Algorithmic applications of low-distortion geometric embeddings

Piotr Indyk*

August 11, 2001

1 Introduction

In this paper we survey algorithmic results obtained using *low-distortion* embeddings of metric spaces into (mostly) normed spaces. Specifically, we will (mostly) consider mappings $f : P_A \to P_B$, such that

- P_A is a set of points in the (*original*) metric space, with distance function $D(\cdot, \cdot)$
- P_B is a set of points in the (*host*) normed space l_s^d
- for any $p, q \in P_A$ we have

  $$1/c \cdot D(p,q) \leq \|f(p) - f(q)\|_s \leq D(p,q)$$

  for a certain parameter c called *distortion*. We will allow more general definitions of distortion later.

During the last decade or so, low-distortion embeddings became recognized as a very powerful toolkit for designing efficient algorithms. Their usefulness comes from the fact that they enable us to reduce problems defined over "difficult" metrics to problems over "much simpler" metrics. Since many problems are defined purely in terms of metric properties of their input, embeddings form a natural and versatile paradigm for solving problems over metric spaces.

To illustrate this concept, consider the following *diameter* problem: given a set P of n points in l_1^d, find a pair of points $p, q \in P$ such that $\|p - q\|_1 = \max_{p',q' \in P} \|p' - q'\|_1$. The problem can be obviously solved in $O(dn^2)$ time, but this running time is not very exciting when, say, n is large but d is small. In the following we show that the input point set P (and in fact the whole space l_1^d) can be embedded into $l_\infty^{d'}$ such that $d' = 2^d$. The embedding (say f) has no distortion (i.e., $c = 1$) and can be computed in $O(ndd')$ time[1]. After applying f, it remains to solve the diameter problem in $l_\infty^{d'}$. This is an easy task, since

$$\begin{aligned}\max_{p,q \in P} \|f(p) - f(q)\|_\infty &= \max_{p,q \in P} \max_{i=1\ldots d'} |f(p)_i - f(q)_i| \\ &= \max_{i=1\ldots d'} \left(\max_{p \in P} f(p)_i - \min_{q \in P} f(q)_i \right)\end{aligned}$$

and therefore the diameter in $l_\infty^{d'}$ can be found in $O(nd')$ time, which implies a $O(ndd')$-time algorithm for computing the diameter in l_1^d.

It remains to construct the embedding $f : l_1^d \to l_\infty^{d'}$. We define $f(p)$ by specifying all of its d' coordinates. Specifically, for each vector $s \in \{-1, 1\}^d$, we define $f_s(p) = s \cdot p$. The value of $f(p)$ is a vector obtained by concatenating the values of $f_s(p)$ for all $s \in \{-1, +1\}^d$.

Why is f a no-distortion embedding ? Consider any pair of points $p, q \in l_1^d$. We need to show $\|p - q\|_1 = \|f(p) - f(q)\|_\infty$. Since f is linear, it is sufficient to show $\|p - q\|_1 = \|f(p - q)\|_\infty$. This, however, is easy to verify, since $\|x\|_1 = \text{sgn}(x) \cdot x$, where $\text{sgn}(x)_i$ contains the sign of x_i (i.e., sgn_i is equal to -1 if x_i is negative and equal to 1 otherwise).

Thus, we obtain a linear time algorithm for the diameter in l_1^d for the case when $d = O(1)$, by embedding the set P into l_∞ and solving the problem in the latter space.

Before we proceed further, we note that the embedding just constructed satisfies several very interesting properties:

- f is an *isometry*, i.e., $c = 1$.
- f is linear.
- f is *oblivious*, i.e., for any $p \in P$ the value of $f(p)$ does not depend on other points in P. This is a consequence of the fact that the domain of f is in fact the whole space l_1^d. Although this property is not important for the diameter application, it will be of large importance later when we consider data structure problems where some points are given on-line by the user and thus not all points are known in advance.

*Laboratory for Computer Science, MIT. E-mail: indyk@theory.lcs.mit.edu

[1] In fact, the running time can be reduced to $O(nd')$ by employing a more careful algorithm.

- f is *deterministic* - we will define *randomized* embeddings later.

- f is *explicit*, i.e., can be defined by a closed-form expression.

As we show later, all of the above properties become useful for some applications. Regrettably, the above embedding $f : l_1^d \to l_\infty^{d'}$ is the only embedding in this paper satisfying *all* of the above conditions.

1.1 Overview

Any embedding $f : A \to B$ can be classified based on the types of spaces A and B. As mention before, in this paper we deal almost exclusively with the case $B = l_{p'}^{d'}$. We will survey the known results, techniques and applications according to the following taxonomy:

1. A is a finite metric $M = (X, D)$ defined as a shortest path metric over a graph: these types of embeddings are considered in section 2. The main applications of such embeddings are approximation algorithms for optimization problems on graphs. Other applications include proximity-preserving labeling and proving hardness of approximation.

2. A is a subset of l_p^d: here we consider the following two scenarios:

   (a) $d' << d$ (and most often $p = p'$): we say that such an embedding results in *dimensionality reduction*. Such embeddings are described in section 3.1. They allow us to speed up algorithms whose running time depends on the dimension (section 3.2).

   (b) $p \neq p'$(and most often $d' >> d$): we describe them in section 3.3. They allow us to switch from "difficult" norms (e.g., l_2) to "easier" norms (e.g., l_∞). An example of such "internorm" embedding was presented in the Introduction.

3. A is a *special* metric, usually more general than a norm. In this survey we consider *edit metric* (defining similarity between strings of characters) and *Hausdorff metric* (defining similarity between *sets* of points). Such embeddings allow us to use algorithms designed for normed spaces to solve problems over the more difficult metrics. We describe the embeddings and their applications in section 4.

1.2 Disclaimer

The study of geometric representations of combinatorial structures (notably graphs) is a very wide area encompassing many disciplines. In this survey, however, we focus exclusively on geometric representations of *metrics* which achieve low distortion, as defined in the introduction. For a survey of many other ways of embedding combinatorial structures into geometric spaces, see [LV99].

1.3 Preliminaries

A *metric space* M (also called a *metric*) is a pair (X, D), where X is a set of *points* and $D : X \times X \to [0, \infty)$ is a *distance function* satisfying the following properties for $p, q, r \in X$:

- $D(p, q) = 0$ iff $p = q$
- $D(p, q) = D(q, p)$
- $D(p, r) + D(r, q) \geq D(p, q)$

A *ball* $B(p, r)$ of radius $r \geq 0$ around a point $p \in X$ is a set of all points q such that $D(q, p) \leq r$. Observe that a ball in a *finite* metric is just a finite set of points and therefore its cardinality is well defined.

We use l_p^d to denote $\Re^d$ under l_p norm. For any $x = (x_1, \ldots, x_d) \in \Re^d$, we use $\|x\|_p = (\sum_i |x_i|^p)^{1/p}$ to denote the l_p norm of x.

In this paper we use d exclusively to denote the dimension of a normed space, and n to denote the size $|X|$ of the metric space X. Whenever n or d appear without prior warning, their meaning should be interpreted according to the above rule.

2 Embeddings of finite metrics

In this section we focus on embedding finite metrics induced by graphs, e.g., obtained by computing all pairs shortest paths. We will focus mostly on embedding such metrics into norms; however, we will also address (probabilistic) embeddings into probabilistic trees.

2.1 Embeddings into norms

In this section we discuss known results on embedding finite metrics into normed spaces. We start from results on embedding general metric, after which we switch to results for more specific classes of metrics, e.g. metrics induced by planar graphs. For more detailed treatment of the topics considered here (including proofs) see Chapter 15 in [Mat].

The mother of all embeddings presented in this section, from both historical and "technological" point of view, is the following lemma by Bourgain [Bou85].

Lemma 1 *Any finite metric (X, D) can be embedded into l_2^d with $d < \infty$ with distortion $O(\log |X|)$.*

The original bound on d proved by Bourgain was exponential in n. However, it can be easily reduced to $O(\log^2 n)$, as shown by [LLR94] [2]. Since the latter proof uses probabilistic method, it immediately yields a polynomial time randomized algorithm which computes the desired embedding. The running time of the algorithm is $O(n^2 \log n)$, assuming that finding the value of $D(p, q)$ for any $p, q \in X$ takes unit time. The algorithm can be derandomized (preserving the polynomial time and the dimension) using the method of conditional probabilities (this result seems to be folklore). Alternatively, it can be derandomized using small sample spaces [LLR94]; that method however results in $d = \Theta(n^2)$. Yet another possibility is to observe [LLR94] that for any metric $M = (X, D)$ one can find, in deterministic polynomial time, an embedding $f : X \to l_2^n$ with distortion at most $1 + \epsilon$ times the smallest distortion achievable by any embedding of M into l_2; the algorithm uses Semidefinite Programming. The latter result implies that if a low-distortion embedding *exists* (i.e., via the original Bourgain's result), then it also can be *computed* in polynomial time.

The proof of Bourgain's lemma is not difficult, but somewhat technical. Therefore, we give a proof of a "sister" version of that lemma for the case of l_∞ norm. This lemma has been shown by Matoušek in [Mat96]. Its proof contains all the ideas needed for the proof of Bourgain's lemma, and is considerably shorter.

Lemma 2 *For any integer $b > 0$ let $c = 2b - 1$ and $M = (X, D)$ be any finite metric. Then M can be embedded into l_∞^d with distortion c, where $d = O(bn^{1/b} \log n)$.*

Before we proceed with the proof for the general c, consider first the case of $c = 1$. This case (considered by Frechet) has the following easy proof. Consider the mapping $f : X \to l_\infty^n$ defined as

$$f(q) =< D(p_1, q), \dots, D(p_n, q) >$$

where $X = \{p_1 \dots p_n\}$. Then we can write

$$\|f(p) - f(q)\|_\infty = \max_{p_i \in X} |D(p, p_i) - D(q, p_i)|$$

By using triangle inequality, $|D(p, p_i) - D(q, p_i)| \leq D(p, q)$ for each p_i, and therefore the mapping f is a contraction. On the other hand, for $p_i = p$, we have $|D(p, p_i) - D(q, p_i)| = D(q, p)$. Thus $\|f(p) - f(q)\|_\infty \geq D(p, q)$ and therefore f is an isometry.

Proof (sketch): The idea of the proof for the general case is similar. The main difference is that the individual points p_i are replaced by *sets* of points $A_i \subset X$, and the distance $D(q, p_i)$ is replaced by $D(q, A_i) = \min_{a \in A_i} D(q, a)$. The rest of the proof remains almost the same. We construct a mapping which is always a contraction (this part is again proved using the triangle inequality). At the same time, for any pair p, q, there is a "witness set" A_i which guarantees that f does not decrease the distance between p and q by too much.

Formally, the embedding f is defined as

$$f(q) =< D(q, A_1), \dots, D(q, A_d) >$$

for sets A_i to be determined later. One can easily verify that the mapping is a contraction, for any choice of the sets A_i. It remains to show that for each $p, q \in X$ we have $\|f(q) - f(p)\|_\infty \geq 1/c \cdot D(p, q)$. To this end, consider a specific pair $p, q \in X$. It is sufficient to make sure that at least one of sets (say, A_i) has the following two properties, for some $r > 0$:

1. A_i intersects the ball of radius r around q (or p, resp.)
2. A_i does not intersect the ball of radius $r + D(p, q)/c$ around p (or q, resp.)

Indeed, if such a set A_i exists, then $\|f(p) - f(q)\|_\infty \geq |D(p, A_i) - D(q, A_i)| \geq D(p, q)/c$. To find such A_i's (working for *all* pairs p, q), observe that if it happens that the cardinality of the ball $B_q = B(q, r + D(p, q)/c)$ is not much larger that the cardinality of the ball $B_p = B(p, r)$, and the two balls are disjoint, then A_i can be "constructed" using random sampling. Specifically, assume that each point in X is included in A_i with probability $\approx 1/|B_q|$. Then it is not difficult to show that the aforementioned two properties are satisfied (for A_i, p and q) with probability $p \approx \frac{|B_p|}{|B_q|}$. Thus repeating the process about $1/p \cdot \log n$ times succeeds in constructing a desired set A_i with high probability. Of course, to define p we need to know the value of $|B_q|$. However, it is sufficient to know it only approximately, and therefore we can construct A_i's for *all* possible approximate values of $|B_q|$.

To complete the argument, we need to show that for any (unordered) pair $p, q \in X$ there exists $r > 0$ such that $\frac{|B(q, r + D(p,q)/c)|}{|B(p, r)|}$ is small (at most $n^{1/b}$, to

[2] One can further reduce d to $O(\log n)$ using the results of section 3.1.

be specific). This is shown using the following "ball growing" argument. To start, take $r = 0$ so that $B(p,0) = \{p\}$. If $\frac{|B(q,D(p,q)/c)|}{|B(p,0)|} \leq n^{1/b}$, we are done. Otherwise, we know that $|B(q,D(p,q)/c)| > n^{1/b}$. In this case p,q switch the roles and we consider $\frac{|B(p,2 \cdot D(p,q)/c)|}{|B(q,D(p,q)/c)|}$ etc. If any of the ratios encountered during this process is small, we are done. Otherwise, after b steps, we have $|B(q,bD(p,q)/c)| > n$ (or a symmetric statement for p), which yields a contradiction.

□

Since for any vector $x \in \Re^d$ we have $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{d}\|x\|_\infty$, by setting $c = \log n$ in the above lemma we obtain that any finite metric (X,D) can be embedded into l_2^d with distortion $O(\log^2 |X|)$. This gives a weaker (but somewhat easier to prove) version of the Bourgain's lemma.

Lower bounds. The $O(\log n)$ distortion for embedding general metric into l_2 norm is tight [LLR94]. In fact, we know specific metrics which cannot be embedded with distortion $o(\log n)$, namely the shortest path metrics over *expander graphs*. For the case of l_∞, the situation is a bit more complex. Let $m(l,n)$ be the maximum number of simple edges in an n-vertex graph with *girth* (i.e., minimum cycle length) greater than l. Matoušek [Mat96] showed that for every fixed $c \geq 1$ and an integer $l > c$ there exist a metric which cannot be embedded with distortion c into l_∞^d for any $d = o(m(l,n)/n)$. The key points in his proof are: (a) there are "many" graphs with cycles length $> l$ (e.g., consider all edge subsets of a graph with high girth), (b) each embedding of a metric induced by any of the large-girth graphs has to be "different" if the distortion of the embedding is small and (c) there is only a "limited" number of "different" embeddings into a normed space with given dimension. Since it is known that for even l we have $m(l,n) = \Omega(n^{1+1/(l-1)})$, we obtain a lower bound for the embedding dimension. The bound can be improved for certain values of l; in particular, for $l = 4 \dots 7, 10, 11$ it matches the upper bound. It is conjectured [Erd64] that for $c = 2b$ we have $m(l,n) = \Theta(n^{1+1/b})$ (the $O(\cdot)$ part of the conjecture is known and easy to prove). If the conjecture is true, then the upper bound of the Matoušek's lemma is essentially tight. We also note that (unlike for l_2) the above proof does not provide an *explicit* metric that is hard to embed, even though constructions of dense graphs with high girth are known.

Other embeddings of graph-induced finite metrics. Bourgain's lemma was the starting point for the investigation of embeddings of graph-induced metrics into normed spaces. By now, many variants of that lemma have been discovered. A major research direction in this area have been *specialization*, i.e., constructing embeddings of restricted families of metric spaces with distortion better than $O(\log n)$. An important motivation here has been the following conjecture concerning embeddings of planar graph metrics. As we will see later in this section, a positive resolution of this conjecture implies efficient approximation algorithms for multicommodity flow problems.

Conjecture 1 *Let $G = (X,E)$ be a planar graph, and let $M = (X,D)$ be the shortest-path metric for the graph G. Then there is an embedding of M into l_1 with $O(1)$ distortion.*

The conjecture has been first stated in a published form in [GNRS99], but it has been known in the theory community for some time before that. There has been several results related to this conjecture. In particular, Rao [Rao99] gave an embedding of such metrics (into l_2)[3] with distortion $O(\sqrt{\log n})$, which improves the bound provided by Bourgain's lemma. His result in fact holds for any family of graphs which do not contain $K_{r,r}$ as a minor, for $r = O(1)$. Another result in that direction has been obtained in [GNRS99], who showed that any family of graphs excluding $K_{2,3}$ (or K_4) as a minor can be embedded into l_1 with constant distortion.

One could ask if the Conjecture 1 holds also for the l_2 norm. Surprisingly, Bourgain [Bou86] showed that a complete binary tree (clearly a planar graph) cannot be embedded into l_2 with distortion better than $O(\sqrt{\log \log n})$. This bound was shown to be tight for *all* trees by Matoušek [Mat99], in fact he showed that any tree metric can be embedded into l_p ($p \in (1,\infty)$) with distortion $O((\log \log n)^{\min(1/2,1/p)})$ as well as a matching lower bound.

Several other results on embedding trees into normed spaces are known. In particular, it is known [LLR94] that any tree can be embedded into l_1^n with *no* distortion. It can be also isometrically embedded into $l_\infty^{O(\log n)}$ [LLR94].

Finally, we mention some results on embedding metrics with distances 1 and 2 into low-dimensional l_p norms. A $(1,2)-B$ metric is a metric $M = (X,D)$ such that for any $p \in X$ the number of q's such that $D(p,q) = 1$ is at most B, and all other distances are equal to 2. It has been shown by Trevisan [Tre01] that any $(1,2)-B$ metric can be embedded into l_p^d, $1 \leq p < \infty$ with $d = O(B \log n)$. His definition of embeddings is somewhat complex and akin to the threshold embeddings discussed at the end of section 3.1. For the case of $p = \infty$ one can in fact

[3]This also implies an embedding into l_1, see section 3.3

prove that any $(1,2)-B$ metric can be isometrically embedded into $l_\infty^{O(B \log n)}$ (this was shown by the author).

Volume respecting embeddings. In a recent paper, Feige [Fei00] introduced the notion of *volume respecting* embeddings (into l_2). Such embeddings are significantly stronger than the embeddings we discussed so far and are defined as follows. For any k-point set $P \in l_2$ define $\mathrm{Evol}(P)$ to be the volume of the $k-1$-dimensional simplex spanned by the points in P. For any *metric* $M=(X,D)$, we define $\mathrm{Vol}(X)=\sup_{f:X\to l_2} \mathrm{Evol}(f(X))$, where f is required to be a contraction (otherwise the volume X is infinite). Now, given an embedding $f: X \to l_2$, we define the k-distortion of f to be

$$\sup_{P \subset X, |P|=k} \left(\frac{\mathrm{Vol}(P)}{\mathrm{Evol}(f(P))} \right)^{1/(k-1)}$$

If the distortion of f is D, we call f (k,D)-*volume respecting.*

Since for $X=\{p,q\}$ we have $\mathrm{Vol}(X)=D(p,q)$ and $\mathrm{Evol}(f(X))=\|f(p)-f(q)\|$, 2-distortion of an embedding is exactly the same as its distortion. For $k>2$, Feige showed an embedding (similar to the one used in the proof of Bourgain's lemma) which is $(k, O(\log n + \sqrt{k \log n \log k})$-volume respecting.

2.2 Applications of embeddings into norms

Approximation algorithms. The application that introduced finite metric embedding tools to theoretical computer science was the approximation algorithm for the *sparsest cut* problem [LLR94]. In this problem we are given an undirected graph $G=(V,E)$ with cost function $c: E \to \Re^+$; moreover, we are given a sequence of k "terminal" pairs $\{s_i,t_i\}$, together with *demands* $d(i)$, $i=1\ldots k$. The goal is to find $S \subset V$ which minimizes

$$\rho(S)=\frac{\sum_{u\in S, v \in V-S} c(\{u,v\})}{\sum_{i:s_i \in S, t_i \in V-S} d(i)}$$

In other words, we want to minimize the cost of the cut while maximizing the (weighted) number of separated pairs. The problem is NP-hard. The first algorithms for this problem were given by Rao and Leighton [Rao87, LR99] who showed a $O(\log n)$-approximation algorithm for the case when $d(i)=1$ and all pairs of vertices are terminal pairs. After a long sequence of improvements, Linial et al [LLR94] gave a $O(\log k)$-approximation algorithm for the general case, which is the best bound to date.

The algorithm of [LLR94] is based on Bourgain's lemma, and inherits its approximation ratio directly from the distortion guaranteed by that lemma. Since the definition of the sparsest cut involves a *graph*, not a *metric*, the use of the lemma is somewhat indirect. In a nutshell, [LLR94] observed that a (fractional) solution to a linear relaxation of a natural integer program for the sparsest cut can be viewed as a metric over V. Moreover, they observed that if that metric can be isometrically embedded into l_1, then one can compute (in polynomial time) an integral solution to the program with the cost equal to the cost of the fractional solution. Finally, they show that if the metric can be only approximately embedded into l_1 (say, with distortion c), then the cost of the integral solution is at most c times larger than the cost of the fractional solution. Thus Bourgain's lemma implies $O(\log n)$ factor approximation for the sparsest cut; small modification of this argument improves the bound to $O(\log k)$.

An interesting and useful feature of the aforementioned algorithm is that it specializes to specific classes of graphs. This means that, if for some class of (weighted) graphs it is known that the metrics induced by graphs from that class can be embedded with distortion c, then the approximation ratio of the algorithm is $O(c)$. Therefore, better approximation factors can be obtained e.g., for planar graphs. In fact, if (the algorithmic version of) Conjecture 1 holds, then there is a $O(1)$-approximation algorithm for sparsest cut for planar graphs, which gives a strong motivation to prove (or disprove) the conjecture.

To the knowledge of the author, the approximation algorithm for the sparsest cut problem constitutes the only application of Bourgain's lemma to optimization problems. However, additional approximation algorithms have been designed by using volume-respecting embeddings. The first problem successfully attacked using this tool was the *bandwidth* problem. The problem is again NP-hard. However, an approximation algorithm with $o(n)$ ratio turned out to be unusually difficult to discover, until Feige [Fei00] gave an algorithm with $\log^{O(1)} n$ approximation ratio (a $O(\sqrt{n})$-approximation algorithm was independently discovered in [BKRV98]). The description of the algorithm and its correctness proof is long and out of the scope of this paper. We only mention that, as before, the definition of the problem involves a graph (not a metric) and therefore the embedding theorem is again used as a building block of the algorithm, not as a reduction to a (simpler) version of the original problem. Other applications of volume respecting embeddings have been given in [Vem98].

Proximity-preserving labeling. Proximity-preserving labeling (introduced in [Pel99]) constitutes another application of embeddings (this time, into l_∞). The idea is to provide an algorithm which for any metric $M = (X, D)$ constructs a *label* function $f : X \to \{0,1\}^d$, such that given $f(p)$ and $f(q)$, for any $p, q \in X$, one can reconstruct the distance $D(p,q)$, possibly with some multiplicative error. To avoid discretization problems, we assume all distances are polynomially bounded integers. Since in such a case a label function always exists (e.g., one can take $f(p)$ equal to the binary representation of D), the goal is to minimize d, possibly as a function of the multiplicative error.

It is quite immediate that (approximate) embeddings discussed in earlier sections provide (approximate) solution to the labeling problem: one just needs to verify that the points-images of the embeddings have small integer coordinates and therefore can be represented using $O(\log n)$ bits. Interestingly enough, the labeling scheme obtained from Matoušek's lemma gives the best known bounds for general metrics !

The usefulness of embeddings is not restricted to the case of general metrics. In particular, the isometric embedding of trees into $l_\infty^{O(\log n)}$ provides the best possible labeling scheme for trees. However, the more general result of [GPPR01] stating that any family of graphs with vertex separator of size $c(n)$ has a labeling scheme with $d = O(c(n)\log n + \log^2 n)$, does not have (yet) a counterpart in the embedding world.

The connection between the embeddings and labeling schemes goes also in the other direction: the lower bounds for labeling schemes provide lower bounds for embeddings. In general this connection holds only for embeddings with integer coordinates. However, for the case of l_∞, Mihai Badoiu (personal communication) observed that if f is an isometry from a metric M with distances in the set $\{0, \ldots, M\}$ into l_∞^d, then there exists another isometry $f' : M \to l_\infty^d$ with coordinates in $\{0 \ldots M\}$. The isometry f' is obtained by rounding each coordinate of f to the nearest integer. Thus the lower bounds for labeling schemes for unweighted graphs provide lower bounds for embeddings into l_∞. In particular, it has been proved in [GPPR01] that any labeling scheme for trees must use $\Omega(\log^2 n)$ bits, which matches the aforementioned upper bound. Several other non-trivial lower bounds can be obtained in this way, including $\Omega(n^{1/2})$ bound for bounded degree graphs and $\Omega(n^{1/3})$ for bounded degree planar graphs.

Finally, we mention that for many applications one does not need the full power of proximity-preserving labeling schemes. Instead, one might just need a low-storage data structure which supports approximate distance queries. In this case, more efficient solution has been given by Thorup and Zwick [TZ01]. In particular, their data structure allows one to find an approximate distance between any two points in *constant* time, as opposed to $n^{\Omega(1)}$-time when the embeddings are used.

Hardness results. The embeddings of metric spaces with distances 1 and 2 into low-dimensional l_p norms have been used by Trevisan [Tre01] and the author to show hardness of an approximation of certain problems (notably TSP) in low-dimensional normed spaces. The hardness follows from the fact that many problems are known to be (quasi)-NP-hard to approximate up to certain constant in $(1,2) - B$ metrics, for $B = O(1)$. Thus, the existence of an approximation scheme for such problems over l_p^d norm, with running time $2^{2^{o(d)}}$, would imply subexponential-time algorithms for NP-hard problems.

Other applications. The concept of embedding finite metrics into Euclidean space is intriguing by itself. It is plausible that an interesting structure of the metric can be discovered by analyzing its embedding into low-dimensional spaces. For a case study see [LLTY97].

2.3 Embeddings into probabilistic trees

This section constitutes the only major departure from the main theme of this survey, in the sense that we address here the problem of embedding metric spaces into *convex combinations of trees*, instead of normed spaces. However, as we will see, there is a fairly close relation between these problems. In particular, since we know that trees (and therefore their convex combinations) can be embedded into l_1 with no distortion, the embeddings presented in this section work as well for the l_1 norm.

Formally, we will consider the following embeddings. Let $T = T_1 \ldots T_k$ be a sequence of (ordinary) metrics $T_i = (X, D_i)$, and let $\alpha = \alpha_1 \ldots \alpha_k$ be positive reals. Then T, α define a *probabilistic metric* $(X, \overline{D})$ via the formula $\overline{D}(p,q) = \sum_i \alpha_i D_i(p,q)$, for any $p, q \in X$. Without loss of generality we will assume that $\sum_i \alpha_i = 1$; in this way, we can think about $\overline{D}$ as the *expected* distance between p and q according to the distribution defined by α_i's.

For any finite metric $M = (Y, D)$ and probabilistic metric $(X, \overline{D})$, we say that an embedding $f : Y \to X$ has distortion c, if

1. f *never* contracts, i.e, $D_i(f(p), f(q)) \geq D(p,q)$ for all $p,q \in Y$ and $i = 1 \dots k$

2. f expands by at most a factor of c *on the average*, i.e., $\overline{D}(f(p), f(q)) \leq cD(p,q)$ for any $p,q \in Y$

The requirement (1) means that such embeddings are *stronger* than ordinary embeddings of (Y, D) into $(X, \overline{D})$, since the non-contraction property has to be satisfied by each of the metrics defining $(X, \overline{D})$. However, this stronger requirement is crucial for the applications, as we will see in a moment. To avoid confusion, we will say that such embeddings are *probabilistic* (as opposed to ordinary or deterministic ones).

The usefulness of probabilistic metrics comes from the fact that a sum of metrics is much more powerful than each individual metric. For example, it is not difficult to show that there are metrics (e.g., cycles [RR, Gup01]) which cannot be embedded into tree metrics with $o(n)$ distortion. In contrast, it is known than any finite metric can be embedded into a probabilistic metric over *trees metrics* with only polylogarithmic distortion ! The first result of this type has been obtained by Alon et al [AKPW91]. Their embeddings had distortion of $2^{O(\sqrt{\log n \log\log n})}$. A few years later Bartal [Bar96] improved the distortion to $O(\log^2 n)$ and later even to $O(\log n \log\log n)$ [Bar98]. He also showed wide applicability of such embeddings to many on-line and off-line problems (we will discuss some of them in the next section). In fact, Bartal's constructions employ trees of very special structure called *Hierarchically well-separated trees*, or *HST's*; this makes the task of designing algorithms for such trees even easier.

Below, we will show a weaker result which yields the bound of $O(\log^3 n \cdot \log\Delta)$, where Δ is the diameter of the metric, assuming the minimum interpoint distance in the metric is 1. Although weaker than the best known bound, the result has a very simple proof. The proof is a modification of the proof from [Bar96], with some additional ideas contributed by David Peleg, Ashish Goel and the author.

Proof: First, we embed (X, D) into l_∞^d with distortion $a = O(\log n)$, where $d = O(\log^2 n)$. This can be done by Matoušek's lemma from section 2.1. By proper scaling, we can assume that the embedding does not contract, and the distances are never expanded by more than a factor of a. Therefore, we can now assume that the metric (X, D) is induced by l_∞^d. We will show that such metrics can be embedded into probabilistic trees with $O(d \log\Delta)$ distortion. Then we will multiply the bound by $a = O(\log n)$ to get the final distortion.

Define an *l-partition* of X to be the set of disjoint *clusters* $X_1 \dots X_l$, whose union covers X and for any p,q from the same cluster we have $D(p,q) \leq l$. In l_∞, a d-dimensional grid with cubic cells of side length l naturally induces an l-partition of X (each X_i is defined as the set of points falling to the same cell, and we ignore empty clusters).

Since we need to define a *probabilistic* metric, we need to define *probabilistic* partitions. Specifically, an (r, ρ)-partition is a probability distribution over $r \cdot \rho$-partitions, such that for any $p,q \in X$ the probability $x(p,q)$ that p,q end up in different clusters is bounded from above by $D(p,q)/r$. Again, it is easy to see that for $\rho = d$, translating a cubic grid of side $r\rho$ by a random vector induces an (r, ρ)-partition.

Now we are ready to build the (probabilistic) tree. Instead of describing the distribution explicitly, we just show how to generate one tree at random from that distribution. We generate a random tree T recursively. Define $r_i = \Delta/2^i$. Firstly, we generate a random $r_0\rho$-partition from a probabilistic (r_0, ρ)-partition (recall that the latter is a distribution). In other words, we translate randomly a grid of side $r_0\rho$. Then, we recursively generate a random tree T_i for each partition set X_i, (using radii r_1, r_2 etc). Denote the root of T_i by u_i Now comes the crux of the construction: we create an artificial root u, and connect u to all u_i's with edges of length ρr_0. This defines our random tree T.

To make sure that the recursion ends, we stop recursion whenever the cluster to be partitioned contains only one element (there is really not much to partition there anyway). It is easy to see that the recursion depth is $O(\log\Delta)$. Also, one can observe that all of the original points of X became leaves of T.

Now we need to prove that the construction is correct. Firstly, we will take care of non-contraction. Let $D_T(p,q)$ be a distance induced by our tree for some p,q. Since r_i's decrease exponentially, there is always one r_i such that $r_i\rho < D(p,q)$ but $r_i\rho \geq D(p,q)/2$. Thus, at the i-th level of the tree the points p,q are separated, i.e., they belong to different trees, say T_p and T_q. Thus the only path from p and q has to go through the common ancestor u of T_p and T_q. But such a path will cost at least two times the length of any edge from u to its children, which is two times ρr_i, which is at least $D(p,q)$. Therefore, $D_T(p,q) \geq D(p,q)$ for any T, p, q.

The upper bound on $\overline{D}$ is only slightly harder to prove. Basically, for any tree T, the distance between $D(p,q)$ is dominated (up to a constant factor) by $2r_i\rho$, where i is the highest tree level on which p,q are still separated. We can bound the latter value from above by $2\rho \sum_i r_i I(i, T, p, q)$, where $I(i, T, p, q)$ is 1 if p and q are separated by the partition on the level i, and is 0 otherwise. The expected value of that sum is equal to

$$\begin{aligned} 2\rho \sum_i r_i x_i(p,q) &\leq 2\rho \sum_i r_i D(p,q)/r_i \\ &= O(\rho \log_2 \Delta \cdot D(p,q)) \end{aligned}$$

where x_i is the probability of a cut on the level i.

□

As we mentioned earlier, the trees generated by the above construction (and the constructions of Bartal) have very special structure. In particular, on each path from the root to the leaf, the lengths of the consecutive edges decrease exponentially (in our case by a factor of 2). Moreover, the distances from any node to all of its children are the same. This special structure makes the task of designing algorithms for such trees even easier.

A slight problem with the above construction is that the generated trees contain artificial nodes (i.e., nodes which do not belong to the original metric (Y, D). However, it is not very difficult to see that those nodes can be removed from the trees by increasing the distortion by only a constant factor. In fact, one can get rid of such nodes in *any* tree, as shown by Gupta [Gup01].

It is easy to see that a convex combination of tree metrics can be embedded into l_1 with no distortion. Thus $\Omega(\log n)$ is a lower bound for the best achievable distortion, since otherwise we could embed finite metrics into l_1 with distortion $o(\log n)$. This means that the upper and lower bounds are almost tight, modulo the $\log\log n$ factor. However, if the embedded metric $M = (X, D)$ is "special", better bounds are known. In particular, if M is induced by l_p^d norm, the distortion becomes $O(d^{\max(1/p, 1-1/p)} \log n)$ [CCG+98]. If M is a planar graph metric, one can achieve the distortion of $O(\log n)$ (Garg, Konjevod and Ravi, personal communication).

We also mention that the probabilistic metric can be found deterministically in polynomial time [CCG+98].

2.4 Applications of embeddings into probabilistic trees

The original motivation for the result of Alon et al was design of competitive *online algorithms*. An online algorithm performs actions in response to a sequence of requests *without* the knowledge of the future requests. Its performance is measured by dividing the cost incurred by the algorithm by the *optimal* cost of serving the requests by an "omniscient" algorithm with full knowledge of all requests, and taking the worst case of this ratio. An example of a classical on-line problem is the *metrical task systems* problem. The input to this problem consists of a metric space $M = (X, D)$, and a stream of *tasks*. Each task is an $|X|$-dimensional vector of non-negative reals, assigning *cost* to each point in X. The goal of the algorithm is to complete each task by maintaining one *server*. When a new task (say τ) comes, the algorithm can move the server from its current position (say x) to a new position (say y). The cost incurred by the algorithm in each such step is equal to $D(x, y) + \tau(y)$; the total cost is equal to the sum of the costs of all steps. The (highly nontrivial) decision which the algorithm has to make is choosing the right point y for each task.

Assume that the "adversary" who designs the worst case sequence of requests does not have the knowledge of the random bits chosen by our algorithm (or alternatively, the worst case input is prepared *before* the algorithm chooses its random bits). The crucial observation of Alon et al is that for many problems defined over metric spaces, embedding M into probabilistic metric $(Y, \overline{D})$ (defined, say, by metrics T_i and coefficients α_i) allows one to reduce the problem over M to the same problem over T_i ! Indeed, assume that we have an algorithm A for T_i's with competitive ratio C. To run the algorithm on the metric M, we choose one of the T_i's at random according to the distribution defined by α_i's. Then we run A on the metric M "pretending" it is the metric T_i.

Why and when does it work ? If the cost of a solution is defined only in "metric terms" (e.g., as in the case of metrical task systems), then we can use the following argument. Firstly observe that by the property (2) of probabilistic embeddings, an optimal solution (with cost S) for M is transformed into a solution for T_i with expected cost $\leq c \cdot S$. Therefore, the algorithm for T_i will produce a solution with expected cost at most $C \cdot c \cdot S$. By the property (1) of probabilistic embeddings, that solution induces a feasible solution for M with equal or smaller cost. Therefore, the resulting algorithm for M is $c \cdot C$-competitive (in the expected sense).

The above approach significantly simplifies the task of designing algorithms for many problems: instead of designing an algorithm for general metrics, it is sufficient to design an algorithm for (quite special) trees ! Thanks to this approach, several novel online algorithms have been discovered for problems for which solutions seemed out of reach using earlier methods [Bar96, Bar98, BBBT97]. In particular, Bartal et al [BBBT97] gave a $\log^{O(1)} n$-competitive algorithm for the aforementioned metrical task systems problem, solving a long-standing open problem. Their result has been further improved in [FM00], also using probabilistic embeddings.

Another area in which the probabilistic embeddings into trees turned out to be useful is the theory of *approximation algorithms.* In this case, the algorithm has full information about the input, but finding the optimum solution is NP-hard, so one needs to approximate it anyway. Since many problems are NP-hard for general metrics but polynomial time solvable for trees, it is not surprising that probabilistic embeddings led to best known approximation ratios for many problems, such as: *buy-at-bulk network design* [AA97], *group Steiner tree problem* [GKR98], *covering Steiner problem* [KR00], *metric labeling* [KT99, CKNZ01], *minimum cost routing tree problem* [WLB$^+$99], *hierarchical caching* [KPR99], *capacitated vehicle routing problem* [CR96], *concurrent distributed queueing* [HTW01] and *min-sum clustering* [BCR01]. In addition, the first nontrivial approximation ratios for *k-median* have been obtained through this method (the bounds have been improved since then).

We also mention that probabilistic embeddings into trees have been also used in the applied context [AS98, Sha98, JJJ$^+$00]. We elaborate more on those applications in section 5.

3 Embeddings of norms

In this section we leave the territory of (more or less) general finite metrics and instead focus on embedding normed spaces. The main difference between these two scenarios is that the embeddings described in this section map the *whole* normed space into the host space, as opposed to a *finite* set of points. This feature becomes extremely useful in many situations where the set of points of the embedded metric is not fully known in advance, e.g., when the points constitute a solution to an NP-hard problem, or when they are given on-line by the user at any point of time. The price for this feature is (a) lesser generality: the original space must have norm (or norm-like) structure, and (b) weaker guarantees: some of the embeddings have only probabilistic guarantees, or apply to only a selected range of distances. Despite of these limitations, embeddings of norms proved very useful in solving geometric problems, both in theory and practice.

3.1 Dimensionality reduction

The goal of the dimensionality reduction is to map a set of points in a high-dimensional space to a space with low dimension, while (approximately) preserving important characteristics of the pointset. In our case, the characteristics to be preserved (approximately) are all the pairwise distances between the points in the data set. It is intuitively clear (although not so easy to prove) that this task cannot be accomplished for all points in all pointsets, since otherwise we could pack the whole high-dimensional space into a lower-dimensional one. To avoid this problem, we will make the embedding *randomized.* Formally, we say that a *distribution* over mappings f from metric $M = (X, D)$ into metric $M' = (X', D')$ is a *randomized* embedding with distortion c and *failure probability* P, if for any pair of points $p, q \in X$ we have $D(p,q)/c \leq D(f(p), f(q)) \leq D(p,q)$ with probability $1 - P$. We also generalize the definition to the case when the lower bound for $D(f(p), f(q))$ holds with probability $1 - P_1$ and the upper bound holds with probability $1 - P_2$; in this case we call P_1 *the contraction probability* and P_2 *the expansion probability.* In the following, we will call the points $f(p)$ *sketches* of p.

It is instructive at this point to compare randomized embeddings with the probabilistic ones, introduced in earlier sections. Although both definitions refer to distributions over embeddings rather than deterministic embeddings, there are two important differences. Firstly, randomized embeddings provide bounds on the *probability* of low distortion, while probabilistic embeddings provide bounds on *expected* distortion. This implies that, e.g., in general we cannot "derandomize" randomized embeddings into l_1, as we did for the probabilistic embeddings. The second difference is that, in case of randomized embeddings, both the lower and the upper bound for $D(f(p), f(q))$ holds with certain probability, while for the probabilistic embeddings the lower bound holds *always.* Thus, the two definitions are substantially different. This said, there does not seem to be any reason why the first embedding is called "probabilistic" while the second is called "randomized" (reversing the names would work as well). However, since the definition of probabilistic embeddings seems already well established, we adopt the term "randomized" for embeddings described in this section.

The fundamental result on randomized dimensionality reduction was proved by Johnson and Lindenstrauss in 1984 (and rediscovered later in the applied setting in [Kas98]). It says that:

Lemma 3 *There is a randomized embedding from the space l_2^d into $l_2^{d'}$ with distortion $1+\epsilon$ and failure probability $e^{\Omega(-d'/\epsilon^2)}$.*

We mention that traditionally the Johnson-Lindenstrauss (JL) lemma is stated in terms of existence of an (ordinary) embedding of a set P of points in l_2^d into $l_2^{d'}$ with $d' = C \log n/\epsilon^2$, $C = O(1)$. This is a simple corollary of the above lemma, achieved by taking failure probability lower than $1/n^2$. However, as we will see later, it is crucial for most applications that the embedding is randomized, since (a) it allows us to guess a good embedding quickly and (b) choosing the embedding does not require prior knowledge of the points we want to embed.

Since the original proof of the above lemma (by Johnson and Lindenstrauss) was fairly complex and guaranteed only a large constant in the exponent, several simpler proofs appeared in the literature [FM88, IM98, DG99, AV99]. Below we present a proof by Motwani and the author, presented first in [IM98]. The formal statement of their version of the lemma is as follows.

Lemma 4 *Let u be a unit vector in $\Re^d$. For any even positive integer k, let $U_1, \dots, U_k$ be random vectors chosen independently from the d-dimensional Gaussian distribution*[4] *$N^d(0,1)$. For $X_i = u \cdot U_i$, define $W = W(u) = (X_1, \dots, X_k)$ and $L = L(u) = \|W\|_2^2$. Then, for any $\beta > 1$,*

1. *$E[L] = k$,*
2. *$\Pr[L \geq \beta k] < O(k) \times \exp(-\frac{k}{2}(\beta - (1 + \ln \beta)))$,*
3. *$\Pr[L \leq k/\beta] < O(k) \times \exp(-\frac{k}{2}(\beta^{-1} - (1 - \ln \beta)))$.*

In other words, the lemma shows that the length of the image of u under a random mapping is sharply concentrated around its mean. As we will see later, the choice of the normal distribution to define the mapping is not crucial and essentially any random mapping will do as well (this fact is often called "concentration of measure phenomenon").

We make a few observations before we proceed with the proof. Firstly, the mappings f defined above (as all other mappings in this section) are *linear*. This implies that $\|f(p) - f(q)\|_2 = \|f(p-q)\|_2 = \|p-q\|_2 \cdot \|f(u)\|_2$, where $u = (p-q)/\|p-q\|_2$ is a unit vector. Therefore, the above statement implies Lemma 3. Second, all entries in the matrix defining f are chosen independently from the same distribution. This will become crucial for some of the applications in section 3.2.

Proof: It is well known that a sum of random variables with normal distribution has itself a normal distribution, and that its variance is equal to the variances of the sum components. Since u is a unit vector, all X_i's follow the normal distribution with unit variance. Define $Y_i = X_{2i-1}^2 + X_{2i}^2$, for $i = 1, \dots, k/2$. Then, Y_i follows the exponential distribution with parameter $\lambda = \frac{1}{2}$ (see [Fel91, page 47]). Thus $E[L] = \sum_{i=1}^{k/2} E[Y_i] = (k/2) \times 2 = k$; also one can see that L follows the Gamma distribution with parameters $\alpha = \frac{1}{2}$ and $v = k/2$ (see [Fel91, page 46]). This distribution is a "dual" of the Poisson distribution, i.e.,

$$\Pr[L \geq \beta k] = \Pr[P_{\beta k}^{1/2} \leq v - 1],$$

where P_t^α is a random variable following the Poisson distribution with parameter αt. From the definition of Poisson distribution

$$\Pr[P_t^\alpha \leq v - 1] = \sum_{i=0}^{v-1} e^{-\alpha t} \frac{(\alpha t)^i}{i!}$$

and therefore

$$\begin{aligned}
\Pr[L \geq \beta k] &= \sum_{i=0}^{v-1} e^{-\beta v} \frac{(\beta v)^i}{i!} \\
&\leq v e^{-\beta v} \frac{(\beta v)^v}{v!} \\
&\leq v e^{-\beta v} \frac{(\beta v)^v}{\frac{v^v}{e^v}} \\
&= v (e^{-\beta} \beta e)^v = v e^{-v(\beta - (1 + \ln \beta))}
\end{aligned}$$

which implies (2) since $v = k/2$. The part (3) can be proved in a similar fashion.

$\square$

By plugging the bound from the above lemma into Lemma 3 while setting the failure probability to $\frac{1}{n^2}$, we obtain that the following corollary.

Corollary 1 *There exists a $(1+\epsilon)$-distortion embedding of n points in l_2^d into $l_2^{d'}$ with d' tending to $4 \ln n/\epsilon^2$ as ϵ tends to 0.*

This is the best (asymptotically) upper bound known so far. As shown in [EIO01], the mapping guaranteed by the above Corollary can be found in *deterministic* $O(n^2 d (\log n + 1/\epsilon)^{O(1)})$ time.

Although the proof of Lemma 4 explicitly uses the fact that the random entries in the embedding matrix are taken from normal distribution, it is fairly intuitive that some version of the lemma should hold even

[4] Each component is chosen independently from the standard normal distribution $N(0,1)$.

if we choose the entries from other "reasonable" distributions. This is indeed true; in fact the first proofs of JL lemma used matrix with random orthonormal vectors. Moreover, as shown in [Ach01], a version of the lemma holds even for the (probably) simplest possible random distribution, namely when each entry of the matrix is chosen independently and uniformly at random from the set $\{-1, 1\}$. In fact, [Ach01] also shows that it is sufficient to choose each entry to be 0 with probability $2/3$ and -1 or 1 with probability $1/6$ each; this allows the mapping to be computed several times faster. Interestingly enough, his bound for d' matches (or even slightly improves) the bound using normal random variables ! (although the proof is much more complicated).

It is not known if (although likely that) the upper bound for d' as in Corollary 1 is tight. The best lower bound so far is $\Omega(\log n/(\epsilon^2 \log(1/\epsilon)))$, which follows (Charikar and Matoušek, personal communication) from the lower bound for the following problem: Given n d'-dimensional unit vectors such that all pairwise dot products are at most ϵ in magnitude, what is the minimum value of d' ? Alon (personal communication) observed that $d' = \Omega(\log n/(\epsilon^2 \log(1/\epsilon)))$, which implies the aforementioned lower bound. Similar lower bounds can be also obtained for $p \neq 2$ (Charikar, personal communication).

Dimensionality reduction for l_p norms. Is it possible to prove an analog of Johnson-Lindenstrauss lemma for norms different than l_2? Surprisingly enough, the answer to this question is not known. However, it was shown in [Ind00b] that a somewhat weaker theorem indeed holds for l_1.

Lemma 5 *For any $1 > \epsilon, \delta > 0$ there is a randomized embedding from l_1^d into $l_1^{d'}$ with distortion $1+\epsilon$ and $d' = (\log 1/\delta)^{O(1/\epsilon)}$, with contraction probability δ and expansion probability $1-\epsilon$.*

We remark that the above lemma holds as well for any l_p norm, where $p \in [1, 2]$.

Note two major differences between the above lemma and JL lemma. Firstly, the dimension d' depends exponentially (not polynomially) on $1/\epsilon$. Second, the probability of (high) expansion is only slightly bounded away from 1. This implies that the above "asymmetric" lemma (unlike JL lemma) cannot be used to embed n point metric in l_1^d into a lower-dimensional space with small distortion; in fact no such result is known. Nevertheless, there are situations when we can afford the probability of expansion to be fairly high; in particular, it is the case when we want to preserve (approximately) the distance from a fixed point to its *nearest neighbor* in a set of many points.

The proof of Lemma 5 uses a tool of independent interest, namely *p-stable* distributions. A distribution over reals is called p-stable, if for any independent random variables X, Y, Z chosen from that distribution, and for any $a, b \in \Re$, the distribution of $aX + bY$ is the same as the distribution of $(|a|^p + |b|^p)^{1/p} Z$. Such distributions are known to exists for any $p \in (0, 2]$. In particular (as witnessed in the proof of Lemma 4) normal distribution is 2-stable. It is also known that *Cauchy* distribution is 1-stable, and *Levy* distribution is $1/2$-stable. Unfortunately, these are the only p-stable distributions whose densities can be expressed in a closed form. However, for any p there are algorithms for generating random numbers according to a p-stable distribution [CMS76].

The proof of Lemma 5 proceeds by choosing each entry in the embedding matrix (call it A) from a 1-stable (i.e., Cauchy) distribution. In this way we know that for any vector $x \in \Re^d$, each coordinate of Ax has Cauchy distribution as well, and thus $\|Ax\|_1$ is proportional to $\|x\|_1$. Unfortunately, Cauchy distribution does not have the mean, and therefore $\|Ax\|_1$ cannot be shown to be sharply concentrated around its (infinite) mean. However, one can nevertheless prove "one-sided" concentration lemma, i.e., that the left tail of the distribution of $\|Ax\|_1$ is exponentially small.

Dimensionality reduction for Hamming metric. The dimensionality reduction lemmas presented so far have been very "continuous", e.g., they were constructed between continues spaces using continuous distribution. However, as shown by Kushilevitz et al [KOR98], one can construct similar embeddings between *Hamming* spaces, which clearly are very discrete objects. To state their result, we need to introduce (yet another) more general definition of embeddings. We say that a distribution over mappings f from metric $M = (X, D)$ into metric $M' = (X', D')$ is a *randomized threshold* (r_1, r_2, r'_1, r'_2)-embedding with failure probability p, if for any pair of points $p, q \in X$ the following two properties are satisfied

- if $D(p, q) \leq r_1$ then $D(f(p), f(q)) \leq r'_1$
- if $D(p, q) > r_2$ then $D(f(p), f(q)) > r'_2$

with probability $1 - p$.

Armed with this definition, we can formally state the result of [KOR98] as follows.

Lemma 6 *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 d-dimensional Hamming metric into a d'-dimensional Hamming metric with failure probability P where $d' = O(\log(1/P)/\epsilon^2)$.*

The lemma was proved by constructing a random binary matrix A (with a proper density of 1's), and mapping each point p to Ap, where the addition is performed *modulo 2*. A slightly different (and non-linear) embedding was considered in [Ind00a], with the goal of simplifying the estimation of failure probabilities. Both versions can be modified so that the embeddings approximately preserve the R/r gap.

3.2 Application of dimensionality reduction

Among the embedding techniques discussed in this survey, dimensionality reduction techniques seem to have spun the largest number of algorithmic applications, at least so far. One explanation for this phenomenon seems to be that reducing the dimension of the input space has an obvious algorithmic advantage, since the running time of most geometric algorithms is at least proportional to the dimensionality of the space. Therefore, dimensionality reduction is a natural tool for designing efficient algorithms in high dimensions. It should be mentioned that for some of the applications the high-dimensional nature of the problem is not immediately apparent and discovering it is a major part of algorithm design process.

In the following we sketch the algorithmic applications of dimensionality reduction discovered so far. We classify them into 4 categories, depending on the way the dimensionality reduction is used:

- straightforward implications: results following from a straightforward composition of dimensionality reduction techniques and known geometric algorithms. The time/space requirements of the algorithms are reduced from $T(n, d)$ to $T(n, \log^{O(1)} n) + O(nd \log^{O(1)} n)$, which yields polynomial or exponential speedup, depending on the time bound $T(\cdot, \cdot)$.
- faster embedding computation: the results obtained by performing the dimensionality reduction in time better than $O(nd)$ (often in $O(n \log^{O(1)} n)$ time). This can be done in situations where the data set is defined *implicitly*, e.g., as a set of all d-length substrings of a given sequence of numbers.
- continuous clustering problems: the results for problems where the goal is to find a set of points $p_1, \ldots, p_k \in \Re^d$ minimizing certain function.
- sublinear-storage computation: the results that use the fact that dimensionality reduction allows one to use smaller *space* to represent a high-dimensional point (i.e., enables *input compression*).

We also mention two other applications of dimensionality reduction which do not fit into any of the above categories. The first of them is *robust learning*, introduced by Arriaga and Vempala [AV99]. That paper addresses the problem of learning geometric concepts (say, halfspaces), in situations where the positive and negative examples are far from the separating hyperplane. In such a case, they show that one can learn much more efficiently by embedding the concept in lower dimensional space and discovering the lower dimensional representation of the concept.

The second application, this time of the deterministic version of JL lemma [EIO01], is a (fairly) simple and efficient derandomization of approximation algorithms based on semidefinite programming.

Straightforward implications. A large class of geometric problems in high-dimensional spaces can be characterized as *proximity* problems. The input to such problems consists of a set P of points in $\Re^d$. The goal is to compute certain properties of P. However, the properties can only be defined in terms of the distances between the points in P, not the actual values of the coordinates of those points. For example, the problem of computing the closest pair, the furthest pair, minimum spanning tree, minimum cost matching and certain clustering problems belong to to the class of proximity problems.

It is fairly immediate that an approximate solution to a proximity problem can be found by applying the dimensionality reduction, and then solving the problem in the lower dimensional space. Although very simple, this approach enables to achieve $O(d/\log^{O(1)} n)$ (i.e., in many cases linear) speedup for the aforementioned proximity problems. In particular, it yields an $O(n^2 \log n/\epsilon^2 + nd \log n/\epsilon^2)$ $(1+\epsilon)$-approximation algorithm for closest pair and diameter in $\Re^d$. It is important to note that the aforementioned applications do not use any special properties of the dimensionality reduction procedure (like linearity etc).

We also mention that, by using a more careful approach (essentially, by varying the reduced dimension in the range $[1/\epsilon^2, O(\log n/\epsilon^2)]$, as opposed to keeping it fixed at $O(\log n/\epsilon^2)$) Kleinberg managed to obtain an improved running time of roughly $O(n^2/\epsilon^2)$ for closest pair as well as similar improvement for other problems (including nearest neighbor discussed below). Similar result for the diameter has been obtained in [FP99].

The aforementioned examples of the proximity problems involve *off-line* scenario, where the whole input is given before the computation starts. Another set of important problems involve the *on-line* case, where part of the input (a set of points P) is given in advance, and the goal is to create a data structure which, given a *query* q, reports, as quickly as possible, an answer to that query. A prototypical on-line problem is the *nearest neighbor search* (also called *post-office* or *best match* problem), in which case the answer to the query q consists of the point in P closest to q. In the approximate version of this problem, the data structure must report any point $p' \in P$ whose distance to q is at most $1+\epsilon$ times the distance from q to its nearest neighbor in P.

It is not difficult to observe that the dimensionality reduction techniques from section 3.1 can be used for the on-line proximity problems, and nearest neighbor search in particular. However, for this purpose, the dimensionality reducing embedding must be oblivious, since (a) we need to embed points from P without the knowledge of future queries q, and (b) we should be able to embed the query point q fast. Fortunately, randomized embeddings are sufficient for this purpose, since they guarantee high-probability of correctness for any fixed pair of points, including pairs $(q,p), p \in P$. Therefore, by reducing the dimension and building a nearest neighbor data structure for the lower-dimensional pointset, we obtain a $(1+\epsilon)$-approximate nearest neighbor data structure which is correct with a certain probability for a fixed query q. By employing several data structures in parallel, one can (with high probability) construct a data structure which is correct for *all* queries q [KOR98].

To apply the above approach we need the "base" nearest neighbor algorithms. It was shown in [IM98] that one can design a $(1+\epsilon)$-approximate nearest neighbor data structure for l_2^d with space $O(1/\epsilon)^d n \log^{O(1)} n$ and query time $(d + \log n + 1/\epsilon)^{O(1)}$. A simpler data structure with better bounds has been recently given in [HP01]. By applying JL lemma, we obtain a $(1+\epsilon)$-approximate nearest neighbor algorithm with similar query time but space bound $n^{O(\log(1/\epsilon)/\epsilon^2)}$. By using similar approach (but reducing the dimension in Hamming space, not in l_2), Kushilevitz et al [KOR98] obtained an algorithm with similar query time and slightly better space bound $n^{O(1/\epsilon^2)}$. Thus, both algorithms use space bounded by (pretty high-degree) polynomial in n and have very fast query time. Also, both of them can be modified to work under any l_p norm, for $p \in [1,2]$.

Faster embedding computation. In the above applications, the cost $O(dn \log^{O(1)} n)$ of computing the embedding was small compared to the cost of the rest of the algorithm. Below, we present several scenarios in which this is not the case, and where it is important to compute embedding much faster. Of course, this task is clearly not possible in general, since any embedding algorithm reading the whole input must take at least $O(dn)$ time. However, it *becomes* possible, if the set P of d-dimensional points is defined *implicitly*.

The first example of such situation involves the following *substring difference* problem (as well as many of its generalizations). Let $t[1 \ldots n]$ be a sequence of numbers from $\Re^d$. The goal is to build a data structure, which for any query pair $i, j \in \{1 \ldots n\}$, reports (quickly) the approximate value of $|t[i \ldots i+d] - t[j \ldots j+d]|_p$, e.g. for $p=2$, where $t[i \ldots i+d]$ denotes the $d+1$-dimensional vector $(t[i], \ldots, t[i+d])$.

A naive solution for this problem is to recompute, for each query, the distance between the two substrings. However, this would take $O(d)$ time per query. A faster approximate solution can be obtained using dimensionality reduction. In particular, we can compute $O(\log n/\epsilon^2)$-length sketches of all d-length substrings of t, and then compute the distance using the sketches. However, a naive algorithm computing the sketches would run in $\Omega(dn)$ time, which is quadratic in n for large d. Fortunately, the embeddings we use are *linear*. This means that, given a random embedding matrix $A = [a_1 \ldots a_k]^T$, $a_i \in \Re^d$, $k = O(\log n/\epsilon^2)$, we need to compute all inner products $a_i \cdot t[j \ldots j+d]$. This, however, can be performed in $O(n \log n)$ time per a_i using Fast Fourier Transform ! Therefore, we can reduce the total time needed to build the data structure to $O(nk \log n)$, and achieve a quadratic speed-up (in the best case).

The above technique (effectively presented in [Ind98a]) has been extended in [IKM00] to support arbitrary values of d given at the query time, while using $O(n \log^{O(1)} n)$ storage. This in turn has been used to give fast algorithm for finding *approximate period* of a sequence, with applications to data mining time sequences. Another application of this technique is a fast preprocessing for the *nearest substring problem*, which is a version of the nearest neighbor problem for substrings. Since we will see later in the section that the c-approximate nearest neighbor for l_2 can be solved using $O(dn^{1+1/c})$ storage/preprocessing time and $O(dn^{1/c})$ query time, the above approach yields an improvement in the preprocessing and query time.

It is interesting to observe a parallel between the above "sketching" technique and "fingerprinting" technique of Karp and Rabin. The fingerprints of Karp-Rabin have the following two properties:

1. They can be computed for all d-substrings of a text t in $O(n)$ time
2. With high probability, two substrings are equal iff their fingerprints are equal

The sketching approach guarantees similar properties, except that the time needed to compute the sketches in $O(n \log n)$, and the word "equal" is replaced by "similar" in Property 2. Since Karp-Rabin fingerprints found many applications for the "exact" combinatorial pattern matching problems, it is not surprising that the sketching techniques are applicable to the "noisy" versions of those problems.

We briefly mention another situation where the embedding can (and should) be computed faster than in $O(dn)$ time. Consider the following problem: given n points and n $(d-1)$-dimensional hyperplanes in $\Re^d$, find a tree spanning the points such that the tree edges cross as few input hyperplanes as possible (for motivation, see [HPI00]). This problem can be solved exactly in quadratic time; however, approximate solution is good enough. Har-Peled and the author observed that the "crossing" metric can be isometrically embedded into the square of l_2^n. Therefore, we could apply JL lemma, reduce the dimensionality to $O(\log n)$, and then use approximate MST algorithm from [IM98]. Unfortunately, naive dimensionality reduction would cost $\Omega(n^2)$ time. However, it turns out that since the metric is implicitly defined by only $O(n)$ parameters, a variant of the embedding can be performed in roughly $n^{2-1/d}$ time using low-dimensional geometric tools.

Continuous clustering problems. The goal of continuous clustering problems is to to find a set of points $c_1 \dots c_k \in \Re^d$ minimizing certain function defined by the input set P of points. For example, one could try to minimize $\sum_{p \in P} \min_i D(p, c_i)$ (this variant is called *k-median*) or $\max_{p \in P} \min_i D(p, c_i)$ (*k-center*); the sum of *squares* of the distances is also widely used in practice.

If the centers c_i have to be chosen from a predefined set of points, using dimensionality reduction to solve (or improve solutions of) the above problems is straightforward. However, for the continuous clustering problems, a center can occupy *any* point in $\Re^d$. Thus it is not clear how to use the information obtained from solving the problem in the lower-dimensional space, since there is no easy method of "pulling back" the lower dimensional points into the high dimensional space. Therefore, special methods need to be developed to take advantage of dimensionality reduction techniques.

The first result of this type has been obtained by Dasgupta [Das99] (see also recent improvements by Arora and Kannan [AK01]) who gave a fast algorithm for unsupervised learning of Gaussian mixtures. For the special case of unit variances and equal mixture weights, this problem can be viewed as finding k centers which minimize the sum, over all input points p, of squares of distances from p to its nearest center. It is assumed that the points p are sampled at random from the mixture itself, instead of being chosen by an adversary.

The main idea of the algorithm of [Das99] is to embed the data points into $d' = O(\log k/\epsilon^2)$-dimensional space, identify a "large" cluster C of points by exhaustive search in d'-dimensional space, and then find the center for C in the *original* d-dimensional space. After that, the cluster C is removed from the data set and the process is applied recursively. Thus the lower-dimensional representation of the input points is only used to identify the structure of the clustering, not its parameters. However, knowing the structure of the clustering is sufficient to reduce the complexity of the problem.

A similar approach was used by Ostrovsky and Rabani [OR00] to obtain an $n^{(k+1/\epsilon)^{O(1)}}$-time $(1+\epsilon)$-approximate algorithm for the continuous k-median problem. In this case, they reduce the dimension to $O(\log n/\epsilon^2)$ and perform the exhaustive search for k medians in the lower-dimensional space[5]. Although it does not seem to be mentioned anywhere, a similar result (using same techniques) can be also obtained for the continuous k-center problem.

Data structures with sublinear storage. One can observe that the "straightforward" applications of dimensionality reduction described earlier achieve not only reduction of the running time but also of the required storage, since they allow us to keep $O(n \log n)$ (instead of nd) coordinates. However, there are problems for which even more substantial (even exponential) storage reduction can be achieved. A prototypical example of such problems is a data structure which maintains a d-dimensional vector x (under increments/decrements of x's coordinates). When queried, the data structure reports an approximate value of $\|x\|_p$. Such problems have been investigated since early 80's. In particular, the case of $p = 0$, corresponding to maintaining an approximate number of non-zero coordinates has been addressed

[5]Technically, they use dimensionality reduction in Hamming space, not l_2, but it does not seem to be crucial for their results.

in [FM88]. The motivation for this problem (as well as most other problems described below) comes from massive databases. In this case the ith coordinate can be thought of as the number of elements of type i; the coordinate is updated whenever an element is added or deleted. The number of non-zero elements in such a vector corresponds to the number of different types of elements, an important parameter for query optimization and approximate query answering.

The paper [FM88] presents a randomized algorithm which maintains an $(1+\epsilon)$-approximate value of $\|x\|_0$ under *increments*, with probability $\delta > 0$ of error,using $O(\log(1/\delta)\log n/\epsilon^2)$ *bits* of space[6]. The algorithm can also handle *decrements*, as long as all coordinates of x are non-negative; in this case the storage gets multiplied by $O(\log n)$. Both algorithms (as well as all algorithms described below) have constant probability of correctness.

Alon et al [AMS96] were first to consider the problem of maintaining $\|x\|_p$ for general (integer) $p \geq 1$. In that seminal paper, they show that sublinear storage can be achieved (with insertions only) for any fixed value of p. They also gave several lower bounds for the required storage, in particular a linear lower bound for the case $p = \infty$. Of special interest was their algorithm for the case $p = 2$, which achieved $O(\log(1/\delta)\log n/\epsilon^2)$ space for both insertions and deletions. Maintaining l_2 norm of the count vector is an important problem in databases, since it allows one to maintain the size of a self-join of a relation.

Their l_2 algorithm proceeds by maintaining the value of Ax, where A is a $d' \times d$ matrix with random entries from $\{-1, 1\}$, for $d' = O(\log(1/\delta)/\epsilon^2)$. Thus, it is reminiscent of the mapping used for JL lemma. However, the values in each row are only 4-wise independent (to make sure the random bits can be stored in small space as well) and therefore $\|x\|_2$ cannot be estimated from $\|Ax\|_2$ since the latter random variable is not sharply concentrated. Instead, they use median estimator to improve the probability of correctness. Nevertheless, the mapping producing the sketch of x is linear, which implies that we can implement any linear operation on the vectors (addition, subtraction etc) as linear operations on their sketches.

An polylogarithmic space algorithm for any[7] $p \in (0, 2]$ has been given by the author [Ind00b][8]. Again, the algorithm uses linear mapping and therefore enables to perform arbitrary linear operations on the vectors. As such, it improves earlier results by [FKSV99, FS00] for $p \geq 1$ (which worked for certain restricted scenarios) and provides a general solution for any $p \in [0, 2]$. The algorithm is explicitly based on linear embeddings, and uses p-stable distributions (defined earlier in this section) together with median estimator; the latter can be replaced by a sum for $p = 2$. In order to be able to generate the matrix A in low space, [Ind00b] used Nisan's pseudorandom generator to generate entries of A. The generator uses $O(\log^2 n)$ random bits as a seed, and this quantity dominates the space bound of the algorithm, which is $O(\log^2 n + \log(1/\delta)\log n/\epsilon^2)$.

This concludes the description of the norm-maintaining data structures. A natural question arises: is there any other information (apart from the norm) about the vectors which can be dynamically maintained in small space ? For example, is it possible to maintain short sketches $S(x), S(y)$ of non-negative vectors x, y, so that we can estimate the value of $x \cdot y$ up to a factor of $(1+\epsilon)$ (for any $\epsilon > 0$) from the sketches ? If so, this would enable fast estimation of size of a join of two relations, a "holy grail" of database query optimization research during last few years. Unfortunately, it is not difficult to show (using communication complexity tools) that such (short) sketches do not exist in general. However, the vectors x arising in applications are not adversarially chosen. In fact, they usually can be approximated by a low-complexity piecewise-constant function [JKM+98] or as a sum of few wavelet vectors [MVW98]. Therefore, maintaining such approximations of the input vector is of large interest. Can it be done ? The answer turns out to be yes, by the following argument. Let N be the number of different low-complexity approximations of the vector (say, the number of piece-wise constant functions with k pieces). One can verify that $N = n^{O(k)}$. Therefore, if we maintain a linear sketch Ax of x of length $O(\log N) = O(k \log n)$, then with high probability, for *all* low-complexity approximations y of x we have $\|A(x-y)\| \approx \|x-y\|$, and therefore we can recover y approximately closest to x !

The problem with naive implementation of the above idea is that exhaustive enumeration of all y's takes time exponential in k. However, in case of the wavelet-based histograms, the orthogonality property of wavelet basis can be used to estimate the wavelet coefficients (from a sketch) in $O(n)$ time [GKMS01]. In [GGK+01] the running time (for both wavelet and

[6]For simplicity, we assume that the values of the coordinates are taken from $\{1 \dots n^{O(1)}\}$.

[7]The proceedings version of the paper gives proofs only for integer values of p, but the general case has been subsequently verified by the author.

[8]The case $p = 0$ can also be handled by taking sufficiently small p, as observed by Cormode, Muthukrishnan and the author.

piecewise constant histograms) has been further reduced to $(\log n + 1/\epsilon + k)^{O(1)}$.

3.3 Switching norms

In this section we address the second class of embeddings between norms, namely embeddings between *different* norms (in most cases from l_p to $l_{p'}$. As in the case of dimensionality reduction, the usefulness of such embeddings comes from the fact that the host norm is "easier" than the original norm. The embedding of l_1 into l_∞ presented in the introduction is a perfect example of such embeddings. In the following we will show more examples and applications of this type. Most of them are classical results (or corollaries of such) in the *geometric functional analysis* (also called *local theory of Banach spaces*). For more thorough and detailed treatment of this type of results, see the survey article by Lindenstrauss and Milman [LM].

Embeddings into l_1. We start from embeddings of l_p norms, with $p \in (1, 2]$, into l_1. In particular, we address first the case $p = 2$. From the celebrated Dvoretzky theorem [Dvo59] it follows that for any d and $\epsilon > 0$ there exists $d' = d'(d, \epsilon)$ such that l_2^d can be embedded into $X = l_1^{d'}$ with distortion $1 + \epsilon$; in fact, Dvoretzky's theorem shows it is true for *any* sufficiently dimensional Banach space X. However, for the special case of $X = l_1$ one can obtain much better dependence of d' on d and ϵ than for the general case. In particular, it was shown in [FLM77] that one can achieve $d' = O(d \log(1/\epsilon)/\epsilon^2)$ (see that paper for references to earlier, somewhat weaker, estimates). As most of the proofs seen so far, the proof of [FLM77] is probabilistic. Specifically, they show that there is a randomized embedding of l_2^d into $l_1^{O(\log(1/\delta)/\epsilon^2)}$ with distortion $1 + \epsilon$ and failure probability δ (as before, the proof uses a random linear embedding). Then, it is shown that in order to extend the mapping to the whole space l_2^d, it is sufficient to ensure the mapping has distortion (say) $1 + \epsilon/4$ for the points in an $\epsilon/2$-net of a unit sphere in l_2^d (recall that an ϵ-net for a set S is a set S' such that for any $p \in S$ there is $p' \in S'$ within distance ϵ from p). Since it is easy to show that such ϵ-nets of size $(1/\epsilon)^{O(d)}$ exists, the theorem follows.

In the original proof, the rows of the embedding matrix were chosen to be random orthonormal vectors. A different proof, which uses a matrix with all entries independently chosen from normal distribution, has been given in [Ind00b]. We also mention that Schechtman [Sch81] showed that it is possible to obtain constant distortion and $d' = O(d)$ using random matrix with entries chosen independently and uniformly at random from $\{-1, 1\}$. His proof technique does not seem to extend to arbitrarily small distortions $1 + \epsilon$.

All of the above proofs are probabilistic in nature. Although this fact is fairly typical (e.g. for the results presented in this survey), it should be noted that the situation here is quite different, since we want to embed a *fixed* metric l_2^d (as opposed to an "input" metric (X, D)). In particular, this means that we *could* potentially achieve an explicit (closed-form) description of the embedding, as opposed to the general metric embedding or dimensionality reduction. This naturally leads to the question if the probabilistic bounds mentioned above can be matched using explicit construction. Although this question still remains unsolved, some progress in that direction has been made. In particular, as noted in [LLR94], the results of Berger [Ber97] imply that there is a constant distortion embedding of l_2^d into $l_1^{O(d^2)}$. Her proof uses embedding matrix obtained by concatenating (horizontally) all vectors from a sample space generating d four-wise independent random variables with values in $\{-1, 1\}$. In a similar way, Indyk [Ind00b] used Nisan's pseudorandom generator to derandomize the aforementioned probabilistic proof using normal random variables. The resulting embedding maps l_2^d into $l_1^{2^{O(\log^2 d)}}$ with distortion $1 + 1/d^{\Omega(1)}$.

The above results can be generalized to the problem of embedding l_p^d for any $p \in [1, 2]$ into $l_1^{d'}$ (or in fact, into any l_q, $q \in [1, p)$). In particular, Johnson and Schechtman [JS82] showed that one can achieve distortion $1 + \epsilon$ with $d' \le c(\epsilon)d$; the function $c(\epsilon)$ "seems to be" $O(\log(1/\epsilon)/\epsilon^2)$. Their proof uses p-stable distributions in a quite elaborate fashion. Note that choosing all entries of the embedding matrix independently from a p-stable distribution does not work by itself, since the resulting randomized embedding does not have the "sharp concentration" property. This is due to the fact that p-stable distributions (for $p < 2$) are heavy-tailed.

Embeddings into l_∞. Another "very accommodating" host norm is the l_∞ norm. In particular, we have seen already that l_1^d can be isometrically embedded into $l_\infty^{2^d}$. In fact, this result can be generalized to any *polyhedral* norm, i.e., any norm whose unit ball is a polyhedron (this result seems to be folklore). Specifically, any polyhedral norm defined by a polyhedron with $2F$ faces can be isometrically embedded into l_∞^F, by using a linear embedding matrix with each row being a normal vector of one of the faces.

Since a unit ball in l_2^d can be approximated arbitrarily well by a polyhedron (using ϵ-nets), it follows that l_2^d can be embedded with distortion $1+\epsilon$ into $l_\infty^{d'}$ where $d' = O(1/\epsilon)^{d/2}$ (for even d). Through the relation between the embeddings and ϵ-nets, one can prove that this bound is tight (again, this seems to be a folklore result). On the other hand, it was shown by Dudley [Dud74] that *any* convex body B with diameter 1 can be approximated by a polyhedron P with $O(1/\epsilon)^{d/2}$ faces, such the Hausdorff distance between C and H is at most ϵ. This can be used to show that *any* d-dimensional norm can be embedded with distortion $1+\epsilon$ into $l_\infty^{d'}$ with $d' \le c(d)(1/\epsilon)^{d/2}$. This fact seems to be folklore, but its proof is somewhat nontrivial since it uses John's theorem. For more background on many of the above techniques (ϵ-nets, John's theorem) see [Mat].

If we allow randomized (and asymmetric) embeddings, it was shown by the author [Ind01] that it is possible to embed l_2^d into $l_\infty^{d'}$ with distortion $1+\epsilon$, contraction probability δ_1 and expansion probability δ_2, where $d' = (1/\epsilon + \log(1/\delta_1) + 1/\delta_2)^{O(1/\epsilon)}$. To compare this result with the earlier (deterministic) embeddings, observe that even if we set δ_1 to be inversely exponential in d, the dimension d' is still polynomial in d. On the other hand, d' depends polynomially on $1/\delta_2$. Making this dependence logarithmic is not possible, since it would imply (deterministic) embedding of l_2^d into $l_\infty^{d^{O(1)}}$ with constant distortion.

One implication of the above result is a randomized (asymmetric) embedding of a *product* of k Euclidean spaces (with the distance measured as maximum over the individual l_2 distances) into l_∞. This is obtained by setting $1/\delta_2 = O(k)$ and applying the above asymmetric embedding to all k Euclidean spaces in parallel.

Embedding into Hamming space. Consider a subset of l_1^d (denoted by M_1^d) in which all points have coordinates in the integer set $\{0 \dots M\}$. It is easy to see that the space M_1^d can be embedded into a Hamming metric with dimension Md: just replace each coordinate x_i by its unary representation. This was first observed in [LLR94].

3.4 Applications of "internorm" embeddings

We start from the applications related to embeddings into l_1 norm. From the previous section, it follows that l_2 can be embedded into l_1, and l_1 can in turn be embedded into Hamming space. Modulo the increase in the dimension (which, although quite severe in general, can be usually reduced to reasonable proportions), this means that it is sufficient to solve a given problem in the Hamming space, instead of Euclidean space. This turns out to be quite useful in a variety of situations. For example, it was shown in [IM98] that one can design a dynamic c-approximate data structure for the nearest neighbor problem in d-dimensional Hamming space, with $dn^{1/c}$-time per query or update. The above argument implies that the algorithm can be extended to l_1 and l_2 space. Similarly, the aforementioned nearest neighbor data structure of [KOR98] was originally designed for the Hamming space, and extended to l_1 and l_2 spaces via embeddings.

The l_∞ norm is in many respects even "nicer" than the l_1 norm. However, the exponential blow-up in the dimension makes the (deterministic) l_∞ embeddings applicable only to situations where the original dimension d is constant or small. One application which satisfies this constraint is $(1+\epsilon)$-approximate computation of the diameter of n points in l_2^d, which by the above results can be performed in $O(1/\epsilon)^{d/2}n$ time. This is due to the fact that in the l_∞ space the diameter can be computed in linear time, as seen in the introduction. A similar result can be also obtained for Maximum Spanning Tree, which is also fairly easy to compute for the l_∞ space.

The randomized asymmetric embedding allows to avoid the exponential blow-up and can be used even in the context of high-dimensional problems. In particular, by embedding a product of Euclidean spaces into l_∞ and then using an $O(1)$-approximate nearest neighbor for l_∞ [Ind98b], one can obtain a $O(1)$-approximate nearest neighbor algorithm for product of l_2's with $(d+\log n)^{O(1)}$ query time and $n^{O(\log d)}$ space.

4 Special metrics

In the last section of this survey we focus on embedding of "special" metrics into norms. In particular, we focus on *Hausdorff* metric (used in computer vision) and *Levenstein*, or edit distance, metric (used in text processing and computational biology). Embedding these metrics into normed spaces allows us to use algorithms (e.g., for clustering, nearest neighbor etc) developed for normed spaces in order to solve problems in the more complicated metrics. This is facilitated by the fact that most of the embeddings described in this section are oblivious.

Hausdorff metric. The Hausdorff metric H_M over a metric space $M = (X, D)$ is defined as a pair (X_M, D_M), where X_M is the set of all subsets of X, and $D_M(A, B)$ (for any $A, B \subset X$) is defined as

$D_M(A,B) = \max(\vec{D}_M(A,B), \vec{D}_M(B,A))$, where

$$\vec{D}_M(A,B) = \max_{a\in A}\min_{b\in B} D(a,b)$$

The usefulness of Hausdorff metric comes from the fact that it allows to define a distance between *sets* of points (e.g., geometric shapes). It becomes even more useful when it is augmented to be invariant with respect to some transformations of the sets A and B (e.g., translations). However, even computing individual distances $D_M(A,B)$ is a non-trivial task, especially when geometric transformations are allowed.

What do we know about embedding Hausdorff metric into normed spaces ? If the underlying metric $M = (X,D)$ is finite, the H_M is finite as well and therefore can be embedded into $l_\infty^{d'}$ with low (or no) distortion. Unfortunately, using Frechet or Matoušek embeddings from section 2.1 would result in d' exponential in $|X|$. However, it was shown in [FCI99] that d' can be reduced to $|X|$ while maintaining the no-distortion property (they used a variation of Frechet's proof). The intuition for this fact is that the Hausdorff metric is defined using the "max" operator, and therefore its structure is very similar to (although somewhat more complex than) the structure of the l_∞ norm.

The dimension of the host norm can be further reduced if we focus on embedding particular Hausdorff metrics. In particular, let H_M^s be the Hausdorff metric over all s-subsets of M. Farach-Colton and Indyk [FCI99] showed that if $M = l_2^d$, then H_M^s can be embedded into $l_\infty^{d'}$ with distortion $1+\epsilon$, where $d' = O(s^2(1/\epsilon)^{O(d)})$ (the $O(\cdot)$ constant depends on the diameter of the embedded pointset). This embedding also holds for the Hausdorff metrics invariant under translation, and is oblivious. For general (finite) metrics $M = (X,D)$ they show that H_M^s can be embedded into $l_\infty^{s^{O(1)}|X|^\alpha}$ for any $\alpha > 0$ with constant distortion (again, the dimension is somewhat dependent on the diameter of the embedded sets).

Levenstein metric. The Levenstein metric is defined over strings over certain alphabet Σ. Given two strings $s,t \in \Sigma^*$, the distance $D_L(s,t)$ is defined as minimum number of insertions, deletions or substitutions needed to transform s into t. Computing $D_L(s,t)$ is a nontrivial task in itself - the only algorithm for performing this task (even approximately) has running time $|s|\cdot|t|$.

Unfortunately, nothing is known so far about embeddability of D_L into normed spaces. However, if we extend the Levenstein metric to allow operations (insertions, deletions and substitutions) of *blocks* of characters, instead of just single characters[9], it was shown in [CPSV00, MS00] that the resulting metric D'_L can be embedded into l_1^d with distortion roughly $O(\log l)$, where l is the length of the embedded strings. It should be noted that this does *not* imply similar embeddability of D_L. However, the D'_L metric is probably as well motivated as D_L as far as the computational biology applications are concerned, and has additional applications to string compression.

An additional feature of the above results is that the embedding can be performed in almost linear time. This yields a very fast approximation algorithm for computing $D'_L(s,t)$ for individual s,t strings. This is of interest, since computing the exact value of $D'_L(s,t)$ is NP-hard.

Other special metrics. Another interesting metric which can be embedded into l_1 is the *transposition distance* metric over the set of permutations. Formally, the distance $D_T(\pi_1,\pi_2)$ is defined as the minimum number of moves of contiguous subsequences to arbitrary positions needed to transform π_1 into π_2. It was shown in [CMS01] that D_T can be embedded with constant distortion into l_1 (in fact, even into Hamming metric). They also show similar results for other permutation metrics, including reversal distance and permutation edit distance.

5 Applied applications

So far in this survey we have been focused on applications of embeddings techniques to problems of interest to theoretical computer science. In this section we depart from this theme and describe various applications of low-distortion embeddings in several applied areas. To keep this section in harmony with the rest of this survey, we will list the applications "by theorems", rather than by application areas.

Bourgain's lemma. The idea of embedding metric spaces into normed spaces in order to discover the structure of the metric has been around for quite a while (e.g., see [KW84]). However, Bourgain's lemma provides a new method for performing such an embedding, with provable distortion guarantees. Along this line of reasoning, Linial et al [LLTY97] used Bourgain's lemma to discover properties of the distance metric between protein sequences. They observed that many interesting biological properties of proteins can be (re)-discovered by analyzing the embedding of the metric into l_2; see the paper for more information.

Bartal's theorem. The theorem of Bartal [Bar96]

[9] See [CPSV00] for formal definition.

provides an efficient method of constructing a hierarchical decomposition of a graph which ensures that only few edges are cut (on the average). Thus, it can be naturally used as a tool for decomposing a communication network into smaller entities. In particular, variants of Bartal's algorithm have been used in [Sha98, AS98, JJJ$^+$00], for topology aggregation in undirected and directed networks, as well as efficient placement of traffic tracers in a network.

Johnson-Lindenstrauss lemma. The randomized dimensionality reduction technique has been used in a fairly large number of applied scenarios. To the knowledge of the author, it has been first used in [RK89] as a heuristic for speeding-up the computation in high-dimensional spaces. Later, the same idea has been experimentally evaluated in [Kas98]. For his data set, Kaski showed that a random dimensionality reduction resulted in reducing the dimension from about 5,000 to about 200, while incurring negligible clustering error. It is interesting to note that the authors of the above two papers were not aware of the work of Johnson and Lindenstrauss, and rediscovered the usefulness of random embeddings by themselves.

More recently, Dasgupta [Das00] evaluated random dimensionality reduction as a tool for speeding up the Expectation Maximization clustering algorithm. He showed that reducing the dimensionality of the input space not only does not decrease the probability of obtaining a good clustering, but in certain cases it can even increase that probability ! This is due to the fact that random projection transforms a non-spherical Gaussian distribution (which is difficult to handle using EM algorithm) into an almost spherical one.

The FFT-based algorithm for fast computation of sketches of substrings of a sequence has been introduced and evaluated in [IKM00]. It has been subsequently used for fast discovery of periodic structure of massive time series.

Recently, several papers used low-storage algorithms for maintaining approximate values of the l_p norms of vectors in various applied settings. In particular, Alon et al [AGMS99] evaluated the l_2-norm maintenance algorithm of [AMS96], for the purpose of maintaining the size of a self-join of massive relations. They showed that it provides estimations of higher quality than sampling-based methods. Their algorithm has been further used in several other projects, e.g., in [HGI00].

Gilbert et al [GKMS01] introduced and evaluated a low-storage algorithm for maintaining wavelet coefficients. Their algorithm (using the l_2 norm maintenance algorithm as a subroutine) was shown to perform almost as well as the exact off-line algorithm, while reducing the required storage from 50 MB to 5 KB for the case of AT&T call data. Many other research projects in this area are under development.

6 Conclusions and Open Problems

In this survey we showed that low-distortion embeddings provide a powerful and versatile toolkit for solving algorithmic problems. Their fundamental nature makes them applicable in a variety of diverse settings, while their relation to rich mathematical fields (e.g., functional analysis) ensures availability of tools for their construction.

The most important (even though somewhat naive) open (-ended) question about algorithmic applications of embeddings seems to be: what else can we solve using embeddings ? The answer to this question grows longer and longer every year, as new ways of using embeddings or new application domains are discovered. The author hopes that this survey will provide enough material to help the reader discover novel applications of embeddings him/herself.

In addition, there are several specific questions which seem to be crucial for deeper understanding of low-distortion embeddings, as well as their algorithmic applications. These questions include the following:

1. Can planar graph metrics be embedded into l_1 with constant distortion ? This seems to be "morally" true, since it is known that all *infinite vertex transitive* planar graphs are (essentially) either trees, meshes or lines, and each of these three classes of graphs are easily embeddable into l_1 with constant distortion.

   In addition to being a very important structural question by itself, a positive answer to it would imply existence of $O(1)$-approximate algorithms for the sparsest cut and related problems over planar graphs (assuming the embedding can be computed in polynomial time).

2. Is it possible to embed any finite metric into probabilistic trees with distortion $O(\log n)$? A positive answer to this question would improve approximation factors of many algorithms using probabilistic embeddings, and in many cases (e.g., for the buy-at-bulk problem) would give a tight approximation bound.

3. Is there a close analog of JL lemma for other norms, especially l_1 ? This would give a powerful technique for designing approximation algorithms for problems in the l_1 norms as in section 3.2 (although some of such problems can be solved even right now, using dimensionality reduction in Hamming space [OR00]). In addition, constant-distortion embedding of n-point subset of l_1^n into $l_1^{\log n}$ would imply a $O(\sqrt{\log n})$-approximation algorithm for finding best embedding of a finite metric M into l_1. This is due to the fact that if the optimal distortion of such an embedding is C, then (1) by the conjectured lemma we can assume $O(C)$-distortion embedding of M into $l_1^{\log n}$, (2) the latter also induces an $O(\sqrt{\log n}C)$-distortion embedding of M into l_2, which (3) can be found in polynomial time using semi-definite programming and (4) can be converted to embedding into l_1 via the results of section 3.3. This approach was suggested by Ashish Goel (personal communication).

4. Is it possible to embed edit distance metric into l_1 with low distortion ? A positive answer to this question would lead to approximate algorithms to several indexing problems under edit metric. In addition, if the embedding itself was efficiently computable (subquadratic in the sequence length), it would imply subquadratic approximation algorithm for computing the edit distance between two sequences. The latter is one of the biggest unsolved problems in the field of combinatorial pattern matching.

Acknowledgments

The author would like to thank a number of people for their comments on the preliminary version of this survey. In particular, he would like to thank Mihai Badoiu, Yair Bartal, Lars Engebretsen, Venkat Guruswami, Sariel Har-Peled and Jiri Matoušek.

Final comments

It is safe to assume that this survey contains (many) errors and omissions. The current version of this survey will be maintained at `theory.lcs.mit.edu/~indyk/tut.ps`.

References

[AA97] B. Awerbuch and Y. Azar. Buy-at-bulk network design. *Proceedings of the Symposium on Theory of Computing*, 1997.

[Ach01] D. Achlioptas. Database-friendly random projections. *Proceedings of the Symposium on Principles of Database Systems*, pages 274–281, 2001.

[AGMS99] N. Alon, P. Gibbons, Y. Matias, and M. Szegedy. Tracking join and self-join sizes in limited storage. *Proceedings of the ACM Symposium on Principles of Database Systems*, 1999.

[AK01] S. Arora and R. Kannan. Learning a mixture of gaussians. *Proceedings of the Symposium on Theory of Computing*, 2001.

[AKPW91] N. Alon, R. Karp, D. Peleg, and D. B. West. Graph-theoretic game and its applications to the k-server problem. *SIAM Journal on Computing*, 1991.

[AMS96] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Proceedings of the Symposium on Theory of Computing*, pages 20–29, 1996.

[AS98] B. Awerbuch and Y. Shavitt. Topology aggregation for directed graphs. *Proceedings of IEEE ISCC*, pages 47–52, 1998.

[AV99] R. I. Arriaga and S. Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Proceedings of the Symposium on Foundations of Computer Science*, 1999.

[Bar96] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. *Proceedings of the Symposium on Foundations of Computer Science*, 1996.

[Bar98] Y. Bartal. Approximating arbitrary metrics by tree metrics. *Proceedings of the Symposium on Theory of Computing*, pages 161–168, 1998.

[BBBT97] Y. Bartal, A. Blum, C. Burch, and A. Tomkins. A polylog(n)-competitive algorithm for metrical task systems. *Proceedings of the Symposium on Theory of Computing*, 1997.

[BCR01] Y. Bartal, M. Charikar, and D. Raz. Approximating min-sum k-clustering in metric spaces. *Proceedings of the Symposium on Theory of Computing*, 2001.

[Ber97] B. Berger. The fourth moment method. *SIAM Journal on Computing*, 26, 1997.

[BKRV98] A. Blum, G. Konjevod, R. Ravi, and S. Vempala. Semi-definite relaxations for minimum bandwidth and other vertex-ordering problems. *Proceedings of the Symposium on Theory of Computing*, 1998.

[Bou85] J. Bourgain. On lipschitz embedding of finite metric spaces into hilbert space. *Isreal Journal of Mathematics*, 52:46–52, 1985.

[Bou86] J. Bourgain. The metrical interpretation of superreflexivity in banach spaces. *Isreal Journal of Mathematics*, 56:222–230, 1986.

[CCG+98] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. Plotkin. Approximating a finite metric by a small number of tree metrics. *Proceedings of the Symposium on Foundations of Computer Science*, 1998.

[CKNZ01] C. Chekuri, S. Khanna, J. Naor, and L. Zossin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 109–118, 2001.

[CMS76] J. M. Chambers, C. L. Mallows, and B. W. Stuck. A method for simulating stable random variables. *J. Amer. Statist. Assoc.*, 71:340–344, 1976.

[CMS01] G. Cormode, M. Muthukrishnan, and C. Sahinalp. Permutation editing and matching via embeddings. *Proceedings of International Colloquium on Automata, Languages and Programming (ICALP)*, 2001.

[CPSV00] G. Cormode, M. Paterson, C. Sahinalp, and U. Vishkin. Communication complexity of document exchange. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2000.

[CR96] S. Chaudhuri and R. Radhakrishnan. Deterministic restrictions in circuit complexity. *Proceedings of the Symposium on Theory of Computing*, pages 30–36, 1996.

[Das99] S. Dasgupta. Learning mixtures of gaussians. *Proceedings of the Symposium on Foundations of Computer Science*, pages 634–644, 1999.

[Das00] S. Dasgupta. Experiments with random projection. *Uncertainty in Artificial Intelligence*, 2000.

[DG99] S. Dasgupta and A. Gupta. An elementary proof of the johnson-lindenstrauss lemma. *ICSI technical report TR-99-006, Berkeley, CA*, 1999.

[Dud74] R. M. Dudley. Metric entropy of some classes of sets with differentiable boundaries. *J. Approx. Theory*, 10(3):227–236, 1974.

[Dvo59] A. Dvoretzky. A theorem on convex bodies and applications to banach spaces. *Proceedings of the National Academy of Sciences USA*, 45:223–226, 1959.

[EIO01] L. Engebretsen, P. Indyk, and R. O'Donnell. Deterministic dimensionality reduction with applications. *Manuscript*, 2001.

[Erd64] P. Erdös. Extremal problems in graph theory. *Theory of Graphs and its Applications (Proc. Symp. Smolenice, 1963)*, pages 29–36, 1964.

[FCI99] M. Farach-Colton and P. Indyk. Approximate nearest neighbor algorithms for hausdorff metrics via embeddings. *Proceedings of the Symposium on Foundations of Computer Science*, 1999.

[Fei00] U. Feige. Approximating the bandwidth via volume respecting embeddings. *Journal of Computer and System Sciences*, 60(3):510–539, 2000.

[Fel91] W. Feller. *An Introduction to Probability Theory and its Applications.* John Wiley & Sons, NY, 1991.

[FKSV99] J. Feigenbaum, S. Kannan, M. Strauss, and M. Viswanathan. An approximate l1-difference algorithm for massive data streams. *Proceedings of the Symposium on Foundations of Computer Science*, 1999.

[FLM77] T. Figiel, J. Lindenstrauss, and V. D. Milman. The dimension of almost spherical sections of convex bodies. *Acta Mathematica*, 139:53–94, 1977.

[FM88] P. Frankl and H. Maehara. The johnson-lindenstrauss lemma and the sphericity of some graphs. *Journal of Combinatorial Theory B*, 44:355–362, 1988.

[FM00] A. Fiat and M. Mendel. Better algorithms for unfair metrical task systems and applications. *Proceedings of the Symposium on Theory of Computing*, pages 725–734, 2000.

[FP99] D. Finocchiaro and M. Pellegrini. On computing the diameter of a point set in high dimensional euclidean space. *Proceedings of the European Symposium on Algorithms*, 1999.

[FS00] J. Fong and M. Strauss. An approximate lp-difference algorithm for massive data streams. *Proceedings of the Symposium on Theoretical Computer Science*, 2000.

[GGK+01] A. Gilbert, S. Guha, Y. Kotidis, P. Indyk, M. Muthukrishnan, and M. Strauss. Efficient maintenance of histograms. *Manuscript*, 2001.

[GKMS01] A. Gilbert, Y. Kotidis, M. Muthukrishnan, and M. Strauss. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. *Proceedings of the International Conference on Very Large Databases (VLDB)*, 2001.

[GKR98] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[GNRS99] A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair. Cuts, trees and l1 embeddings. *Proceedings of the Symposium on Foundations of Computer Science*, 1999.

[GPPR01] C. Gavoille, D. Peleg, S. Perennes, and R. Raz. Distance labeling in graphs. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 210–219, 2001.

[Gup01] A. Gupta. Steiner nodes in trees don't (really) help. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2001.

[HGI00] T. Haveliwala, A. Gionis, and P. Indyk. Scalable techniques for clustering the web. *WebDB Workshop*, 2000.

[HP01] S. Har-Peled. A replacement for voronoi diagrams of near linear size. *Proceedings of the Symposium on Foundations of Computer Science*, 2001.

[HPI00] S. Har-Peled and P. Indyk. When crossings count - approximating the minimum spanning tree. *Proceedings of the Annual ACM Symposium on Computational Geometry*, 2000.

[HTW01] M. Herlihy, S. Tirthapura, and R. Wattenhofer. Competitive concurrent distributed queueing. *Proceedings of the 20th ACM Symposium on Principles of Distributed Computing*, 2001.

[IKM00] P. Indyk, N. Koudas, and S. Muthukrishnan. Identifying representative trends in massive time series datasets using sketches. *Proceedings of the 26th International Conference on Very Large Databases (VLDB)*, 2000.

[IM98] P. Indyk and R. Motwani. Approximate nearest neighbor: towards removing the curse of dimensionality. *Proceedings of the Symposium on Theory of Computing*, 1998.

[Ind98a] P. Indyk. Faster algorithms for string matching problems: matching the convolution bound. *Proceedings of the Symposium on Foundations of Computer Science*, 1998.

[Ind98b] P. Indyk. On approximate nearest neighbors in l_∞ norm. *Journal of Computer and System Sciences, to appear. Preliminary version appeared in Proceedings of the Symposium on Foundations of Computer Science*, 1998.

[Ind00a] P. Indyk. Dimensionality reduction techniques for proximity problems. *Proceedings of the Ninth ACM-SIAM Symposium on Discrete Algorithms*, 2000.

[Ind00b] P. Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. *Proceedings of the Symposium on Foundations of Computer Science*, 2000.

[Ind01] P. Indyk. Better algorithms for high-dimensional proximity problems via asymmetric embeddings. *Manuscript*, 2001.

[JJJ+00] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang. On the placement of internet instrumentation. 2000.

[JKM+98] H. V Jagadish, N. Koudas, S. Muthukrishnan, V. Poosala, K. C. Sevcik, , and T. Suel. Optimal histograms with quality guarantees. *Proceedings of the International Conference on Very Large Databases (VLDB)*, pages 275–286, 1998.

[JS82] W.B. Johnson and G. Schechtman. Embedding l_p^m into l_1^n. *Acta Mathematica*, 149:71–85, 1982.

[Kas98] S. Kaski. Dimensionality reduction by random mapping: Fast similarity computation for clustering. *Proceedings of IJCNN, International Joint Conference on Neural Networks*, 1998.

[KOR98] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *Proceedings of the Thirtieth ACM Symposium on Theory of Computing*, pages 614–623, 1998.

[KPR99] M. R. Korupolu, C. G. Plaxton, and Rajmohan Rajaraman. Placement algorithms for hierarchical cooperative caching. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 586–595, 1999.

[KR00] G. Konjevod and R. Ravi. An approximation algorithm for the covering steiner problem. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 338–344, 2000.

[KT99] J. M. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *Proceedings of the Symposium on Foundations of Computer Science*, 1999.

[KW84] J. B. Kruskal, , and M. Wish. *Multidimensional Scaling.* Beverly Hills and London: Sage Publications, 1984.

[LLR94] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Proceedings of 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 577–591, 1994.

[LLTY97] M. Linial, N. Linial, N. Tishby, and G. Yona. Global self organization of all known protein sequences reveals inherent biological signatures. *J. Mol. Biol.*, 268:539 – 556, 1997.

[LM] J. Lindenstrauss and V.D. Milman. Local theory of normed spaces and convexity. *Handbook of convex geometry*, B:1149–1220.

[LR99] F. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46:787–832, 1999.

[LV99] L. Lovasz and K. Vesztergombi. Geometric representations of graphs. *Paul Erdös and his Mathematics, Proc. Conf. Budapest*, 1999.

[Mat] J. Matoušek. Lectures on discrete geometry. *Springer, in press.*

[Mat96] J. Matoušek. On the distortion required for embedding finite metric spaces into normed spaces. *Israel Journal of Mathematics*, 93:333–344, 1996.

[Mat99] J. Matoušek. On embedding trees into uniformly convex banach spaces. *Isreal Journal of Mathematics*, 114:221–237, 1999.

[MS00] S. Muthukrishnan and C. Sahinalp. Approximate nearest neighbors and sequence comparison with block operations. *Proceedings of the Symposium on Theory of Computing*, 2000.

[MVW98] Y. Matias, J. Scott Vitter, and M. Wang. Wavelet-based histograms for selectivity estimation. *Proceedings of the Symposium on Management of Data (SIGMOD)*, 1998.

[OR00] R. Ostrovsky and Y. Rabani. Polynomial time approximation schemes for geometric k-clustering. *Proceedings of the Symposium on Foundations of Computer Science*, 2000.

[Pel99] D. Peleg. Proximity-preserving labeling schemes for graphs. *Proceedings of the Symposium on Mathematical Foundations of Computer Science*, pages 30–41, 1999.

[Rao87] S. Rao. Finding near optimal separators in planar graphs. *Proceedings of the Symposium on Foundations of Computer Science*, pages 225–237, 1987.

[Rao99] S. Rao. Small distortion and volume preserving embeddings for planar and euclidean metrics. *Proceedings of the Symposium on Computational Geometry*, pages 300–306, 1999.

[RK89] H. Ritter and T. Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61:241–254, 1989.

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

[Sch81] G. Schechtman. Random embeddings of euclidean spaces in sequence spaces. *Israel J. Math.*, 40:187–192, 1981.

[Sha98] Y. Shavitt. Topology aggregation for networks with hierarchical structure: A practical approach. *Proceedings of 36th Annual Allerton Conference on Communication, Control, and Computing*, 1998.

[Tre01] L. Trevisan. When hamming meets euclid: the approximability of geometric tsp and mst. *SIAM J. on Computing*, 30:475–485, 2001.

[TZ01] M. Thorup and U. Zwick. Approximate distance oracles. *Proceedings of the Symposium on Theory of Computing*, 2001.

[Vem98] S. Vempala. Random projection: A new approach to vlsi layout. *Proceedings of the Symposium on Foundations of Computer Science*, 1998.

[WLB+99] B.Y. Wu, G. Lancia, V. Bafna, K. Chao, R. Ravi, and C.Y. Tang. A polynomial time approximation scheme for minimum routing cost spanning trees. *SIAM J. Computing*, 29:761–778, 1999.