Better Algorithms for High-dimensional Proximity Problems via Asymmetric Embeddings

Piotr Indyk
MIT

Abstract

In this paper we give several results based on randomized embeddings of l_2 into l_∞ (or "l_∞-like") spaces. Our first result is a $(1+\epsilon)$-distortion *asymmetric* embedding of n points in l_2 into l_∞ with polylog(n) dimension, for any $1+\epsilon$. This gives the first known $O(1)$-approximate nearest neighbor algorithm with fast query time and almost polynomial space for a product of Euclidean norms, a common generalization of both l_2 and l_∞ norms. Our embedding also clarifies the relative complexity of approximate nearest neighbor in l_2 and l_∞ spaces.

Our second result in a $(1+\epsilon)$-approximate algorithm for the diameter of n points in l_2^d, running in time $\tilde{O}(dn^{1+1/(1+\epsilon)^2})$; the algorithm is fully dynamic. This improves several previous algorithms for this problem (see Table 1 for more information).

1 Introduction

Embeddings between normed spaces are known to be very useful tools for designing geometric algorithms. A classic example is an $O(2^d n)$-time algorithm for computing diameter of n points in l_1^d [GBT84]: since the problem seems difficult in the original space, we can embed[1] l_1^d isometrically into $l_\infty^{2^d}$ and then compute the diameter of the points in l_∞, which can be done in linear time. More recently, embeddings have been used to design geometric algorithms which avoid exponential dependence of their running time on dimension (e.g., see [KOR98, IM98, FCI99, Ind00a, MS00, Ind00b, AV99, Das99]). With the exception of [FCI99] (which relied on a structural similarity between Hausdorff metric and l_∞), the above papers used dimensionality reduction techniques by performing randomized linear mappings from a norm of interest into l_1, l_2 or Hamming space with low dimension.

In this paper we give several results based on randomized embeddings of l_2 into l_∞ (or "l_∞-like") spaces. Similar approach has been used before, e.g., in [AMS92], where a $(1+\epsilon)$-approximation algorithm for diameter in l_2^d with complexity $O(1/\epsilon)^{d/2}n$ has been obtained by mapping l_2^d into $l_\infty^{O(1/\epsilon)^{d/2}}$. However, it seemed unlikely that this technique would result in an algorithm with running time subexponential in d, due to the fact that for any linear embedding of l_2^d into $l_\infty^{d'}$ with constant distortion, d' must be exponential[2]. in d (e.g., see [LM], equation 3.1.5). In this paper we avoid this problem by providing an *asymmetric* embedding, i.e., randomized embedding f such that:

- the probability that f *contracts* a fixed pair of points is very small (inversely *exponential* in the l_∞ dimension)
- the probability that f *expands* the distance between a fixed pair of points is small but only inversely *polynomial* in the l_∞ dimension

Asymmetric embeddings have been introduced in [Ind00b] to show a dimensionality reduction lemma for l_1. In this paper we show that l_2^d can be embedded into $l_\infty^{d'}$ with $d' = (N_2 + \log N_1 + 1/\epsilon)^{O(1/\epsilon)}$, such that the probability of contraction is $1/N_1$ and the probability of expansion by at least a factor of $(1+\epsilon)$ is at most $1/N_2$. In fact, the mapping can be easily extended to give an embedding of a *product* of l_2 norms into l_∞. The *product* $M = M_1 \times \ldots \times M_k$ of metric spaces $M_i = (X_i, D_i)$, is a metric space[3] over $X_1 \times \ldots X_k$, with distance function D such that $D((p_1, \ldots, p_k), (q_1, \ldots, q_k)) = \max_{i=1\ldots k} D_i(p_i, q_i)$. Product metrics are very useful in scenarios when one needs to combine information of different types (e.g., color and texture of an image), see [Fag96, Fag98, AGGM98]. In this paper, we show an embedding of a product of k Euclidean spaces into into

[1] The (linear) mapping is defined by a $d \times 2^d$ matrix obtained by concatenating (vertically) all elements of $\{-1,1\}^d$.

[2] Note that this lower bound also excludes constant-distortion randomized linear embeddings of n points in l_2^d into $l_\infty^{n^{o(1)}}$, since by standard arguments such an embedding could be extended to the whole l_2^d by mapping $n = 2^{O(d)}$ points uniformly distributed on a unit sphere.

[3] Observe that if D_i's are norms, then their product is also a norm.

l_∞ with dimension $d' = O(N_2 + k + \log N_1 + 1/\epsilon)^{O(1/\epsilon)}$. Since asymmetric embeddings with constant N_2 are sufficient for nearest neighbor search, we can use an approximate nearest neighbor algorithm for l_∞ (see Section 3 for details) to obtain an approximate nearest neighbor algorithm for product of Euclidean spaces. In particular, we obtain an algorithm which for any $\rho > 0$ answers any query in time $O(d' \log n)$, uses $O(n^{1+\rho})$ space and achieves approximation factor $C = O(\log_{1+\rho} \log d')$. Note that by setting $\rho = \log d'$ we can achieve constant approximation factor.

We mention that by using techniques mentioned in [Ind98] (i.e., by embedding each metric into l_∞^d with $d = n^{1/c}$ and distortion $O(c)$, and then applying the aforementioned algorithm for l_∞^{dk}), one could obtain an algorithm with either query time polynomial in n, or approximation error $O(\log n)$. In the latter case the algorithm would require superpolynomial space, we skip the details. In contrast, by using our techniques we can achieve query time polynomial in $(d + \log n)$ while keeping error and space moderate.

We also note that the algorithm recently presented in [Ind02] can be also used for product of Euclidean spaces. However, it achieves worse parameters (in particular, $O(\log \log n)$ approximation factor).

We also observe that the above result implies that approximate nearest nearest problem is formally harder (with respect to certain parameters) in l_∞ than in l_2 (see Section 3 for discussion). Finally, it can also be used to show that any n-point metric in l_2 can be embedded into $l_\infty^{n^\alpha}$ with distortion $\sqrt{2/\alpha + 1}$. This improves (for the special case of l_2 norm) the general result of [Mat96] who showed that *any* finite metric over n points can be embedded into $l_\infty^{n^\alpha}$ with distortion $2/\alpha - 1$ (for α's such that the distortion is an integer).

Our second result is a dynamic data structure for c-approximate furthest neighbor in l_2^d, with (roughly) $O(dn^{1/c^2})$-time per operation (insertion, deletion or query). By the result of [Epp95] it also implies a dynamic data structure for c-approximate diameter, with similar (although amortized) time per operation. This yields the best known algorithms for both *computing* or *maintaining* the diameter of a given set of n points, improving the earlier results shown in Table 1. Moreover, the algorithm is much simpler than the ones in [BOR99b, Ind00a, GIV01], which were obtained by composing several reductions between problems and metric spaces.

Finally, our algorithm can be modified to approximate the diameter of the set of points when the data points come one by one in a *stream* using sublinear space. Specifically, it maintains a c-approximate diameter using $O(dn^{1/(c^2-1)})$ space and $O(dn^{1/(c^2-1)})$ time per each new point, for $c > \sqrt{2}$. The only previously known result of this type was a simple 2-approximation using $O(d)$ space.

Ref	Appr.	Total time	Time per operation
[EK89]	$\sqrt{3}$	$O(dn)$	N/A
[FP99]	$1+\epsilon$	$\tilde{O}(n^2)$	$O(n/\epsilon^2)$
[BOR99b]	$1+\epsilon$	$\tilde{O}(n^{2-O(\epsilon^2)})$	$\tilde{O}(n^{1-O(\epsilon^2)} \log \Delta)$
[Ind00a]	$1+\epsilon$	$\tilde{O}(n^{1+1/(1+\epsilon/6)})$	$\tilde{O}(n^{1/(1+\epsilon/6)} \log \Delta)$
[GIV01]	$1+\epsilon$	$\tilde{O}(n^{1+1/(1+\epsilon)})$	$\tilde{O}(n^{\frac{1}{2}+\frac{1}{2(1+\epsilon)}} \log \Delta)$
[GIV01]	$\sqrt{2}$	$\tilde{O}(dn)$	$\tilde{O}(n^{1/2})$
here	$1+\epsilon$	$\tilde{O}(n^{1+1/(1+\epsilon)^2})$	$\tilde{O}(n^{1/(1+\epsilon)^2})$

Table 1: Results on diameter in high dimensions. Δ denotes the diameter/closest pair ratio. We use $\tilde{O}(\cdot)$ to hide the polynomial dependence on $\log n, 1/\epsilon$. For brevity, the running times exclude the additive factors bounding the time needed to read the input and reduce its dimension, which is $\tilde{O}(dn)$ for the total time, and $\tilde{O}(d)$ for the query/update time. Note that by dimensionality reduction techniques we can reduce d to $O(\log n/\epsilon^2)$.

Our techniques. Both of our results make use of multiple random mappings from $\Re^d$ to $\Re$ defined as $a(x) = a \cdot x$, where a is a d-dimensional vector whose coordinates are i.i.d. random variables with normal $N(0,1)$ distribution. The choice of the normal distribution is motivated by the fact that, by spherical symmetry, the random variable $a(x)$ has again normal distribution with variance $|x|_2^2$, which facilitates the analysis. The embedding of l_2^d into $l_\infty^{d'}$ is performed by mapping x into $(a_1 \cdot x, \ldots, a_{d'} \cdot x) = Ax$. The analysis of the mapping proceeds by analyzing the distribution of $Z = |Ax|_\infty$. It turns out that the left tail of Z is much "sharper" than the right tail, which results in asymmetric dependence of d' on the probabilities of expansion and contraction.

The approximate furthest neighbor algorithm also proceeds by mapping all points p into Ap. The basic idea of the algorithms is as follows. If the right tail of $|Ap|_\infty$ was sharper than it is in reality, then we would know that for a fixed query q and for all pairs p, p' satisfying $|p' - q|_2 < |p - q|_2/c$, the inequality $|Ap' - Aq|_\infty < |Ap - Aq|_\infty$ would hold with high probability. In this case it would be sufficient to maintain a furthest neighbor data structure for points in $l_\infty^{d'}$, which would take only $O(d' \log n)$ time per operation, for $d' = O(\log^{O(1/\epsilon)} n)$. Unfortunately, the right tail of $|Ap|_\infty$ is not sharp enough for this purpose, and the above approach only results in a data structure with running time which is sublinear only for c large enough. However, it turns out that one can slightly modify the algorithm operating on points Ap so that it computes several candidates for a furthest neighbor (instead of just one), and obtain an algorithm for furthest neighbor in l_2^d with roughly $O(dn^{1/c^2})$ time per operation.

It is interesting to note that all previous sub-quadratic $(1+\epsilon)$-approximate algorithms for furthest neighbor and diameter used techniques developed for approximate *nearest* neighbor (the results in [Ind00a, GIV01] were in fact obtained by black-box reductions to an approximate nearest neighbor data structure). On the other hand, our algorithm does not use any such techniques. In fact, our algorithm is faster than the c-approximate nearest neighbor data structure of [IM98], which requires roughly $O(dn^{1/c})$ time per operation. It is also interesting to note that the technique we use, namely searching for the furthest neighbor by performing multiple projections of points on a line, provably does not work for the nearest neighbor (e.g., see [Kle97]). This indicates that the algorithm indeed exploits the special properties of the furthest neighbor problem.

2 Preliminaries

Notation. We use $c = 1+\epsilon > 0$ to denote the approximation factors of algorithms or distortions of embeddings. A function f is an embedding of a For any metric $M = (X, D)$, a mapping $f : X \to l_p^d$ is an embedding with distortion c, if for any $p, q \in X$, $D(p,q) \le |p-q|_p \le cD(p,q)$.

Probability. We use $\Phi(t)$ to denote $\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{t} e^{-x^2/2}dx$, i.e., the distribution function of a random variable with normal distribution $N(0,1)$.

The following claim can be easily verified using elementary analysis (e.g., see [Fel91], Chapter VII.1, Lemma 2):

CLAIM 1. *The following inequalities hold:*

- *for any $t \ge 0$ we have $1 - \Phi(t) \le e^{-t^2/2}/t$*
- *there exists a constant $B > 0$ such that for any $t \ge 0$ we have $1 - \Phi(t) \ge Be^{-t^2/2}/\max(1,t)$*

3 Embedding of the product of Euclidean norms into l_∞

In this section we show how to embed a product L of k Euclidean norms $l_2^{d_1}, \ldots, l_2^{d_k}$ into l_∞^{dk} with distortion $1+\epsilon$, where $d = (k + \ln N_1 + 1/\epsilon)^{O(1/\epsilon)}$, for a parameter N_1. The embedding is asymmetric: the probability of contraction is very small (i.e., $1/N_1$), while the probability of expansion by a factor larger than $1+\epsilon$ is at most $1/2$. The probability of expansion can be reduced to $1/N_2$ (while increasing d to $(k + N_2 + \ln N_1 + 1/\epsilon)^{O(1/\epsilon)}$) in a similar way.

The embedding f is defined as follows . Let A_i for $i = 1 \ldots k$ be a $d \times d_i$ matrix whose entries are i.i.d. random variables with normal distribution $N(0,1)$, where d is as above. Then $f : L \to l_\infty^{dk}$ is a linear mapping such that for $p = (p_1, \ldots, p_k) \in L$, $f(p) = (A_1 p_1, \ldots, A_k p_k)$.

THEOREM 3.1. *There exists T such that for any $p \in L$*

1. $\Pr[|f(p)|_\infty \le T|p|_L] \le 1/N_1$
2. $\Pr[|f(p)|_\infty > (1+\epsilon)T|p|_L] < 1/2$

In order to prove the theorem, we need the following auxiliary lemma about the tails of the maximum of independent variables with normal distribution.

LEMMA 3.1. *Let $Y_1 \ldots Y_d$ be i.i.d. normal variables and let $Z = \max(|Y_1|, \ldots, |Y_d|)$. For any positive real numbers N_1, N_2*

- $\Pr[Z < T_1] \le 1/N_1$ *for* $T_1 = \sqrt{2\ln(\frac{2d}{\ln N_1 \cdot \sqrt{2\ln 2d} \cdot 1/B})}$, *and*
- $\Pr[Z > T_2] \le 1/N_2$ *for* $T_2 = \sqrt{2\ln(2dN_2)}$.

Proof. We start from T_1. For any $t \ge 0$

$$\begin{aligned}\Pr[Z < t] &= (1 - 2(1-\Phi(t)))^d \\ &\le (1 - 2Be^{-t^2/2}/t)^d \\ &\le \exp(-d2Be^{-t^2/2}/t)\end{aligned}$$

It is easy to see that in order to make sure that the latter expression is not greater than $1/N_1$, it is sufficient to assume that $t \le T_1$.

Now we proceed with T_2. For any $t \ge 0$ we have

$$\begin{aligned}\Pr[Z > t] &\le d2(1-\Phi(t)) \\ &\le 2de^{-t^2/2}\end{aligned}$$

Therefore, in order to make sure that the latter expression is not greater than $1/N_2$, it is sufficient to assume that $t \ge \sqrt{2\ln(2dN_2)} = T_2$.

Proof. **(of the Theorem)**. Since f is linear, without loss of generality we can assume $|p|_L = 1$. Set $T = T_1$ as in Lemma 3.1.

We start from the first bound. Since $|p|_L = 1$, it follows that $|p_i|_2 = 1$ for some i. By spherical symmetry of normal distribution, it follows that each coordinate $(A_i p_i)_j$ has normal distribution $N(0,1)$. Therefore, from Lemma 3.1 it follows that $\Pr_{A_i}[|A_i p_i|_\infty < T] \leq 1/N_1$. Thus, the first bound follows.

Now consider the second bound. From Lemma 3.1, if we set $T_2 = \sqrt{2(\ln 2d + \ln 2k)}$, then $\Pr_{A_i}[|A_i p_i|_\infty > T_2] \leq \frac{1}{2k}$; this implies that $\Pr_A[|f(p)|_\infty > T_2] \leq \frac{k}{2k} = 1/2$. Therefore, it is sufficient to make sure that $(1+\epsilon)T \geq T_2$, or $T_2/T \leq 1+\epsilon$. Let $\delta = \Theta(\epsilon)$ be such that

$$\frac{\sqrt{1+\delta}}{\sqrt{1-2\delta}} \leq 1+\epsilon$$

and set $d = \max(2k, \ln N_1, 1/B^2, 1/\delta^a)^{1/\delta}/2$, for a large enough constant a. Then

$$T_2/T \leq \frac{\sqrt{2(\ln(2d)+\ln(2k))}}{\sqrt{2(\ln(2d) - \ln\ln N_1 - \frac{1}{2}\ln(2\ln 2d) - \ln(1/B))}}$$

By dividing the numerator and denominator by $\sqrt{2\ln(2d)}$ and simplifying the expression, we obtain

$$T_2/T \leq \frac{\sqrt{1+\delta}}{\sqrt{1-2\delta}} \leq 1+\epsilon$$

Application to near(est) neighbor. Assume we are given n points $p_1 \ldots p_n$ in a product L of k Euclidean spaces; our goal is to build a data structure which for any query q, if there is a database point p such that $|q-p|_L \leq 1$, it returns a point p' such that $|q-p|_L \leq C$ (by the result of [IM98] such a data structure is sufficient to provide a data structure for C-approximate *nearest* neighbor problem with similar bounds). We will obtain such a data structure by mapping the data set into $l_\infty^{d'}$ for $d' = dk$; for the latter norm there is an algorithm [Ind98] which for any $\rho > 0$:

- answers any query in time $O(d' \log n)$
- uses $O(n^{1+\rho})$ space
- achieves approximation factor $C = O(\log_{1+\rho} \log d')$

Consider the mapping $F(x) = \frac{1}{T(1+\epsilon)} f(x)$, where $f(\cdot)$ is a random mapping as in Theorem 3.1, with $N_1 = 4n$. It follows that with probability $1/2$ we have $|F(q) - F(p)|_\infty = |F(q-p)|_\infty \leq 1$. On the other hand, for any point p' such $|q-p'|_L > C(1+\epsilon)$, we have that $\Pr[|F(q) - F(p')|_\infty \leq C] \leq 1/N_1 = \frac{1}{4n}$. Therefore, the probability that there is at least one such p' is at most $1/4$. Thus, with probability at least $1 - 1/2 - 1/4 = 1/4$ there exists $F(p)$ with distance 1 from $F(q)$, and all points $F(p)$ within distance C from $F(q)$ are such that $|p-q|_L \leq C(1+\epsilon)$. Thus, with constant probability the algorithm returns a $C(1+\epsilon)$-approximate near neighbor (with respect to the product metric). The probability can be amplified by using standard repetition techniques.

Other applications. By using Lemma 3.1 in the way similar to the above, we can also show that for any set P of n points in l_2^d there exists a linear mapping $f : P \to l_\infty^{d'}$, $d' = \tilde{O}(n^\alpha)$, $\alpha > 0$, such that for any pair of points $p, p' \in P$ we have

$$|f(p) - f(p')|_\infty \leq |p-p'|_2 \leq c|f(p) - f(p')|_\infty$$

where $c = \sqrt{2/\alpha + 1}$. This improves (for the special case of l_2 norm) the general result of [Mat96] who showed that *any* finite metric over n points can be embedded into $l_\infty^{n^\alpha}$ with distortion $2/\alpha - 1$ (for α's such that the distortion is an integer).

Another implication of Theorem 3.1 is the formalization of the apparent fact that approximate nearest neighbor problem is harder in l_∞ than in l_2. This fact seemed plausible, since e.g., there is an algorithm for $(1+\epsilon)$-approximate nearest neighbor in l_2^d with $(\log n + d + 1/\epsilon)^{O(1)}$ query time and $n^{1/\epsilon^{O(1)}}$ space [KOR98, IM98], while by the result of [Ind98] any such algorithm for $\epsilon < 2$ implies a similar solution to the *partial match* (also called *subset query* problem) [BOR99a], which is conjectured to require exponential storage to achieve a fast query time. However, Theorem 3.1 offers first known formalization of this intuition.

4 Furthest neighbor

In this section we present an algorithm which maintains a set of n points in l_2^d (under insertions and deletions) and which answers $(c+\delta)$-approximate furthest neighbor queries, for any $\delta > 0$. The cost of any operation (i.e., insertion, deletion or query) is

$$O(dn^{1/c^2} \log^{1+(1-1/c^2)/2} n \log_{1+\delta} d \log\log_{1+\delta} d).$$

The probability that a given query is answered correctly is constant; it can be amplified to an arbitrarily large value by standard repetition techniques.

We first describe an algorithm for the decision version of c-approximate furthest neighbor (we denote this problem by c-DFN). In this case our goal is to maintain a data structure which given a query point $q \in l_2^d$ and $r > 0$ does the following: if there exists a point $p \in P$ such that $|p-q|_2 \geq r$, then the data structure reports a point $p' \in P$ such that $|p'-q|_2 \geq r/c$; it can return an arbitrary point otherwise. Note that the value of r is *not* fixed a priori, but comes together with the query. We will later show how to reduce c-approximate furthest neighbor to c-DFN via binary search.

The data structure for c-DFN is as follows. Firstly, we create $l = O(n^{1/c^2} \log^{(1-1/c)/2} n)$ d-dimensional vectors $a_1, \ldots a_l$, where each element of each a_i is chosen independently at random from normal distribution $N(0,1)$. Moreover, we maintain l sorted lists $L_1, \ldots, L_l$, where each list L_i contains the values of dot products $a_i \cdot p_1, \ldots a_i \cdot p_n$. Such lists can be maintained in time $O((d + \log n)l)$ per insertion/deletion.

In order to answer a query (q, r), we retrieve the pairs (p, i) (using lists L_i) such that $|a_i \cdot p - a_i \cdot q| \geq T = rt/c$, where t is a solution of an equation

$$e^{t^2/2}/t = 2n.$$

(observe that $t = O(\sqrt{\log n})$). If there are more than $2l + 1$ such pairs, we only retrieve first $2l + 1$ of them. Then, for all retrieved pairs (p, i), we check if $|p - q|_2 \geq r/c$; if we find such a point, we report it. Note that the total query time is bounded by $O(l(d + \log n))$.

The correctness of the algorithm follows from the following two claims.

CLAIM 2. *Let p' be a point such that $|p' - q|_2 < r/c$. Then*

$$\Pr_a[|a \cdot q - a \cdot p'| \geq T] \leq 1/n.$$

CLAIM 3. *Let p be a point such that $|p - q|_2 \geq r$. Then*

$$\Pr_a[|a \cdot q - a \cdot p| \geq T] \geq 1/l.$$

Assume for now that both claims are true. Then, from Claim 2, the expected total number of pairs (p', i) such that $|p' - q|_2 < r/c$ and $|a_i \cdot q - a_i \cdot p'| \geq T$ is at most l. Therefore, with probability at least $1/2$ there are no more than $2l$ such pairs. Moreover, from Claim 3, the probability that for *some* i we have $|a_i \cdot q - a_i \cdot p| \geq T$ is at least $1 - (1 - 1/l)^l \geq 1 - 1/e$. Thus, with a constant probability, the data structure returns *a* point which is *not* within distance r/c from q, and therefore is a correct answer.

Now we prove the two claims.

Proof. **(of Claim 2)**. Consider

$$\begin{aligned} P &= \Pr_a[|a \cdot q - a \cdot p'| \geq T] \\ &= \Pr_a\left[\frac{|a \cdot (q - p')|}{|q - p'|_2} \geq T/|q - p_2'|\right] \\ &\leq \Pr_a\left[\frac{|a \cdot (q - p')|}{|q - p'|_2} \geq \frac{T}{r/c}\right] \end{aligned}$$

Observe that $Y = \frac{|a \cdot (q - p')|}{|q - p'|_2}$ has a normal $N(0,1)$ distribution, and therefore

$$\begin{aligned} P &\leq \Pr[|Y| \geq \frac{T}{r/c}] \\ &= \Pr[|Y| \geq t] \\ &\leq 2e^{-t^2/2}/t \\ &\leq 1/n \end{aligned}$$

Proof. **(of Claim 3)**. Consider

$$\begin{aligned} P' &= \Pr_a[|a \cdot q - a \cdot p| \geq T] \\ &= \Pr_a\left[\frac{|a \cdot (q - p)|}{|q - p|_2} \geq T/|q - p|_2\right] \\ &\geq \Pr_a\left[\frac{|a \cdot (q - p)|}{|q - p|_2} \geq \frac{T}{r}\right] \end{aligned}$$

By the same argument as in Claim 2 we have

$$\begin{aligned} P' &\geq 2Be^{-(t/c)^2/2}/(t/c) \\ &= 2Bc(e^{-t^2/2}/t)^{1/c^2}/t^{1-1/c^2} \\ &= \frac{2Bc}{n^{1/c^2} t^{1-1/c^2}} \end{aligned}$$

Let l be equal to the inverse of the latter value, note that $l = O(n^{1/c^2} \log^{(1-1/c^2)/2} n)$. The claim follows.

Furthest neighbor. Now we show how to find a $(c + \delta)$-approximate furthest by reducing it to c-DFN. To this end, we define a *box width* $bw(P)$ of a set of points P to be the value of

$$\max_{i=1\ldots d} |\max_{p \in P}(p)_i - \min_{p \in P}(p)_i|.$$

It is easy to see that we can maintain the value of $bw(P)$ within the time $O(d \log n)$ per insertion/deletion. Moreover, for any query q, if p is the furthest neighbor of q in P, then $bw(P)/2 \leq |q - p|_2$. On the other hand, if the distance from q to the center of the box is at least $\sqrt{d}/\epsilon \cdot bw(P)$, then any point from P is an $(1 + \epsilon)$-approximate furthest neighbor of q. Therefore, given $bw(P)$, we can determine a $(c + \delta)$-approximate furthest neighbor of q by performing $O(\log_{1+\delta}(d/\epsilon))$ calls to c-DFN, using binary search. In fact we need to perform each call using $O(\log \log_{1+\delta} d)$ different data structures, to amplify the probability that all of the binary search calls were answered correctly.

In this way we proved the following theorem.

THEOREM 4.1. *The $(c + \delta)$-approximate furthest neighbor in l_2^d can be maintained in time $O(dn^{1/c^2} \log^{1+(1-1/c^2)/2} n \log_{1+\delta} d \log \log_{1+\delta} d)$ per operation.*

Stream model. If we want to maintain the diameter in the scenario when the input points are coming in a stream (and use sublinear space), we need to modify the algorithm a bit. We still maintain $l = O(n^{1/(c^2-1)} \log n)$ random vectors $a_1 \dots a_l$ used as before. However, for each a_i, we maintain only the two points p with the largest and the smallest value of $a_i \cdot p$, and these are the only points we use in order to compute the furthest neighbor when a new point arrives. Clearly, the space and time bounds are $O(dl)$. The analysis is similar as before and relies on the fact that if we set $T \approx \sqrt{\frac{2c^2 \log n}{c^2-1}}$, the probability that for $n+1$ unit vectors $x_1 \dots x_n, x$ we have (for a fixed i)

$$|a_i \cdot x_j| \leq T \text{ for all } j \text{ and } |a_i \cdot cx| > T$$

is at least $1/n^{1/(c^2-1)}$.

5 Additional observations and open problems

The results in this paper give rise to several important open problems. Firstly, is it possible to obtain similar asymmetric embeddings into l_∞ from norms different than l_2 ? A somewhat restricted positive answer (although sufficient for the nearest neighbor applications) holds for l_p for $p \in [1, 2)$; this result can be obtained by embedding l_p into l_1 [JS82], then into Hamming space and then into the square of l_2. A positive answer for other norms would imply existence of fast approximate nearest neighbor algorithms for those norms, as well as for their products.

The second open problems concerns the complexity of $(1+\epsilon)$-approximate nearest neighbor in l_2. It can be seen (cf. [GIV01]) that if the data set is *random* (e.g., each points is chosen independently and uniformly at random from the unit sphere), $d = \Omega(\log n)$, and the queries lie on the unit sphere, then finding a $(1+\epsilon)$-approximate nearest neighbor of q can be reduced to finding a $(1 + h(\epsilon)\epsilon)$-approximate furthest neighbor of the "antipode" of q. The function $h(\epsilon)$ tends to 1 as ϵ tends to 0. This means that for this restrictive setting the algorithm given in this paper implies an improved algorithm for the approximate nearest neighbor. Is it possible to obtain such an algorithm (with $O(dn^{1/(1+\epsilon)^2})$ time per operation) for general data sets ?

References

[AGGM98] P.K. Agarwal, A. Gionis, L. Guibas, and R. Motwani. Rank-based similarity searching. *Manuscript*, 1998.

[AMS92] P.K. Agarwal, J. Matoušek, and S. Suri. Farthest neighbors, maximum spanning trees and related problems in higher dimensions. *Computational Geometry: Theory and Applications*, 1(4):189–201, 1992.

[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.

[BOR99a] A. Borodin, R. Ostrovsky, and Y. Rabani. Lower bounds for high dimensional nearest neighbor search and related problems. *Proceedings of the Symposium on Theory of Computing*, 1999.

[BOR99b] A. Borodin, R. Ostrovsky, and Y. Rabani. Subquadratic approximation algorithms for clustering problems in high dimensional spaces. *Proceedings of the Symposium on Theory of Computing*, 1999.

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

[EK89] O. Egecioglu and B. Kalantari. Approximating the diameter of a set of points in the euclidean space. *Information Processing Letters*, 32:205, 1989.

[Epp95] D. Eppstein. Dynamic euclidean minimum spanning trees and extrema of binary functions. *Disc. Comp. Geom.*, 13:111–122, 1995.

[Fag96] R. Fagin. Combining fuzzy information from multiple systems. *Proceedings of the ACM Symposium on Principles of Database Systems*, pages 216–227, 1996.

[Fag98] R. Fagin. Fuzzy queries in multimedia database systems (invited paper). *Proceedings of the ACM Symposium on Principles of Database Systems*, 1998.

[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.

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

[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.

[GBT84] H. N. Gabow, J. L. Bentley, and R. E. Tarjan. Scaling and related techniques for geometry problems. *Proceedings of the Symposium on Theory of Computing*, pages 135–143, 1984.

[GIV01] A. Goel, P. Indyk, and K. Varadarajan. Reductions among high-dimensional geometric problems. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2001.

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

[Ind98] 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.

[Ind02] P. Indyk. Approximate nearest neighbor algorithms for frechet metric via product metrics. *Symposium on Computational Geometry*, 2002.

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

[Kle97] J. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, 1997.

[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.

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

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

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