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 $l_\infty$ (or $l_\infty$-like) spaces. Our first result is a $(1 + \epsilon)$-distortion asymmetric embedding of $n$ points in $l_2$ into $l_\infty$ 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 $l_\infty$ norms. Our embedding also clarifies the relative complexity of approximate nearest neighbor in $l_2$ and $l_\infty$ spaces.

Our second result in a $(1 + \epsilon)$-approximate algorithm for the diameter of $n$ points in $l_2^d$, running in time $	ilde{O}(dn^{1+1/(1+\epsilon)})$; 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 $O(2^n)$-time algorithm for computing diameter of $n$ points in $l_2^d$ [GRT84]; since the problem seems difficult in the original space, we can embed\footnote{The [linear] mapping is defined by a $d \times 2^d$ matrix obtained by concatenating (vertically) all elements of $\{-1, 1\}^d$.} $l_2^d$ isometrically into $l_\infty^d$ and then compute the diameter of the points in $l_\infty$, 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, AY99, Das99]). With the exception of [FCI99] (which relied on a structural similarity between Hausdorff metric and $l_\infty$), 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 $l_\infty$ (or $l_\infty$-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\footnote{Note that this lower bound also excludes constant-distortion randomized linear embeddings of $n$ points in $l_2^d$ into $l_\infty^{O(1)}$, 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.}, 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 $l_\infty$ dimension)
- the probability that $f$ expands the distance between a fixed pair of points is small but only inversely polynomial in the $l_\infty$ 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 $l_\infty$. The product $M = M_1 \times \ldots \times M_k$ of metric spaces $M_i = (X_i, D_i)$, is a metric space\footnote{Observe that if $D_i$'s are norms, then their product is also a norm.} over $X_1 \times \ldots \times X_k$, 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, AGG98]. In this paper, we show an embedding of a product of $k$ Euclidean spaces into...
$l_\infty$ with dimension $d' = O(N_2 + k + \log N_1 + 1/e)^{O(1/\epsilon)}$. Since asymmetric embeddings with constant $N_2$ are sufficient for nearest neighbor search, we can use an approximate nearest neighbor algorithm for $l_\infty$ (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 $C = O(\log n^{1+\rho} \log d')$. Note that by setting $\rho = \log d'$ we can achieve constant approximation factor.

We mention that by using techniques mentioned in [Ind098] (i.e., by embedding each metric into $l_\infty$ with $d = n^{1/\epsilon}$ and distortion $O(\epsilon)$, and then applying the aforementioned algorithm for $l_\infty^{d/8}$), 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 neighbor problem is formally harder (with respect to certain parameters) in $l_\infty$ 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 $l_\infty$ 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 $l_\infty$ with distortion $2/\alpha - 1$ (for $\alpha$'s such that the distortion is an integer).

Our second result is a dynamic data structure for $c$-approximate furthest neighbor in $l_2$, 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.

Our techniques. Both of our results make use of multiple random mappings from $\mathbb{R}^d$ to $\mathbb{R}$ 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 $x_i^2$, which facilitates the analysis. The embedding of $l_2^d$ into $l_\infty$ 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 $A_p$. The basic idea of the algorithms is as follows. If the right tail of $|A_p|_\infty$ is much sharper than it is in reality, then we would know that for a fixed query $q$ and for all pairs $p, p'$ satisfying $|p' - q| < |q - p|/c$, the inequality $|A_{p'} - A_q| < |A_p - A_q|_\infty$ 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 $d' = O(\log n^{1/\epsilon})$. Unfortunately, the right tail of $|A_p|_\infty$ 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

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Approx.</th>
<th>Total time</th>
<th>Time per operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[EK99]</td>
<td>$\sqrt{3}$</td>
<td>$O(dn)$</td>
<td>$N/A$</td>
</tr>
<tr>
<td>[FP99]</td>
<td>$1 + \epsilon$</td>
<td>$\tilde{O}(n^2)$</td>
<td>$O(n^{1+\epsilon})$</td>
</tr>
<tr>
<td>[BOR99b]</td>
<td>$1 + \epsilon$</td>
<td>$\tilde{O}(n^2 \log \Delta)$</td>
<td>$O(n^{1+\epsilon} \log \Delta)$</td>
</tr>
<tr>
<td>[Ind00a]</td>
<td>$1 + \epsilon$</td>
<td>$\tilde{O}(n^{1+\epsilon}/(1+\epsilon))$</td>
<td>$O(n^{1+\epsilon}/(1+\epsilon)) \log \Delta)$</td>
</tr>
<tr>
<td>[GIV01]</td>
<td>$\sqrt{2}$</td>
<td>$\tilde{O}(dn)$</td>
<td>$O(n^{1/2})$</td>
</tr>
<tr>
<td>[GIV01]</td>
<td>$\sqrt{d}$</td>
<td>$\tilde{O}(n^{1+\epsilon}/1+c^2)$</td>
<td>$O(n^{1/(1+c^2)} \log \Delta)$</td>
</tr>
</tbody>
</table>

Table 1: Results on diameter in high dimensions. $\Delta$ 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 $O(dn)$ for the total time, and $O(d)$ for the query/update time. Note that by dimensionality reduction techniques we can reduce $d$ to $O(\log n/\epsilon^2)$.
enough. However, it turns out that one can slightly modify the algorithm operating on points $A_p$ so that it computes several candidates for a furthest neighbor (instead of just one), and obtain an algorithm for furthest neighbor in $l^d_2$ with roughly $O(d n^{1/l^2})$ 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 $O(d n^{1/c})$ 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 $f : X \to l^d$ is an embedding with distortion $c$, if for any $p, q \in X$, $D(p, q) \leq c D(p, q)$.

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 $t \geq 0$ we have $1 - \Phi(t) \leq e^{-t^2/2}/t$

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

3 Embedding of the product of Euclidean norms into $l_{\infty}$

In this section we show how to embed a product $L$ of $k$ Euclidean norms $l^d_1, \ldots, l^d_k$ into $l^d_2$ with distortion $1 + \epsilon$, where $d = (k + \ln N_1 + 1/\epsilon^{2/3})/\ln N_0$, 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 $(k + N_2 + \ln N_1 + 1/\epsilon^{2/3})/\ln N_0$), 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 $f : L \to l^d_2$ 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. $\Pr[\|f(p)\| \leq T \|p\|] \leq 1/N_1$

2. $\Pr[\|f(p)\| > (1 + \epsilon)\|T \|p\|] < 1/2$

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 $N_1, N_2$

- $\Pr[Z < T_1] \leq 1/N_1$ for $T_1 = \sqrt{2\ln(\frac{N_0}{2}\ln(2dN_0))}$

- $\Pr[Z > T_2] \leq 1/N_2$ for $T_2 = \sqrt{2\ln(2dN_2)}$

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

$\Pr[Z < t] = \left(1 - 2(1 - \Phi(t)) \right)^d \leq \left(1 - 2Be^{-t^2/2}/t \right)^d \leq \exp(-d2Be^{-t^2/2}/t)$

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 $t \leq T_1$.

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

$\Pr[Z > t] \leq d2(1 - \Phi(t)) \leq 2d e^{-t^2/2}$

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

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 $p_i \|z\|_2$ 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 \( \Pr_{A_i} [ |A_i p_i|_\infty < T ] \leq \frac{1}{N_1} \). Thus, the first bound follows.

Now consider the second bound. From Lemma 3.1, if we set \( T_2 = \sqrt{2\ln(2d + \ln(k))} \), then \( \Pr_{A_i} [ |A_i p_i|_\infty > T_2 ] \leq \frac{1}{N_1} \). This implies that \( \Pr_{A_i} [ |f(p_i)|_\infty > T_2 ] \leq \frac{1}{N_1} \). Therefore, it is sufficient to make sure that \( (1 + \epsilon)T > T_2 \), 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

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

**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 \( l_2^d \) 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 \( F(x) = \frac{1}{M(x)} f(x) \), 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 that \( |q - p'|_L > C(1 + \epsilon) \), 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 \( p' \) at most \( 1/4 \). Thus, with probability at least \( 1 - 1/2 - 1/4 = 1/4 \) there exists \( p' \) 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 \( f : P \rightarrow l_2^d, d = O(n^{1+\rho}) \), \( \rho > 0 \), 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 \( l_2^3 \) with distortion \( 2/\alpha - 1 \) (for \( \alpha \)'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 \( l_\infty \) 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 \( n^{1+o(1)} \) 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 \( (\epsilon + \delta) \)-approximate furthest neighbor queries, for any \( \delta > 0 \). The cost of any operation (i.e., insertion, deletion or query) is

\[
O(d n^{1/\epsilon^2} \log^{1+1/\epsilon^2} n \log_{1+\rho} d \log \log_{1+\rho} d).
\]

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 \( \epsilon \)-approximate furthest neighbor (we denote this problem by \( \epsilon \)-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 \( \epsilon \)-approximate furthest neighbor to \( \epsilon \)-DFN via binary search.
The data structure for c-DFN is as follows. Firstly, we create \( l = O(n^{1/2} \log^{1/2} n) \) \( d \)-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
\[
Pr[a \cdot q - a \cdot p' \geq T] \leq 1/n.
\]

**Claim 3.** Let \( p \) be a point such that \( |p - q|_2 \geq r \). Then
\[
Pr[a \cdot q - a \cdot p \geq T] \geq 1/l.
\]

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 \( 1/2l \) 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
\[
P = Pr_a [a \cdot q - a \cdot p' \geq T] = Pr_a \left[ \frac{|a \cdot (q - p')|}{|q - p'_2|} \geq \frac{T}{|q - p'_2|} \right] \leq Pr_a \left[ \frac{|a \cdot (q - p')|}{|q - p'_2|} \geq \frac{T}{r/c} \right].
\]

Observe that \( Y = \frac{a \cdot (q - p')}{|q - p'_2|} \) has a normal \( N(0, 1) \) distribution, and therefore
\[
P \leq Pr[Y \geq \frac{T}{r/c}]
\]
\[
= Pr[|Y| \geq \frac{T}{r/c}] \leq 2e^{-T^2/2l} \leq 1/n.
\]

**Proof. (of Claim 3).** Consider
\[
P' = Pr_a [a \cdot q - a \cdot p \geq T]
\]
\[
= Pr_a \left[ \frac{|a \cdot (q - p)|}{|q - p'_2|} \geq \frac{T}{|q - p'_2|} \right] \geq Pr_a \left[ \frac{|a \cdot (q - p)|}{|q - p'_2|} \geq \frac{T}{r/c} \right].
\]

By the same argument as in Claim 2 we have
\[
P' \geq 2Be^{-lT^2/2l} \geq 2Be^{-lT^2/2l} \geq 2Be^{-lT^2/2l}
\]
\[
= \frac{2Be^{-lT^2/2l}}{n^{1/2} \cdot r^2/c^2}
\]

Let \( l \) be equal to the inverse of the latter value, note that \( l = O(n^{1/2} \log^{1/2} n) \). 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
\[
\max_{i=1}^{d} \left| \max_{p \in P} (p)_i - \min_{p \in P} (p)_i \right|
\]
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 \( \sqrt{1/e} \cdot bw(P) \), 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 \( O(\log_3 3d/d(e)) \) calls to c-DFN, using binary search. In fact we need to perform each call using \( O(\log \log_3 3d) \) 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 \( \ell^2 \) can be maintained in time
\[
O(dn^{1/2} \log^{1/2} n) \log_3 3d \log \log_3 3d)
\]
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 \(a_1 \ldots a_l\) 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 
\(x_1 \ldots x_n, x\) 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 \(l_\infty\) 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 point 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 \((1 + h(\epsilon))\)-approximate furthest neighbor of the “antipode” of \(q\). The function \(h(\epsilon)\) tends to \(1\) as \(\epsilon\) 
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 \(O(dn^{1/(1+c)^2})\) 
time per operation) for general data sets?

References


[EK99] O. Egecioglu and B. Kalantari. Approximating the diameter of a set of points in the euclidean space. 


[FP99] D. Finocchi and M. Pellegrini. On computing the diameter of a point set in high dimensional euclidean space. 


[Ind98] P. Indyk. On approximate nearest neighbors in \(l_\infty\) norm. Journal of Computer and System Sciences, 
1998.

[Ind00a] P. Indyk. Dimensionality reduction techniques for proximity problems. Proceedings of the Ninth ACM-

[Ind00b] P. Indyk. Stable distributions, pseudorandom gen-
erators, embeddings and data stream computation. 


