require $2^{\Omega(\log(n)^{\delta/2})}$ dimensions, which is super-logarithmic for any constant $\delta > 0$.

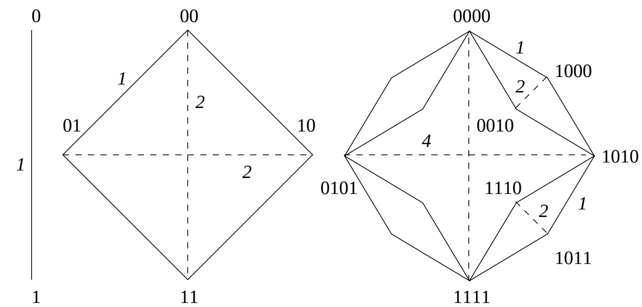

Figure 6: Using the diamond graph to generate our point set

3.2 Some ℓ_1 metrics require high dimension

So far we have proved that some series-parallel graphs do not admit low distortion, low dimension embeddings, while Theorem 1.6.6 of Gupta et al. [32] proves that series-parallel graphs can be embedded into ℓ_1 with constant distortion (with high dimension). This immediately implies the existence of ℓ_1 metrics which require high dimension. We can go one step further and provide an explicit family of point sets native to ℓ_1 which have the same properties as the recursive diamond graph. This gives our final theorem:

Theorem 3.2.1. *There are n point metrics in $\ell_1^{O(\sqrt{n})}$ which require $n^{\Omega(1/D^2)}$ dimensions if only D distortion is allowed.*

Proof. We build our point set with a construction analogous to the construction of the recursive diamond graph (see Figure 6). Let the original edge have end points at 0 and 1. Our "vertices" will be points in $\{0,1\}^i$ (that is, vertices of the hamming-cube). To go from level i to level $i+1$, first double the number of dimensions. The vertices of the parent edge are at the points u and v. Replace them with the points uu (u concatenated with u) and vv. The children will be the points uv and vu. The level-k

recursive diamond graph corresponds to a set of $\Theta(4^k)$ points in 2^k dimensions.

Claim 3.2.2. *Every "edge" in a level k point set has length 1.*

Proof. We prove this by induction on the level of the point set.

Base case: The original edge in the point set is between 0 and 1.

Inductive step: By the inductive hypothesis, the end points u and v of an edge at level i has length 1. The four child edges at level $i+1$ are (uu, uv), (uu, vu), (vv, uv) and (vv, vu). Since $||u - u||_1 = ||v - v||_1 = 0$ and $||u - v||_1 = 1$, $||uu - uv||_1 = ||uu - vu||_1 = ||vv - uv||_1 = ||vv - vu||_1 = 1$. □

Claim 3.2.3. *Each diagonal at level i has length 2^{k-i} in the level k point set.*

Proof. Again, we proceed by induction on the level of the point set.

Base case: The level 0 diagonal in the level 1 point set is from 01 to 10, which has length 2.

Inductive step: In the level j graph a level $i < j$ diagonal between points u and v has length 2^{j-i}. In the level $j+1$ graph these points are replaced with uu and vv. $||uu - vv||_1 = 2||u - v||_1 = 2^{j+1-i}$. The new diagonals in the level $j+1$ graph are at level j. A given new diagonal with parents u and v has end points uv and vu, and $||uv - vu||_1 = 2||u - v||_1 = 2 = 2^{j+1-j}$ by claim 3.2.2. □

If we divide all distances by 2^k this point set has exactly the same "edge lengths" and "diagonal lengths" as the recursive diamond graph. Since our constraints only depend on these distances, our lower bound for the recursive diamond graph immediately applies to this point set. □

As we noted at the end of Section 3.1, any distortion $D = (1 + \epsilon)$ (for ϵ small enough) requires $n^{\frac{1}{2} - O(\epsilon \log(1/\epsilon))}$. This is in stark contrast to the case of the Johnson-Lindenstrauss Lemma 1.6.1, where the dependence on ϵ is only $O(1/\epsilon^2)$.

3.2.1 The point set as an embedding of the diamond graph

As a brief aside, we would like to observe that the point set above is in fact an embedding of the diamond graph into ℓ_1 with distortion at most 2 and dimension $\Theta(\sqrt{n})$. We implicitly defined a mapping from points of the diamond graph into points of the Hamming cube by defining the construction of our ℓ_1 point set in analogy to the construction of the diamond graph.

Claim 3.2.4. *If two points u and v lie on a shortest path from the original top vertex to the original bottom vertex, then the distance between u and v is preserved exactly in this embedding.*

Proof. Consider every shortest path from the original top vertex (corresponding to $(0, 0, \ldots, 0)$) to the original bottom vertex (now $(1, 1, \ldots, 1)$). By Claim 3.2.2 we see that at each step exactly one coordinate switches from 0 to 1 (or vice versa). Since there are k dimensions and each shortest path from the top to bottom is length exactly k, this implies that at each step along the path exactly one "0" is switched to a "1," and then it is never switched back. This means the length of any path which lies wholly on a single shortest path from the top to the bottom is preserved exactly. $\square$

Let $\diamondsuit_x$ denote the diamond with label x. If u and v do not lie on the same shortest path, there must be some diamond $\diamondsuit_x$ such that u descends (w.l.o.g.) from the left side of $\diamondsuit_x$ and v from the right.

Claim 3.2.5. *If u and v descend from opposite sides of a diamond $\diamondsuit_x$ but lie on the same side of diag(x) (the diagonal of $\diamondsuit_x$), then their distance is preserved exactly in the embedding.*

Proof. Refer to the top vertex of $\diamondsuit_x$ as $\mathrm{top}(\diamondsuit_x)$, and similarly $\mathrm{bot}(\diamondsuit_x)$ is the bottom vertex, $\mathrm{left}(\diamondsuit_x)$ the left vertex and $\mathrm{right}(\diamondsuit_x)$ the right vertex. The vectors for $\mathrm{top}(\diamondsuit_x)$ and $\mathrm{bot}(\diamondsuit_x)$ differ at exactly p positions, where $\mathrm{top}(\diamondsuit_x)$ has $0s$ and $\mathrm{bot}(\diamondsuit_x)$ has $1s$. $\mathrm{left}(\diamondsuit_x)$ and $\mathrm{right}(\diamondsuit_x)$ differ from $\mathrm{top}(\diamondsuit_x)$ and $\mathrm{bot}(\diamondsuit_x)$ in exactly $\frac{p}{2}$ positions, and from each other in p positions by Claim 3.2.3 and Claim 3.2.4. Assume (w.l.o.g.) that u and v are both on the $\mathrm{top}(\diamondsuit_x)$ side of $\mathrm{diag}(x)$. Consider any shortest path from $\mathrm{left}(\diamondsuit_x)$ to $\mathrm{right}(\diamondsuit_x)$ that passes through both u and v. Then at each step from $\mathrm{left}(\diamondsuit_x)$ to $\mathrm{top}(\diamondsuit_x)$ we switch a 1 to a 0, and at each step from $\mathrm{top}(\diamondsuit_x)$ to $\mathrm{right}(\diamondsuit_x)$ we switch a 0 to a 1. But the positions where the 1 to 0 switches occur must be *disjoint* from the 0 to 1 switches, otherwise there would be a path from $\mathrm{left}(\diamondsuit_x)$ to $\mathrm{right}(\diamondsuit_x)$ shorter than p. Therefore, $||u-v||_1 = ||u-\mathrm{top}(\diamondsuit_x)||_1 + ||\mathrm{top}(\diamondsuit_x)-v||_1 = \Delta_G(u, \mathrm{top}(\diamondsuit_x)) + \Delta_G(\mathrm{top}(\diamondsuit_x), v)$ (where Δ_G is the distance in the graph), showing that the distance is preserved exactly. $\square$

There is only one remaining case.

Claim 3.2.6. *If u and v descend from opposite sides of a diamond $\diamondsuit_x$, and they also lie on opposite sides of diag(x), their distance is preserved up to a factor* 2.

Proof. In this case, $\frac{\Delta_G(\mathrm{top}(\diamondsuit_x), \mathrm{bot}(\diamondsuit_x))}{2} \leq \Delta_G(u,v) \leq \Delta_G(\mathrm{top}(\diamondsuit_x), \mathrm{bot}(\diamondsuit_x))$. Now assume (w.l.o.g.) that u is below $\mathrm{left}(\diamondsuit_x)$ and v is above $\mathrm{right}(\diamondsuit_x)$. u must have $1s$ at any position where $\mathrm{left}(\diamondsuit_x)$ has $1s$, and v must have $0s$ at any position where $\mathrm{right}(\diamondsuit_x)$ has $0s$ (by Claim 3.2.4). Also note that $\mathrm{left}(\diamondsuit_x)$ and $\mathrm{right}(\diamondsuit_x)$ have the same number of $1s$ and $0s$. This means there are $\frac{p}{2} = \frac{\Delta_G(\text{left}(\diamondsuit_x), \text{right}(\diamondsuit_x))}{2} = \frac{\Delta_G(\text{top}(\diamondsuit_x), \text{bot}(\diamondsuit_x))}{2}$ positions where left($\diamondsuit_x$) has $1s$ and right($\diamondsuit_x$) has $0s$. Since these positions cannot change for u and v, $||u - v||_1 \geq \frac{\Delta_G(\text{top}(\diamondsuit_x), \text{bot}(\diamondsuit_x))}{2} \geq \frac{\Delta_G(u,v)}{2}$. □

We conclude that this is indeed a distortion 2 embedding for the recursive diamond graph of k levels with dimension $d = 2^k \approx \sqrt{\frac{3}{2}n}$ where n is the number of vertices.

Chapter 4

Optimization techniques for embeddings

We have proved that there are some classes of ℓ_1 embeddable metrics which require polynomially many dimensions for any constant distortion. We now wish to look at how the result was actually proved. In the terms of Section 3.1.1, how did we derive the γ_{uv} and λ_{uv} to arrive at the final main inequality? Why did we decide to group the edges and diagonals? How did we arrive at the scheme for labeling edges and diagonals that made this possible? In this chapter we will explore these questions.

The reasons for doing this are two-fold. Firstly, our proof in its final form has many parts, and this chapter should aid the reader in making sense of them. More importantly, we believe that these techniques can be used in the future on difficult embedding questions. We have used linear program optimization and duality throughout. As we will see, this actually helped us to discover the combinatorial structure of the recursive diamond graph, and was a large reason for the success of our research. After we discuss our problem in specific we will discuss some general approaches to

$\min s$		
	$D\left(d_* + \sum_{i=0}^{k-1} \sum_{y \in \{0,1,2,3\}^i} d_y/2^i\right) - \gamma \sum_{x \in \{0,1,2,3\}^k} e'_x/2^k \geq k+1-\gamma$	(1)
$\forall x \in \{0,1,2,3\}^k$	$e_x \geq \frac{d_*}{2^{\lvert x\rvert}} + \sum_{y \sqsubset x} S(x_{\lvert y\rvert+1}) \frac{d_y}{2^{\lvert x\rvert-\lvert y\rvert}} + \sum_{y \sqsubset x} T(x_{\lvert y\rvert+1}) \frac{o_y}{2^{\lvert x\rvert-\lvert y\rvert-1}}$	(2a)
$\forall x \in \{0,1,2,3\}^k$	$e_x \geq -\frac{d_*}{2^{\lvert x\rvert}} - \sum_{y \sqsubset x} S(x_{\lvert y\rvert+1}) \frac{d_y}{2^{\lvert x\rvert-\lvert y\rvert}} - \sum_{y \sqsubset x} T(x_{\lvert y\rvert+1}) \frac{o_y}{2^{\lvert x\rvert-\lvert y\rvert-1}}$	(2b)
$\forall x \in \{0,1,2,3\}^k$	$-e_x \geq -\frac{s}{2^k}$	(3)

Table 10: Original linear program (no grouping)

using optimization in exploring ℓ_1, ℓ_2 and $(\ell_2)^2$ embeddings, and in Chapter 5 list some open problems that might be approached in this manner.

4.1 Discovering combinatorial structure via experiments

Recall that we derived a single linear constraint involving the distortion D. Let us call this constraint L, and leave the problem of finding the appropriate γ_{uv} and λ_{uv} for Section 4.2. Given the linear inequality L, we have written down the initial linear program we derived to minimize stretch (see Table 4.1). We can see in this linear program that inequality (1) is simply the D-distortion constraint L. Similarly, (3) is trivial, and simply comes from the definition of stretch. Equations (2a) and (2b) give the bound on the length of edge(x) in terms of x's ancestor diagonals and offsets. Deriving (2a) and (2b) takes a bit of work, which we did in Chapter 3.1. This linear program will then minimize the stretch needed to embed the recursive diamond graph into a single line, as described above.

Our next goal is to find a good feasible dual solution for this linear program. If we do not already have a good idea what the correct dual solution should be, how can we find it? In this case we actually generated explicit instances of this linear program for various sizes of graphs and various fixed D and solved them using the CPLEX solver. This approach has been considered in the past (e.g., in Andoni et al. [4] as discussed below), but it is often the case that the resulting optimization problem is infeasible. Consider our case: The level k graph has $\Theta(4^k)$ vertices, so both our number of variables and our number of constraints grows exponentially in k. We benefit from the fact that we have reduced our problem to a problem of embeddings into lines, but is it enough?

With the given linear program we were not able to derive the feasible dual solution we needed. The results that we got from small k actually did not seem to indicate polynomial dependence on n. Fortunately, however, the solutions to these small problems brought to light several combinatorial properties of optimal embeddings of our graph.

Property 1: $o_z = 0$. We discovered that in every case the o_z were always set to zero. Could we *prove* that the offsets were immaterial? We could not remove them without proof because, a priori, removing variables (or fixing them $= 0$) could actually raise the optimal s artificially.

Property 2: If $x \equiv y (mod\ 2)$, then $e_x = e_y$ and $d_x = d_y$. In other words, any two edges or diagonals whose labels were component-wise equivalent modulo 2 always got assigned the same length. This means that if we define

$$\overline{e_z} = \frac{1}{2^k} \sum_{\{x:x(mod\ \ 2)=z\}} |e_x|$$
$$\overline{d_z} = \frac{1}{2^i} \sum_{\{x:x(mod\ \ 2)=z\}} d_x$$

as we did in Section 3.1.3 and re-write our linear program in terms of $\overline{e_z}$ and $\overline{d_z}$ we should expect that the optimum is not changed.

Making this combination also causes the offsets to drop out of the linear program, and the result is our main linear program (see Table 6). Now by taking advantage of these two properties we have greatly simplified our linear program: Our number of variables has dropped from roughly $\frac{5}{3}4^k$ to roughly 2^{k+1}. This means that if the largest graphs we could embed before were on n vertices, we can now embed graphs on n^2 vertices. This allowed us to solve large enough LPs that the dual solutions could be used to derive our theorem.

Property 3: Given two labels $x, y \in \{0,1\}^i$ (that is, the same length), then if $||x||_1 = ||y||_1$, $e_x = e_y$ or $d_x = d_y$ (whichever is appropriate).

In other words, in the solutions provided by the CPLEX the average length of an edge or diagonal group depended only on the number of 1s in the label. This was the key insight that allowed us to derive our feasible dual solution given in Table 9.

In general, we may use mathematical optimization to generate optimal embeddings, and thereby discover combinatorial properties of the problem. We may then try to prove hypotheses based on these observations, and simplify our embedding task. We might also use our observations to simplify our linear program so that it provably can only give a *weaker* bound, but with the belief based on our observations that it will not *actually* weaken the resulting bound. By iterating this process of exploration and simplification we can take a problem that is poorly understood and perform experiments that help us develop an intuition. As we saw in Chapter 3.1 this process was very successful in resolving the ℓ_1 dimension reduction question.

4.2 Finding the distortion constraint

Recall that we wish to find a single linear inequality L of the form

$$\sum_{u,v\in X} \left(\gamma_{uv} NE_{uv} + \lambda_{uv} LD_{uv}\right),$$

with $\gamma_{uv}, \lambda_{uv} \geq 0$. It may be helpful for the reader to refer back to Section 3.1.1 at this point. How did we actually discover the weighting of γ_{uv} and λ_{uv} to arrive at our main distortion constraint? Is there any reason to believe that the γ and λ we selected are optimal? We will explore these issues now. We noticed one other property from our embedding experiments of the previous section:

Property 4: Edges only expanded, diagonals only contracted.

This means that γ_{uv} will be 0 for all diagonals (u, v), because the diagonals never run into the non-expansion constraint, and similarly λ_{uv} will be 0 for all edges (u, v). Now let us re-write our main linear program so that we make the use of the weights γ and λ explicit (see Table 11). Notice that for simplicity we have re-indexed the γ and λ to match the rest of the notation. In our main linear program (see Table 6) we set all the $\lambda_i = \lambda = 1$, and then optimized γ. In fact we derived the "correct" values of λ using another linear program. Consider what would happen if we were to break the main linear constraint into separate distortion constraints for the diagonals

Table 11: LP_A: The linear program with undetermined weights

$$
\begin{array}{|lll|}
\hline
\min s & & \\
\hline
D\left(\lambda\overline{d_*} + \sum_{i=0}^{k-1}\sum_{y\in\{0,1\}^i}\lambda_i\overline{d_y}\right) - \gamma\sum_{z\in\{0,1\}^k}\overline{e_z} & \geq \lambda + \sum_{i=0}^{k-1}\lambda_i - \gamma & [\mu] \\
\forall z\in\{0,1\}^k \quad \frac{s}{2^k} - \overline{e_z} & \geq 0 & [p_z] \\
\forall z\in\{0,1\}^k \quad \overline{e_z} + \left(\frac{\overline{d_*}}{2^k} + \sum_{y\sqsubset z}\frac{S(z_{|y|+1})\overline{d_y}}{2^{k-|y|}}\right) & \geq 0 & [\alpha_z] \\
\forall z\in\{0,1\}^k \quad \overline{e_z} - \left(\frac{\overline{d_*}}{2^k} + \sum_{y\sqsubset z}\frac{S(z_{|y|+1})\overline{d_y}}{2^{k-|y|}}\right) & \geq 0 & [\beta_z] \\
\hline
\end{array}
$$

Table 12: LP_B: The linear program with separate constraints for each level

$$
\begin{array}{|llll|}
\hline
\min s & & & \\
\hline
 & \sum_{z\in\{0,1\}^k}\overline{e_z} & \leq 1 & [\gamma] \\
 & D\overline{d_*} & \geq 1 & [\lambda] \\
\forall i\in[0,k-1] & D\sum_{y\in\{0,1\}^i}\overline{d_y} & \geq 1 & [\lambda_i] \\
\forall z\in\{0,1\}^k & \frac{s}{2^k} - \overline{e_z} & \geq 0 & [p_z] \\
\forall z\in\{0,1\}^k & \overline{e_z} + \left(\frac{\overline{d_*}}{2^k} + \sum_{y\sqsubset z}\frac{S(z_{|y|+1})\overline{d_y}}{2^{k-|y|}}\right) & \geq 0 & [\alpha_z] \\
\forall z\in\{0,1\}^k & \overline{e_z} - \left(\frac{\overline{d_*}}{2^k} + \sum_{y\sqsubset z}\frac{S(z_{|y|+1})\overline{d_y}}{2^{k-|y|}}\right) & \geq 0 & [\beta_z] \\
\hline
\end{array}
$$

on each level of the graph (see Table 12). As we will see, the choice of the names for the dual variables of LP_B to match the constants in LP_A is deliberate.

Consider the value of s optimized by LP_B. This will be the minimum stretch incurred by any embedding of the recursive diamond graph into a line under some distortion constraints, but a priori this does not seem to give a lower bound on the stretch needed for a general stretch-limited embedding. Recall that in Section 3.1.1 we argued that, because we had a single inequality relating D to the lengths of the edges and diagonals, we could focus on a single line embedding. In LP_B we do not have a single inequality involving D, but several inequalities, and so our convexity argument

Table 13: $DUAL_B$: γ,λ and the λ_i are variables

$$
\begin{array}{|llll|}
\hline
\multicolumn{4}{|l|}{\max \lambda + \sum_{i=0}^{k-1} \lambda_i - \gamma} \\
\hline
\forall z \in \{0,1\}^k & -\gamma - p_z + \alpha_z + \beta_z & \leq 0 & [\overline{e_z}] \\
 & \sum_{z \in \{0,1\}^k} p_z & \leq 2^k & [s] \\
\forall y \in \bigcup_{i \in [0,k-1]} \{0,1\}^i & D\lambda_i + \sum_{v \in \{0,1\}^{k-|y|}} \frac{S((yv)_{|y|+1})(\alpha_{yv} - \beta_{yv})}{2^{k-|y|}} & \leq 0 & [\overline{d_y}] \\
 & D\lambda + \sum_{z \in \{0,1\}^k} \frac{\alpha_z - \beta_z}{2^k} & \leq 0 & [\overline{d_*}] \\
\hline
\end{array}
$$

Table 14: $DUAL_A$: γ,λ and the λ_i are constants, μ is a variable

$$
\begin{array}{|llll|}
\hline
\multicolumn{4}{|l|}{\max \left(\lambda + \sum_{i=0}^{k-1} \lambda_i - \gamma\right) \mu} \\
\hline
\forall z \in \{0,1\}^k & -\gamma\mu - p_z + \alpha_z + \beta_z & \leq 0 & [\overline{e_z}] \\
 & \sum_{z \in \{0,1\}^k} p_z & \leq 2^k & [s] \\
\forall y \in \bigcup_{i \in [0,k-1]} \{0,1\}^i & D\lambda_i\mu + \sum_{v \in \{0,1\}^{k-|y|}} \frac{S((yv)_{|y|+1})(\alpha_{yv} - \beta_{yv})}{2^{k-|y|}} & \leq 0 & [\overline{d_y}] \\
 & D\lambda\mu + \sum_{z \in \{0,1\}^k} \frac{\alpha_z - \beta_z}{2^k} & \leq 0 & [\overline{d_*}] \\
\hline
\end{array}
$$

does not work. Nonetheless we will show that any feasible solution to $DUAL_B$ will give a feasible solution to $DUAL_A$ when the γ and λ are set to be the values given by the $DUAL_B$ (see Tables 14, 13).

In order to see this, simply set $\mu = 1$ and γ, λ and the λ_i to be the values of the corresponding variables in $DUAL_B$. It is very easy to check that the feasibility conditions are still met, and that the two programs have the same optimal value. This means that any feasible solution to $DUAL_B$ gives a lower bound on LP_A, and it provides us with the "correct" values of γ, λ and the λ_i in the process. This is in fact how we derived the main distortion constraint.

4.3 Lower bound techniques through semidefinite programming

One difficulty with our approach is that, in general, finding an optimal ℓ_1 embedding of n points via linear programming takes time exponential in n (see Section 1.5.3). In our case we worked around this problem by studying a counter-example that was very special, and had a high degree of symmetry. This allowed us to write a very efficient linear program for minimizing stretch, but finding such linear programs in general might be difficult. We now introduce an alternative lower bound technique that uses polynomially sized (in n) semidefinite programs. This results in programs that can be solved in polynomial time. It seems reasonable to expect that this technique should give weaker bounds than the LP approach because finding an optimal ℓ_1 embedding is known to be NP-hard [39, 43]. On the other hand, it may suffice for many applications.

It is not hard to see that any ℓ_1 metric can be embedded into a Hamming cube with arbitrary accuracy. Recall that in Section 1.5.3 it was claimed that any ℓ_1 metric is a linear combination of cut semi-metrics. For each cut C think of the coefficient of C, λ_C, as specifying the number of copies of that cut metric to use. This immediately gives an embedding into a Hamming cube (of arbitrary dimension), but it can have some error because the coefficients need not be whole numbers. We can get arbitrary accuracy by scaling the ℓ_1 metric up, thereby getting more precision at the cost of higher dimensionality.

Now consider the distance measure given by the square of the ℓ_2 norm, which we call $(\ell_2)^2$. In general this is not a metric, but on the set $\{0,1\}^d$, $(\ell_2)^2 \equiv \ell_1$, which implies that any finite ℓ_1 metric can be embedded into $(\ell_2)^2$ with arbitrarily low distortion. Any lower bound for the distortion of embedding a metric into $(\ell_2)^2$ therefore immediately implies a lower bound for embedding into ℓ_1. Though $(\ell_2)^2$ is not even a metric, optimal embeddings into $(\ell_2)^2$ may be $(1+\epsilon)$-approximated by semi-definite programming as in Theorem 1.5.15 in polynomial time. This approach to ℓ_1 lower bounds was mentioned by Andoni et al. [4], but was rejected for their purposes.

Chapter 5

Conclusions and research directions

The study of metric space embeddings is a very active research field, and many interesting questions remain open. In Chapter 1 we mentioned two popular open questions:

- **Conjecture 1.5.11**: Given a small fixed graph H, graphs from the class of H-minor free graphs embed into ℓ_1 with constant distortion.

- **Open Problem 1.5.14**: Can the optimal distortion for embedding a finite metric into ℓ_1 be approximated in polynomial time?

We refer the reader back to Chapter 1 for more discussion of these two problems. We would also like to direct the reader to the list of open problems from the Haifa Workshop on Discrete Metrics and Their Applications, [57]. This list, maintained by Jiri Matoušek, contains quite a large list of interesting problems about metric embeddings, and includes breaking news about new related results.

Our work suggests some new problems and has tantalizing connections to other well known questions, and we would like to conclude with a brief discussion of these issues.

5.1 Lower bounds

Following our work, Lee and Naor [48] have provided a very nice geometric proof of our result. In place of our linear program, they show that low dimensional ℓ_1 is isomorphic (up to a small constant factor) to low dimensional $\ell_{1+\delta}$ for δ small, and then prove that the recursive diamond graph requires high distortion in such $\ell_{1+\delta}$. They use a convexity inequality that generalizes the Short Diagonals Lemma 1.2.7 for ℓ_p with $1 < p \leq 2$.

The main question left open by our work is the following:

Open Problem 5.1.1 (Improved lower bound). *Can our lower bound trade-off for embedding $\ell_1^n \to \ell_1^d$ with distortion D be improved to $d = n^{\Omega(1/D)}$?*

As we noted in Corollary 3.1.10, an affirmative answer to this question cannot be proved using any family of planar metrics. On the other hand we do not know of any good candidate examples to improve the lower bound. We know that every metric can be embedded in ℓ_1^d with $d = O(\log(n))$ and $D = O(\log(n))$ by Bourgain [11] and Linial et al. [50] (see Theorem 1.5.2). This implies that any dependence on D better than $n^{\Omega(1/D)}$ cannot hold, though nothing is implied when $D = o(\log(n))$.

This question seems to be closely related to another question:

Open Problem 5.1.2. *What is the maximum, over all $M = (X, \ell_1)$, of $c_2(M)$ with $|X| = n$?*

This question was proposed by Linial [49] among others. In other words, what is the maximum distortion needed to embed any finite ℓ_1 metric into ℓ_2? Linial notes that the answer is known to be $\Omega(\sqrt{\log(n)})$ and $O(\log(n))$. This is related to Problem 5.1.1 because we may use it to reduce dimension in ℓ_1:

- Embed M into ℓ_2 (with distortion $c_2(M)$)
- Reduce dimension to $O(\log(n)/\epsilon^2)$ in ℓ_2
- Embed back into ℓ_1 (See the discussion of Theorem 1.5.2)

This gives a technique to reduce ℓ_1 dimensionality to $O(\log(n)/\epsilon^2)$ while incurring distortion $(1+\epsilon)c_2(M)$. If one could show that for all $M = (X, \ell_1)$, $c_2(M) = O(\sqrt{\log(n)})$, then this would imply that the $n^{\Omega(1/D^2)}$ dependence on D is the strongest possible for general finite metrics and arbitrary D.

5.2 Positive results for low-dimensional ℓ_1 embeddings

As we hinted in Section 2.2, it is in fact possible to use our techniques to give low-dimensional embeddings into ℓ_1 as well as to prove lower bounds. Recall that stretch characterizes ℓ_1 dimension up to a factor $D\log(n)$: We implicitly gave a sampling algorithm for creating a low dimensional ℓ_1 embedding from a low stretch embedding in the proof of Claim 2.1.2. In order to get a polynomial time algorithm for embedding a metric into low-dimensional ℓ_1 we now simply need an algorithm which generates a random line metric $f_\sigma(X)_i$ with probability w_i in polynomial time. The cost of taking $O(D\log(n))$ samples will increase the running time by only an $O(D\log(n))$ factor.

Now consider the specific case of our research on embedding series-parallel graphs into ℓ_1. Recall that we used CPLEX to generate stretch-optimal line embeddings of the recursive diamond graph. We used the dual solutions to our LPs extensively, but we have not yet examined the primal solutions. It is possible that by analyzing the primal solutions we will actually find an embedding that can be generalized to work for all series-parallel graphs. This would immediately give us embeddings of series-parallel graphs into ℓ_1 with ideal dimension/distortion trade-offs (up to $\log(n)$ factors), interpolating between the results of Rao [65] and Gupta et al. [32].

Open Problem 5.2.1. *Give a polynomial time algorithm which achieves a distortion D, dimension $n^{O(1/D^2)}$ trade-off for all metrics supported on series-parallel graphs when D is at least some universal constant c.*

Another interesting direction is to study the exact relationship between ℓ_1 embeddings and $(\ell_2)^2$ embeddings. Though there is a very nice duality theory for semi-definite programs which provides lower bounds (as noted in the previous chapter), it is not clear whether good embeddings into $(\ell_2)^2$ are of use in designing ℓ_1 embeddings. A number of authors (see [57]) have asked the following:

Open Problem 5.2.2. *Can any set of points under $(\ell_2)^2$ for which the triangle inequality is satisfied be embedded into ℓ_1 with constant distortion?*

Notice the connection with Open Problem 1.5.14. The question there was whether or not the optimal distortion for ℓ_1 embeddings could be well approximated. If the optimal distortion for embedding a metric into ℓ_1 *cannot* be approximated up to a constant factor in polynomial time, this would imply that $(\ell_2)^2$ with the triangle inequality must in fact be far from ℓ_1. This follows from the fact that optimal

$(\ell_2)^2$ embeddings with the triangle inequality can be found efficiently: The triangle inequality for each set of three points can be introduced as an extra constraint resulting in a polynomial (in n) number of new constraints, and this SDP can be solved in polynomial time.

If all finite $(\ell_2)^2$ metrics can be embedded into ℓ_1 with distortion at most a fixed universal constant, then this would give a polynomial time constant factor approximation for $c_1(M)$. Note that in this algorithm one never needs to actually embed $(\ell_2)^2$ metrics into ℓ_1, so the complexity of actually *finding* the embedding is not significant when approximating $c_1(M)$.

5.3 Conclusions

We have presented an approach to studying metric space embeddings into ℓ_1 that relies heavily on mathematical optimization techniques and duality. While these relationships were already well known, we were able to exploit them in a new way to demonstrate that some ℓ_1 metrics require a high number of dimensions. In order to accomplish this we refined Charikar and Sahai's [15] concept of the stretch of an ℓ_1 embedding and proved that there is a very close relationship between ℓ_1 dimension and stretch.

We have also tried to communicate the main elements of our "experimental" approach to working with embeddings. Not only did we give our lower bound via LP duality, but we were actually able to discover the "correct" form for the main distortion constraint through the use of LPs.

We hope and believe that our techniques may be used to discover other facts about ℓ_1 embeddings. In particular, a solution to Open Problem 5.2.1 might direct us to some new approaches to resolving Conjecture 1.5.11.

Bibliography

[1] D. Achlioptas. Database-friendly random projections. In *Proceedings of 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2001.

[2] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal of Optimization*, 5(1):13–51, 1995.

[3] N. Alon, R. Karp, D. Peleg, and D. West. A graph-theoretic game and its application to the k-server problem. *SIAM Journal of Computing*, 24(1):78–100, 1995.

[4] A. Andoni, M. Deza, A. Gupta, P. Indyk, and S. Raskhodnikova. Lower bounds for embedding edit distance into normed spaces. In *Proceedings of the 14th Annual ACM/SIAM Symposium on Discrete Algorithms*, pages 523–525, 2003.

[5] S. Arora and R. Kannan. Learning mixtures of arbitrary gaussians. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 247–257, 2001.

[6] R. Arriaga and S. Vempala. An algorithmic theory of learning: Robust concepts and random projection. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 616–623, 1999.

[7] P. Assouad. Plongements isométriques dans L^1: aspect analytique. *Séminaire d'Initiation à l'Analyse*, 14:1–23, 1979-1980.

[8] Y. Aumann and Y. Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM Journal on Computing*, 27(1):291–301, 1998.

[9] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 184–193, 1996.

[10] Y. Bartal, A. Blum, C. Burch, and A. Tomkins. A polylog(n)-competitive algorithm for metrical task systems. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 711–719, 1997.

[11] J. Bourgain. On Lipschitz embedding of finite metric spaces into Hilbert space. *Israel Journal of Mathematics*, 52:46–52, 1985.

[12] J. Bourgain, J. Lindenstrauss, and V. Milman. Approximation of zonoids by zonotopes. *Acta Mathematica*, 162:73–141, 1989.

[13] B. Brinkman and M. Charikar. On the impossibility of dimension reduction in ℓ_1. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 514–523, 2003.

[14] M. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 380–388, 2002.

[15] M. Charikar and A. Sahai. Dimension reduction in the ℓ_1 norm. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 551–560, 2002.

[16] C. Chekuri, A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair. Embedding k-outerplanar graphs into ℓ_1. In *Proceedings of the 12th Annual ACM/SIAM Symposium on Discrete Algorithms*, pages 527–536, 2003.

[17] V. Chepoi and B. Fichet. A note on circular decomposable metrics. *Geometriae Dedicata*, 69(3):237–240, 1998.

[18] S. Dasgupta. Learning mixtures of Gaussians. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 634–644, 1999.

[19] S. Dasgupta and A. Gupta. An elementary proof of the Johnson-Lindenstrauss lemma. Technical Report TR-99-06, International Computer Science Institute, UC Berkeley, 1999.

[20] M. Deza and M. Laurent. *Geometry of Cuts and Metrics*. Springer, 1997.

[21] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, NY, 2nd, electronic edition, 2000.

[22] P. Enflo. On the nonexistence of uniform homeomorphisms between L_p-spaces. *Arkiv för Mathematik*, 8:103–105, 1969.

[23] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 448–455, 2003.

[24] U. Feige. Approximating the bandwidth via volume respecting embeddings. *Journal of Computer and System Sciences*, 60(3):510–539, 2000.

[25] W. Fernandez de la Vega, M. Karpinski, C. Kenyon, and Y. Rabani. Approximation schemes for clustering problems. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 50–58, 2003.

[26] P. Frankl and H. Maehara. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *Journal of Combinatorial Theory, Series B*, 44:355–362, 1988.

[27] D. Gale, H. Kuhn, and A. Tucker. *Linear Programming and the Theory of Games*, pages 317–329. Wiley, New York, NY, 1951.

[28] C. Gavoille, D. Peleg, S. Perennes, and R. Raz. Distance labeling in graphs. In *Proceedings of the ACM/SIAM Symposium on Discrete Algorithms*, pages 210–219, 2001.

[29] A. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. Strauss. Fast, small-space algorithms for approximate histogram maintenance. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 389–398, 2002.

[30] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

[31] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer Verlag, 1993.

[32] A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair. Cuts, trees and ℓ_1 embeddings of graphs. In *Proceedings of the 40th Annual IEEE Conference on Foundations of Computer Science*, 1999.

[33] V. Guruswami and P. Indyk. Embeddings and non-approximability of geometric problems. In *Proceedings of the 14th Annual ACM/SIAM Symposium on Discrete Algorithms*, pages 537–538, 2003.

[34] P. Indyk. Dimensionality reduction techniques for proximity problems. In *Proceedings of the 9th Annual ACM/SIAM Symposium on Discrete Algorithms*, pages 371–378, 2000.

[35] P. Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 189–197, 2000.

[36] P. Indyk. Algorithmic applications of low-distortion embeddings. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 10–33, 2001.

[37] P. Indyk. Better algorithms for high-dimensional proximity problems via asymmetric embeddings. In *Proceedings of the 14th annual ACM/SIAM Symposium on Discrete Algorithms*, pages 539–545, 2003.

[38] P. Indyk. *Handbook of Discrete and Computational Geometry: Nearest Neighbors in High Dimensional Spaces*, volume 27 of *Discrete Mathematics and its Applications*, chapter 39. CRC Press LLC, second edition, 2004.

[39] P. Indyk and J. Matoušek. *Handbook of Discrete and Computational Geometry: Low Distortion Embeddings of Finite Metric Spaces*, volume 27 of *Discrete Mathematics and its Applications*, chapter 5. CRC Press LLC, second edition, 2004.

[40] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 604–613, 1998.

[41] W. Johnson and J. Lindenstrauss. Extensions of Lipschitz mapping into Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[42] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[43] A. Karzanov. Metrics and undirected cuts. *Mathematical Programming*, 32(2):183–198, 1985.

[44] L. Khachian. A polynomial algorithm in linear programming. *Dokl. Akad. Nauk SSSR*, 244:1093–1096, 1979.

[45] P. Klein, S. Plotkin, and S. Rao. Excluded minors, network decomposition and multicommodity flow. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 682–690, 1993.

[46] R. Krauthgamer, J. Lee, M. Mendel, and A. Naor. Measured descent: A new embedding method for finite metric spaces. Preprint, 2004. See http://www-sal.cs.uiuc.edu/ mendelma/mypapers/.

[47] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM Journal on Computing*, 30(2):457–474, 2000.

[48] J. Lee and A. Naor. Embedding the diamond graph in L_p and dimension reduction in L_1. *Geometric and Functional Analysis*, 2003. To appear, personal communication in June 2003.

[49] N. Linial. Finite metric spaces - combinatorics, geometry and algorithms. In *Proceedings of the International Congress of Mathematicians III*, pages 573–586, 2002.

[50] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15:215–245, 1995.

[51] N. Linial, A. Magen, and A. Naor. Girth and Euclidean distortion. *Geometric and Functional Analysis*, 12(2):380–394, 2002.

[52] L. Lovász and A. Schrijver. *Polyhedral Combinatorics*, volume 1 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, chapter Matrix cones, projection representations, and stable set polyhedra. American Mathematical Society, Providence, RI, 1989.

[53] L. Lovász and A. Schrijver. Cones of matrices and setfunctions, and 0-1 optimization. *SIAM Journal of Optimization*, (1):166–190, 1990.

[54] J. Matoušek. On the distortion required for embedding finite metric spaces into normed spaces. *Israel Journal of Mathematics*, 93:333–344, 1996.

[55] J. Matoušek. On embedding expanders into L_p spaces. *Israel Journal of Mathematics*, 102:189–197, 1997.

[56] J. Matoušek. *Lectures on Discrete Geometry*, volume 212 of *Graduate Texts in Mathematics*, chapter 15: Embedding finite metric spaces into Euclidean spaces. Springer Verlag, 2002.

[57] J. Matoušek. Workshop on discrete metric spaces and their algorithmic applications, Haifa: Open problems. Maintained online at http://kam.mff.cuni.cz/ matousek/haifaop.ps, 2002.

[58] M. Minsky and S. Papert. *Perceptrons - Expanded Edition: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, 1987.

[59] I. Newman and Y. Rabinovich. A lower bound on the distortion of embedding planar metrics into Euclidean space. In *Proceedings of the 18th Annual Symposium on Computational Geometry*, pages 94–96, 2002.

[60] R. Ostrovsky and Y. Rabani. Polynomial time approximation schemes for geometric k-clustering. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 349–358, 2000.

[61] C. Papadimitriou and M. Yannakakis. The traveling salesman problem with distances one and two. *Mathematics of Operations Research*, 18:1–11, 1993.

[62] D. Peleg. Proximity-preserving labeling schemes for graphs. In *Proceedings of the 25th International Workshop on Graph-Theoretic Concepts in Computer Science*,

volume 1665 of *Lecture Notes in Computer Science*, pages 30–41. Springer-Verlag, 1999.

[63] Y. Rabinovich and R. Raz. Lower bounds on the distortion of embedding finite metric spaces in graphs. *Discrete and Computational Geometry*, 19, 1998.

[64] M. Ramana, L. Tunçel, and H. Wolkowicz. Strong duality for semidefinite programming. *SIAM Journal on Optimization*, 7(3):641–662, 1997.

[65] S. Rao. Small distortion and volume preserving embeddings for planar and Euclidean metrics. In *Proceedings of the 15th Annual Symposium on Computational Geometry*, pages 300–306, 1999.

[66] R. Ravi. Metric embeddings and methods. Communicated in scribe notes for Carnegie Mellon University course 15-859(J), Lecture 2, by Hubert Chan., 2003. See the course web page at http://www-2.cs.cmu.edu/ anupamg/metrics/.

[67] G. Schechtman. More on embedding subspaces of L_p in ℓ_r^n. *Compositio Mathematica*, 61(2):159–169, 1987.

[68] M. Talagrand. Embedding subspaces of L_1 into ℓ_1^n. *Proceedings of the American Mathematical Society*, 108(2):363–369, 1990.

[69] L. Trevisan. When Hamming meets Euclid: The approximability of geometric TSP and MST. *SIAM Journal on Computing*, 30:475–485, 2001.