MIT Open Access Articles

Beyond locality-sensitive hashing

Citation: Andoni, Alexandr. "Beyond Locality-Sensitive Hashing." SODA '14 Proceedings of the Twenty-fifth Annual ACM-SIAM Aymposium on Discrete Algorithms, 5-7 January, 2014, Philadelphia, Pennsylvania, Association for Computing Machinery, 2014, pp. 1018-1028.

As Published: http://dl.acm.org/citation.cfm?id=2634150&CFID=969998081&CFTOKEN=28708107

Publisher: Association for Computing Machinery

Persistent URL: http://hdl.handle.net/1721.1/114155

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Beyond Locality-Sensitive Hashing

Alexandr Andoni	Piotr Indyk	Huy L. Nguyễn	Ilya Razenshteyn
Microsoft Research	MIT	Princeton	MIT
andoni@microsoft.com	indyk@mit.edu	hlnguyen@princeton.edu	ilyaraz@mit.edu

Abstract

We present a new data structure for the c-approximate near neighbor problem (ANN) in the Euclidean space. For n points in $\mathbb{R}^d$, our algorithm achieves $O_c(n^\rho + d \log n)$ query time and $O_c(n^{1+\rho} + d \log n)$ space, where $\rho \leq 7/(8c^2) + O(1/c^3) + o_c(1)$. This is the first improvement over the result by Andoni and Indyk (FOCS 2006) and the first data structure that bypasses a locality-sensitive hashing lower bound proved by O'Donnell, Wu and Zhou (ICS 2011). By a standard reduction we obtain a data structure for the Hamming space and ℓ_1 norm with $\rho \leq 7/(8c) + O(1/c^{3/2}) + o_c(1)$, which is the first improvement over the result of Indyk and Motwani (STOC 1998).

1 Introduction

The near neighbor search problem is defined as follows: given a set P of n points in a d-dimensional space, build a data structure that, given a query point q, reports any point within a given distance r to the query (if one exists). The problem is of major importance in several areas, such as databases and data mining, information retrieval, computer vision, databases and signal processing.

Many efficient near(est) neighbor algorithms are known for the case when the dimension d is "low" (e.g., see [Mei93], building on [Cla88]). However, despite decades of effort, the current solutions suffer from either space or query time that are exponential in the dimension d. This phenomenon is often called "the curse of dimensionality". To overcome this state of affairs, several researchers proposed *approximation* algorithms for the problem. In the (c, r)-approximate near neighbor problem (ANN), the data structure is allowed to return any data point whose distance from the query is at most cr, for an approximation factor $c > 1$. Many approximation algorithms for the problem are known, offering tradeoffs between the approximation factor, the space and the query time. See [And09] for an up to date survey.

From the practical perspective, the space used by an algorithm should be as close to linear as possible. If the space bound is (say) sub-quadratic, and the approximation factor c is a constant, the best existing solutions are based on *locality-sensitive hashing* [IM98, HPIM12]. The idea of that approach is to hash the points in a way that the probability of collision is much higher for points which are close (with the distance r) to each other than for those which are far apart (with distance at least cr). Given such hash functions, one can retrieve near neighbors by hashing the query point and retrieving elements stored in buckets containing that point. If the probability of collision is at least p_1 for the close points and at most p_2 for the far points, the algorithm solves the (c, r)-ANN using $n^{1+\rho+o(1)}$ extra space and $dn^{\rho+o(1)}$ query time[1], where $\rho = \log(1/p_1)/\log(1/p_2)$ [HPIM12].

---

[1]Assuming that each hash function can be sampled and evaluated in $n^{o(1)}$ time, stored in $n^{o(1)}$ space, that distances can be computed in $O(d)$ time, and that $1/p_1 = n^{o(1)}$.

The value of the exponent ρ depends on the distance function and the locality-sensitive hash functions used. In particular, it is possible to achieve $\rho = 1/c$ for the ℓ_1 norm [IM98], and $\rho = 1/c^2 + o_c(1)$ for the ℓ_2 norm [AI06].

It is known that the above bounds for the value of ρ are *tight.* Specifically, we have that, for all values of c, $\rho \geq 1/c - o_c(1)$ for the ℓ_1 norm[2] [OWZ11]. A straightforward reduction implies that $\rho \geq 1/c^2 - o_c(1)$ for the ℓ_2 norm. Thus, the running time of the simple LSH-based algorithm, which is determined by ρ, cannot be improved.

Results. In this paper we show that, despite the aforementioned limitation, the space and query time bounds for ANN can be substantially improved. In particular, for the ℓ_2 norm, we give an algorithm with query time $O_c(n^\eta + d \log n)$ and space $O_c(n^{1+\eta} + d \log n)$, where $\eta = \eta(c) \leq 7/(8c^2) + O(1/c^3) + o_c(1)$ that gives an improvement for large enough c. This also implies an algorithm with the exponent $\eta \leq 7/(8c) + O(1/c^{3/2}) + o_c(1)$ for the ℓ_1 norm, by a classic reduction from ℓ_1 to ℓ_2-squared [LLR95]. These results constitute the first improvement to the complexity of the problem since the works of [IM98] and [AI06].

Techniques. Perhaps surprisingly, our results are obtained by using essentially the same LSH functions families as described in [AI06] or [IM98]. However, the properties of those hash functions that we exploit, as well as the overall algorithm, are different. On a high-level, our algorithms are obtained by combining the following two observations:

1. After a slight modification, the existing LSH functions can yield better values of the exponent ρ if the search radius r is comparable to the diameter[3] of the point-set. This is achieved by augmenting those functions with a "center point" around which the hashing is performed. See Section 1.1 for an intuition why this approach works, in the (somewhat simpler) context of the Hamming distance.

2. We can ensure that the diameter of the point-set is small by applying standard LSH functions to the original point-set P, and building a separate data structure for each bucket.

This approach leads to a two-level hashing algorithm. The *outer hash table* partitions the data sets into buckets of bounded diameter. Then, for each bucket, we build the *inner hash table*, which uses (after some pruning) the center of the minimum enclosing ball of the points in the bucket as a center point. Note that the resulting two-level hash functions cannot be "unwrapped" to yield a standard LSH family, as each bucket uses slightly different LSH functions, parametrized by different center points. That is, the two-level hashing is done in a *data-aware* mannerwhile the standard LSH functions are chosen from a distribution independent from the data. This enables us to overcome the lower bound of [OWZ11].

Many or most of the practical applications of LSH involve designing data-aware hash functions. Unfortunately, not many rigorous results in this area are known. The challenge of understanding and exploiting the relative strengths of data-oblivious versus data-aware methods has been recognized as a major open question in the area (e.g., see [fmd13], page 77). Our results can be viewed as a step towards that goal.

Related work. In this paper we assume worst case input. If the input is generated at random, it is known that one can achieve better running times. Specifically, assume that all points are generated uniformly at random from $\{0, 1\}^d$, and the query point is "planted" at distance $d/(2c)$

---

[2] Assuming $1/p_1 = n^{o(1)}$.

[3] In the analysis we use a notion that is weaker than the diameter. However, we ignore this detail for now for the sake of clarity.

from its near neighbor. In this setting, the work of [CR93, GPY94, KWZ95, PRR95] gives an exponent of roughly $\frac{1}{\ln 4 \cdot c} \approx \frac{1}{1.39c}$.

Even better results are known for the problem of finding the *closest pair* of points in a dataset. In particular, the algorithm of [Dub10] for the random closest pair has an exponent of $1 + \frac{1}{2c-1}$.[4] More recently, [Val12] showed how to obtain an algorithm with a runtime exponent < 1.79 for any approximation $c = 1 + \varepsilon$ in the random case. Moreover, [Val12] also gives an algorithm for the worst-case closest pair problem with a runtime exponent of $2 - \Omega(\sqrt{\varepsilon})$ for $c = 1 + \varepsilon$ approximation.

There are also two related lines of lower bounds for ANN. First, the work of [MNP07] showed that LSH for Hamming space must have $\rho \geq 1/(2c) - O(1/c^2) - o_c(1)$, and [OWZ11] improved the lower bound to $\rho \geq 1/c - o_c(1)$. Second, [PTW08, PTW10] have given cell-probe lower bounds for ℓ_1 and ℓ_2, roughly showing that any randomized ANN algorithm for the ℓ_1 norm must either use space $n^{1+\Omega(1/(tc))}$ or more than t cell-probes. We note that the LSH lower bound of $\rho \geq 1/(2c)$ from [MNP07] might more naturally predict lower bounds for ANN because it induces a "hard distribution" that corresponds to the aforementioned "random case" . In contrast, if one tries to generalize the LSH lower bound of [OWZ11] into a near neighbor hard distribution, one obtains a dataset with special structure, which one can exploit (and our algorithm will indeed exploit such structure). In fact, the LSH lower bound of [MNP07] has been used (at least implicitly) in the data structure lower bounds from [PTW08, PTW10].

1.1 Intuition behind the improvement

We give a brief intuition on why near neighbor instances with bounded diameter are amenable to more efficient LSH functions. For simplicity we consider the Hamming distance as opposed to the Euclidean distance.

Assume that all input points, as well as the query point, are within the Hamming distance of s from each other. By shifting one of the data points to the origin, we can assume that all points have at most s non-zeros (i.e., ones). Consider any data point p and the query point q. To make calculations easier, we assume that both p and q have exactly s ones.

The "standard" LSH functions for the Hamming distance project the points on one of the coordinates selected uniformly at random. For two points p and q this results in a collision probability of $1 - \|p-q\|_1/d$, which is $1 - r/d$ and $1 - cr/d$ for points within the distance of r and cr, respectively. The probability gap of $1 - x$ vs. $1 - cx$ leads to the exponent ρ equal to $1/c$ [IM98]. To improve on this, we can instead use the min-wise hash functions of [Bro98]. For those functions, the probability of collision between two points p and q is equal to $\frac{|p \cap q|}{|p \cup q|}$, where $\cup$ and $\cap$ denote the union and intersection of two Boolean vectors, respectively. Since we assumed that $\|p\|_1 = \|q\|_1 = s$, we have

$$\frac{|p \cap q|}{|p \cup q|} = \frac{\|p\|_1 + \|q\|_1 - \|p-q\|_1}{\|p\|_1 + \|q\|_1 + \|p-q\|_1} = \frac{2s - \|p-q\|_1}{2s + \|p-q\|_1} = \frac{1 - \|p-q\|_1/(2s)}{1 + \|p-q\|_1/(2s)}$$

As a result, the collision probability gap for distances r and cr becomes $\frac{1-x}{1+x}$ vs. $\frac{1-cx}{1+cx}$. This leads to ρ that is lower than $1/c$.

---

[4]Note that a near neighbor search algorithm with query time n^ρ and space/preprocessing time of $n^{1+\rho}$ naturally leads to a solution for the closest pair problem with the runtime of $n^{1+\rho}$.

2   Preliminaries

In the text we denote the ℓ_2 norm by $\|\cdot\|$. When we use $O(\cdot)$, $o(\cdot)$, $\Omega(\cdot)$ or $\omega(\cdot)$ we explicitly write all the parameters that the corresponding constant factors depend on as subscripts.

Definition 1. *The (c,r)-approximate near neighbor problem (ANN) with failure probability f is to construct a data structure over a set of points P in metric space (X,D) supporting the following query: given any fixed query point $q \in X$, if there exists $p \in P$ with $D(p,q) \le r$, then report some $p' \in P$ such that $D(p',q) \le cr$, with probability at least $1 - f$.*

Remark: note that we allow preprocessing to be randomized as well, and we measure the probability of success over the random coins tossed during *both* preprocessing and query phases.

Definition 2 ([HPIM12])**.** *For a metric space (X,D) we call a family of hash functions $\mathcal{H}$ on X (r_1, r_2, p_1, p_2)-sensitive, if for every $x, y \in X$ we have*

- *if $D(x,y) \le r_1$, then $\Pr_{h \sim \mathcal{H}}[h(x) = h(y)] \ge p_1$;*

- *if $D(x,y) \ge r_2$, then $\Pr_{h \sim \mathcal{H}}[h(x) = h(y)] \le p_2$.*

Remark: for $\mathcal{H}$ to be useful we should have $r_1 < r_2$ and $p_1 > p_2$.

Definition 3. *If $\mathcal{H}$ is a family of hash functions on a metric space X, then for any $k \in \mathbb{N}$ we can define a family of hash function $\mathcal{H}^{\otimes k}$ as follows: to sample a function from $\mathcal{H}^{\otimes k}$ we sample k functions $h_1, h_2, \ldots, h_k$ from $\mathcal{H}$ independently and map $x \in X$ to $(h_1(x), h_2(x), \ldots, h_k(x))$.*

Lemma 4. *If $\mathcal{H}$ is (r_1, r_2, p_1, p_2)-sensitive, then $\mathcal{H}^{\otimes k}$ is (r_1, r_2, p_1^k, p_2^k)-sensitive.*

Theorem 5 ([HPIM12])**.** *Suppose there is a (r, cr, p_1, p_2)-sensitive family $\mathcal{H}$ for (X,D), where $p_1, p_2 \in (0,1)$ and let $\rho = \ln(1/p_1)/\ln(1/p_2)$. Then there exists a data structure for (c,r)-ANN over a set $P \subseteq X$ of at most n points, such that:*

- *the query procedure requires at most $O(n^\rho/p_1)$ distance computations and at most $O(n^\rho/p_1 \cdot \lceil \log_{1/p_2} n \rceil)$ evaluations of the hash functions from $\mathcal{H}$ or other operations;*

- *the data structure uses at most $O(n^{1+\rho}/p_1)$ words of space, in addition to the space needed to store the set P.*

The failure probability of the data structure can be made to be arbitrarily small constant.

Remark: this theorem says that in order to construct a good data structure for the (c,r)-ANN it is sufficient to have a (r, cr, p_1, p_2)-sensitive family $\mathcal{H}$ with small $\rho = \ln(1/p_1)/\ln(1/p_2)$ and not too small p_1.

We use the LSH family crafted in [AI06]. The properties of this family that we need are summarized in the following theorem.

Theorem 6 ([AI06])**.** *For every sufficiently large d and n there exists a family $\mathcal{H}$ of hash functions for ℓ_2^d such that*

- *a function from $\mathcal{H}$ can be sampled in time, stored in space, and computed in time $t^{O(t)} \cdot \log n + O(dt)$, where $t = \log^{2/3} n$;*

- *the collision probability of $\mathcal{H}$ for two points $u, v \in \mathbb{R}^d$ depends only on the distance between u and v; let us denote it by $p(\|u - v\|)$;*

- *one has the following inequalities for $p(\cdot)$:*

$$
\begin{aligned}
p(1) &\geq L, &\text{where}&& L &= \frac{A}{2\sqrt{t}} \cdot \frac{1}{(1+\varepsilon+8\varepsilon^2)^{t/2}} \\
\forall c > 1 \quad p(c) &\leq U(c), &\text{where}&& U(c) &= \frac{2}{(1+c^2\varepsilon)^{t/2}},
\end{aligned}
$$

where A is an absolute positive constant that is less than 1, and $\varepsilon = \Theta(t^{-1/2}) = \Theta(\log^{-1/3} n)$.

Combining Theorem 5 and Theorem 6 one has the following corollary.

Corollary 7. *There exists a data structure for (c, r)-ANN for ℓ_2^d with preprocessing time and space $O_c(n^{1+1/c^2+o_c(1)} + nd)$ and query time $O_c(dn^{1/c^2+o_c(1)})$.*

Proof. By rescaling one can assume w.l.o.g. that $r = 1$. Then, it is left to check that $L = n^{-o_c(1)}$ and $\ln(1/L)/\ln(1/U(c)) \leq 1/c^2 + o_c(1)$. These computations can be found in [AI06]. $\qquad\square$

We use the following standard estimate on tails of Gaussians (see, e.g., [KMS98]).

Lemma 8 ([KMS98]). *For every $t > 0$*

$$
\frac{1}{\sqrt{2\pi}} \cdot \left(\frac{1}{t} - \frac{1}{t^3} \right) \cdot e^{-t^2/2} \leq \Pr_{X \sim N(0,1)}[X \geq t] \leq \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{t} \cdot e^{-t^2/2}.
$$

We use Johnson-Lindenstrauss dimension reduction procedure.

Theorem 9 ([JL84], [DG03]). *For every $d \in \mathbb{N}$ and $\varepsilon, \delta > 0$ there exists a distribution over linear maps $A \colon \mathbb{R}^d \to \mathbb{R}^{O(\log(1/\delta)/\varepsilon^2)}$ such that for every $x \in \mathbb{R}^d$ one has $\Pr_A[\|Ax\| \in (1 \pm \varepsilon)\|x\|] \geq 1 - \delta$. Moreover, such a map can be sampled in time $O(d\log(1/\delta)/\varepsilon^2)$.*

Finally, let us state Jung's theorem.

Theorem 10 (see, e.g., Exercise 1.3.5 in [Mat02]). *Every subset of $\mathbb{R}^d$ with diameter Δ can be enclosed in a ball of radius $\Delta/\sqrt{2}$.*

3 Gaussian LSH

In this section we present and analyze a $(1, c, p_1, p_2)$-sensitive family of hash functions for the ℓ_2 norm that gives an improvement upon [AI06] for the case, when all the points and queries lie on a spherical shell of radius $O(c)$ and width $O(1)$. The construction is similar to an SDP rounding scheme from [KMS98].

First, we present an "idealized" family. In the following theorem we do not care about time and space complexity and assume that all points lie on a *sphere* of radius $O(c)$.

Theorem 11. *For a sufficiently large c, every $\nu \geq 1/2$ and $1/2 \leq \eta \leq \nu$ there exists an LSH family for $\eta c \cdot S^{d-1} = \left\{ x \in \mathbb{R}^d \mid \|x\| = \eta c \right\}$ with the ℓ_2 norm that is $(1, c, p_1, p_2)$-sensitive, where*

- $p_1 = \exp(-o_{c,\nu}(d))$;

- *one has*

$$\rho = \frac{\ln(1/p_1)}{\ln(1/p_2)} = \left(1 - \frac{1}{4\eta^2}\right) \cdot \frac{1}{c^2} + O_\nu\left(\frac{1}{c^3}\right) + o_{c,\nu}(1).$$

Proof. Let $\varepsilon > 0$ be a positive parameter that depends on d as follows: $\varepsilon = o(1)$ and $\varepsilon = \omega(d^{-1/2})$. Let $\mathcal{H}$ be a family of hash functions described by Algorithm 1 (the pseudocode describes how to sample $h \sim \mathcal{H}$).

Algorithm 1 Gaussian partitioning

$\mathcal{P} \leftarrow \emptyset$ $\triangleright$ eventually, $\mathcal{P}$ will be a partition of $\eta c \cdot S^{d-1}$
while $\bigcup \mathcal{P} \neq \eta c \cdot S^{d-1}$ **do** $\triangleright$ we denote $\bigcup \mathcal{P}$ the union of all sets that belong to $\mathcal{P}$
     sample $w \sim N(0,1)^d$
     $S \leftarrow \left\{ u \in \eta c \cdot S^{d-1} \mid \langle u, w \rangle \geq \eta c \cdot \varepsilon \sqrt{d} \right\} \setminus \bigcup \mathcal{P}$
     if $S \neq \emptyset$ **then**
         $\mathcal{P} \leftarrow \mathcal{P} \cup \{S\}$
     end if
end while
define h to be the function that maps a point $u \in \eta c \cdot S^{d-1}$ to the part of $\mathcal{P}$ that it belongs to

---

Clearly for $u, v \in \eta c \cdot S^{d-1}$ with angle α between them

$$\Pr_{h \sim \mathcal{H}}[h(u) = h(v)] = \frac{\Pr_{w \sim N(0,1)^d}[\langle u, w \rangle \geq \eta c \cdot \varepsilon \sqrt{d} \wedge \langle v, w \rangle \geq \eta c \cdot \varepsilon \sqrt{d}]}{\Pr_{w \sim N(0,1)^d}[\langle u, w \rangle \geq \eta c \cdot \varepsilon \sqrt{d} \vee \langle v, w \rangle \geq \eta c \cdot \varepsilon \sqrt{d}]}$$

$$= \Theta(1) \cdot \frac{\Pr_{X,Y \sim N(0,1)}[X \geq \varepsilon \sqrt{d} \wedge \cos \alpha \cdot X - \sin \alpha \cdot Y \geq \varepsilon \sqrt{d}]}{\Pr_{X \sim N(0,1)}[X \geq \varepsilon \sqrt{d}]}$$

$$= \Theta(\varepsilon \sqrt{d}) \cdot \frac{\Pr_{X,Y \sim N(0,1)}[X \geq \varepsilon \sqrt{d} \wedge \cos \alpha \cdot X - \sin \alpha \cdot Y \geq \varepsilon \sqrt{d}]}{e^{-\varepsilon^2 d/2}}. \quad (1)$$

In the last equality we use Lemma 8 and the fact that $\varepsilon = \omega(d^{-1/2})$.

The following two lemmas allow us to estimate the numerator of the right-hand side of (1).

Lemma 12.

$$\Pr_{X,Y \sim N(0,1)}[X \geq \varepsilon \sqrt{d} \wedge \cos \alpha \cdot X - \sin \alpha \cdot Y \geq \varepsilon \sqrt{d}] = O\left(\frac{e^{-\varepsilon^2 d \cdot (1 + \tan^2 \frac{\alpha}{2})/2}}{\varepsilon \sqrt{d}}\right).$$

Proof.

$$\Pr_{X,Y \sim N(0,1)}[X \geq \varepsilon \sqrt{d} \wedge \cos \alpha \cdot X - \sin \alpha \cdot Y \geq \varepsilon \sqrt{d}]$$

$$\leq \Pr_{X,Y \sim N(0,1)}[(1 + \cos \alpha) \cdot X - \sin \alpha \cdot Y \geq 2\varepsilon \sqrt{d}]$$

$$= \Pr_{Z \sim N(0,1)}[\sqrt{(1 + \cos \alpha)^2 + \sin^2 \alpha} \cdot Z \geq 2\varepsilon \sqrt{d}]$$

$$= \Pr_{Z \sim N(0,1)}[\sqrt{2 \cdot (1 + \cos \alpha)} \cdot Z \geq 2\varepsilon \sqrt{d}] = O\left(\frac{e^{-\varepsilon^2 d \cdot (1 + \tan^2 \frac{\alpha}{2})/2}}{\varepsilon \sqrt{d}}\right)$$

In the last equality we used Lemma 8, the fact that $\varepsilon = \omega(d^{-1/2})$ and the identity $\frac{2}{1 + \cos \alpha} = 1 + \tan^2 \frac{\alpha}{2}$. $\qquad \square$

Lemma 13. *If $0 \leq \alpha < \alpha_0$ for some* constant $0 < \alpha_0 < \pi/2$, *then*

$$\Pr_{X,Y \sim N(0,1)}[X \geq \varepsilon\sqrt{d} \wedge \cos\alpha \cdot X - \sin\alpha \cdot Y \geq \varepsilon\sqrt{d}] = \Omega\left(\frac{e^{-\varepsilon^2 d \cdot (1+\tan^2 \frac{\alpha_0}{2})/2}}{\varepsilon^2 d \cdot \tan\frac{\alpha_0}{2}}\right).$$

Proof.

$$\Pr_{X,Y \sim N(0,1)}[X \geq \varepsilon\sqrt{d} \wedge \cos\alpha \cdot X - \sin\alpha \cdot Y \geq \varepsilon\sqrt{d}]$$

$$\geq \Pr_{X,Y \sim N(0,1)}[X \geq \varepsilon\sqrt{d} \wedge Y \leq -\tan\frac{\alpha}{2} \cdot \varepsilon\sqrt{d}]$$

$$= \Pr_{X \sim N(0,1)}[X \geq \varepsilon\sqrt{d}] \cdot \Pr_{Y \sim N(0,1)}[Y \geq \tan\frac{\alpha}{2} \cdot \varepsilon\sqrt{d}]$$

$$\geq \Pr_{X \sim N(0,1)}[X \geq \varepsilon\sqrt{d}] \cdot \Pr_{Y \sim N(0,1)}[Y \geq \tan\frac{\alpha_0}{2} \cdot \varepsilon\sqrt{d}] = \Omega\left(\frac{e^{-\varepsilon^2 d \cdot (1+\tan^2 \frac{\alpha_0}{2})/2}}{\varepsilon^2 d \cdot \tan\frac{\alpha_0}{2}}\right).$$

In the first inequality we use that for $\alpha < \alpha_0 < \pi/2$ the right-hand side event implies the left-hand side event. Indeed,

$$\cos\alpha \cdot X - \sin\alpha \cdot Y \geq \cos\alpha \cdot \varepsilon\sqrt{d} + \sin\alpha \cdot \tan\frac{\alpha}{2} \cdot \varepsilon\sqrt{d} = \varepsilon\sqrt{d},$$

since $\cos\alpha, \sin\alpha > 0$. In the last equality we used Lemma 8, the fact that α_0 is constant and $\varepsilon = \omega(d^{-1/2})$. $\qquad\square$

Thus, combining (1), Lemma 12 and Lemma 13, we have the following estimates on the probability of collision.

Lemma 14. *One has*

$$\ln\frac{1}{\Pr_{h \sim \mathcal{H}}[h(u) = h(v)]} \geq \frac{\varepsilon^2 d}{2} \cdot \tan^2\frac{\alpha}{2} - O(1);$$

and if $\alpha < \alpha_0$ for some constant $0 < \alpha_0 < \pi/2$, *then*

$$\ln\frac{1}{\Pr_{h \sim \mathcal{H}}[h(u) = h(v)]} \leq \frac{\varepsilon^2 d}{2} \cdot \tan^2\frac{\alpha_0}{2} + \ln\left(\varepsilon\sqrt{d} \cdot \tan\frac{\alpha_0}{2}\right) + O(1).$$

Since

$$\tan^2\frac{\alpha}{2} = \frac{\|u - v\|^2/(\eta c)^2}{4 - \|u - v\|^2/(\eta c)^2},$$

by setting $\varepsilon = d^{-1/4}$ and invoking Lemma 14 for the angles that correspond to distances 1 and c, we have

$$\ln\frac{1}{p_1} \leq \frac{\sqrt{d}}{2} \cdot \frac{1/(\eta c)^2}{4 - 1/(\eta c)^2} + O_{c,\nu}(\ln d),$$

$$\ln\frac{1}{p_2} \geq \frac{\sqrt{d}}{2} \cdot \frac{1/\eta^2}{4 - 1/\eta^2} - O(1).$$

Note that here we use that c is large enough, since we must have $\alpha_0 < \pi/2$ in order to be able to apply Lemma 14.

Thus, we have $p_1 = \exp(-o_{c,\nu}(d))$. A similar estimate holds for p_2 provided that η is separated from $1/2$ (but we do not really need it). Therefore

$$\rho = \frac{\ln(1/p_1)}{\ln(1/p_2)} = \frac{4 - 1/\eta^2}{4 - 1/(\eta c)^2} \cdot \frac{1}{c^2} + o_{c,\nu}(1) = \left(1 - \frac{1}{4\eta^2}\right) \cdot \frac{1}{c^2} + O_\nu\left(\frac{1}{c^3}\right) + o_{c,\nu}(1).$$

$\square$

Remark: we could have had $O_\nu(1/c^4)$ term in the expression for ρ, but we state the theorem with $O_\nu(1/c^3)$ in order to be consistent with the next theorem.

Now we show how to convert this "idealized" family to a real one.

Theorem 15. *For a sufficiently large c, every $\nu \geq 1/2$ and $1/2 \leq \eta \leq \nu$ there exists an LSH family $\mathcal{H}$ for*

$$\left\{x \in \mathbb{R}^d \mid \|x\| \in [\eta c - 1; \eta c + 1]\right\}$$

with the ℓ_2 norm such that

- *it satisfies the conclusion of Theorem 11;*

- *for every $k \in \mathbb{N}$ one can sample a function from $\mathcal{H}$ in time $\exp(o(d))$, store it in space $\exp(o(d))$ and query in time $\exp(o(d))$.*

Proof. We use the family from the proof of Theorem 11, but with two modifications. First, if we want to compute $h(x)$ for $h \sim \mathcal{H}$, then before doing so, we normalize x to the length ηc. Second, in Algorithm 1 instead of checking the condition $\bigcup \mathcal{P} = \eta c \cdot S^{d-1}$, we simply run the partitioning process for $\exp(o(d))$ steps. Namely, we require that after the end the probability of the event $\bigcup \mathcal{P} = \eta c \cdot S^{d-1}$ is at least $1 - \exp(-d)$ (one can see that this will be the case after $\exp(o(d))$ steps by a standard ε-net argument). Such a high probability means that this LSH family achieves the same parameters as the one from Theorem 11. Clearly, such a function can be stored in space $\exp(o(d))$ and queried in time $\exp(o(d))$.

It is left to argue that normalizing a vector before computing h does not affect the quality (namely, we are interested in p_1 and ρ) by a lot.

Lemma 16. *For any vectors u and v,*

$$\|u/\|u\| - v/\|v\|\|^2 = \frac{1}{\|u\| \cdot \|v\|}\left(\|u - v\|^2 - (\|u\| - \|v\|)^2\right)$$

Proof.

$$\|u/\|u\| - v/\|v\|\|^2 = 2 - \frac{2\langle u, v\rangle}{\|u\| \cdot \|v\|} = \frac{1}{\|u\| \cdot \|v\|}\left(\|u - v\|^2 - (\|u\| - \|v\|)^2\right).$$

$\square$

By the above lemma, one can check that for $u, v \in \left\{x \in \mathbb{R}^d \mid \|x\| \in [\eta c - 1; \eta c + 1]\right\}$

- if $\|u - v\| \leq 1$, then $(\eta c \cdot \|u/\|u\| - v/\|v\|\|)^2 \leq \frac{(\eta c)^2}{(\eta c - 1)^2} \leq 1 + O_\nu\left(\frac{1}{c}\right)$

- if $\|u - v\| \geq c$, then $(\eta c \cdot \|u/\|u\| - v/\|v\|\|)^2 \geq \frac{(\eta c)^2}{(\eta c + 1)^2}(c^2 - 4) \geq c^2 \cdot \left(1 - O_\nu\left(\frac{1}{c}\right)\right).$

Clearly, from these inequalities we can see that the conclusion of Theorem 11 is still true for our case. $\square$

4 Two-level hashing

We now describe our near neighbor data structure. It is composed of several independent data structures, where each one is a two-level hashing scheme, described next. We will conclude with proving our main theorem for ANN search.

First, we provide some very high-level intuition.

Intuition

The general approach can be seen as using LSH scheme composed of two levels: the hash function is $h = (h_C, h_G)$ chosen from two families $h_C \in \mathcal{H}^{k'}, h_G \in \mathcal{G}^l$ for some k', l, where $\mathcal{H}$ is the "ball carving LSH" from Theorem 6, and $\mathcal{G}$ is the "Gaussian LSH" from Theorem 15. In particular, the hash function $h_G(p)$ will depend on the bucket $h_C(p)$ and the other dataset points in the bucket $h_C(p)$. Intuitively, the "outer level" hash function h_C performs a partial standard LSH partitioning (with $\rho \approx 1/c^2$), but also has the role of improving the "geometry" of the points (namely, the points in a buckets roughly will have a bounded diameter). After an application of h_C, the pointset (inside a fixed bucket) has bounded diameter, allowing us to use the improved Gaussian LSH partitioning ("inner level"), with $\rho < 1/c^2$.

In more detail, first, let us recall the main idea of the proof of Theorem 5. Suppose that $\mathcal{H}$ is, say, a family from Theorem 6. Then, we choose k to be an integer such that for every $p, q \in \mathbb{R}^d$ with $\|p - q\| \geq c$ we have

$$\Pr_{h \sim \mathcal{H}^{\otimes k}}[h(p) = h(q)] \approx n^{-1}. \tag{2}$$

Then, by Theorem 6, we have for every $p, q \in \mathbb{R}^d$ with $\|p - q\| \leq 1$

$$\Pr_{h \sim \mathcal{H}^{\otimes k}}[h(p) = h(q)] \approx n^{-1/c^2}.$$

Now suppose we hash all the points using a function $h \sim \mathcal{H}^{\otimes k}$. For a query q the average number of "outliers" in a bin that corresponds to q (points p such that $\|p - q\| > c$) is at most 1 due to (2). On the other hand, for a data point p such that $\|p - q\| \leq 1$ the probability of collision is at least n^{-1/c^2}. Thus, we can create n^{1/c^2} independent hash tables, and query them all in time around $O(n^{1/c^2})$. The resulting probability of success is constant.

The above analysis relies on two distance scales: 1 and c. To get a better algorithm for ANN we introduce the third scale: τc for $\tau > 1$ being a parameter. First, we hash all the points using $\mathcal{H}^{\otimes k'}$ (where $k' \approx k/\tau$) so that the collision probabilities are roughly as follows.

Distance	1	c	τc
Probability of collision	$n^{-1/(\tau c)^2}$	n^{-1/τ^2}	n^{-1}

Now we can argue that with high probability any bucket has diameter $O_\tau(c)$. This allows us to use the family from Theorem 15 for each bucket and set probabilities of collision as follows.

Distance	1	c
Probability of collision	$n^{-(1-\Omega_\tau(1))\cdot(1-1/\tau^2)/c^2}$	n^{-1+1/τ^2}

Due to the independence, we expect overall collision probabilities to be products of "outer" collision probabilities from the first table and "inner" probabilities from the second table. Thus, in total, we have the following probabilities.

Algorithm 2 Two-level hashing

1: **function** BUILD(P, τ, T, k, $\widetilde{k_l}$)
2:     sample $h \sim \mathcal{H}_1^{\otimes k}$, where $\mathcal{H}_1$ is a family from Theorem 6 (w.l.o.g. h maps $\mathbb{R}^d$ to $[m]$)
3:     $B_i \leftarrow \{p \in P \mid h(p) = i\}$
4:     **for** $i \leftarrow 1 \ldots m$ **do**
5:         **while** there exists $p_1, p_2 \in B_i$ such that $\|p_1 - p_2\| > \tau c$ **do**
6:             $B_i \leftarrow B_i \setminus \{p_1, p_2\}$
7:         **end while**
8:         **if** $B_i \neq \emptyset$ **then**
9:             let u_i be the center of the smallest enclosing ball of B_i
10:             let $s_i \in B_i$ be the nearest neighbor of u_i
11:             **for** $l \leftarrow 0 \ldots T$ **do**
12:                 $\widetilde{P}_{il} \leftarrow \{p - u_i \mid p \in B_i, c/2 + l - 1 \leq \|p - u_i\| \leq c/2 + l + 1\}$
13:                 sample $\widetilde{h}_{il} \sim \mathcal{H}_2^{\otimes \widetilde{k}_l}$, where $\mathcal{H}_2$ is a family from Theorem 15 for $\eta = 1/2 + l/c$
14:                 $\widetilde{B}_{ilj} \leftarrow \left\{p \in \widetilde{P}_{il} \mid \widetilde{h}_{il}(p) = j\right\}$
15:             **end for**
16:         **end if**
17:     **end for**
18: **end function**
19: **function** QUERY(q, T)
20:     $i \leftarrow h(q)$
21:     **if** $B_i = \emptyset$ **then**
22:         **return** $\perp$
23:     **end if**
24:     **if** $\|q - s_i\| \leq c$ **then**
25:         **return** s_i
26:     **end if**
27:     **for** $l \leftarrow 0 \ldots T$ **do**
28:         **if** $c/2 + l - 1 \leq \|q - u_i\| \leq c/2 + l + 1$ **then**
29:             $j \leftarrow \widetilde{h}_{il}(q - u_i)$
30:             **for** $p \in \widetilde{B}_{ilj}$ **do**
31:                 **if** $\|q - (p + u_i)\| \leq c$ **then**
32:                     **return** $p + u_i$
33:                 **end if**
34:             **end for**
35:         **end if**
36:     **end for**
37:     **return** $\perp$
38: **end function**

Distance	1	c
Probability of collision	$n^{-(1-\Omega_\tau(1))/c^2}$	n^{-1}

Then we argue as before and achieve

$$\rho \approx \frac{1 - \Omega_\tau(1)}{c^2}.$$

This plan is not quite rigorous for several reasons. One of them is we do not properly take care of conditioning on the event "all buckets have low diameter". Nevertheless, in this section we show how to analyze a similar scheme rigorously.

Construction

We want to solve $(c, 1)$-ANN for ℓ_2^d. As a first step, we apply Johnson-Lindenstrauss transform (Theorem 9) and reduce our problem to $(c - 1, 1)$-ANN for $\ell_2^{O_c(\log n)}$ by increasing the failure probability by an arbitrarily small constant. This means that all quantities of order $\exp(o(d))$ are now $n^{o_c(1)}$ (in particular, various parameters of the hash family from Theorem 15). Abusing notation, let us assume that we are solving $(c, 1)$-ANN in $\ell_2^{O_c(\log n)}$.

For the description of preprocessing and query algorithms see Algorithm 2. Roughly speaking, we first hash points using a hash family from Theorem 6 and then for every bucket we utilize a family from Theorem 15 (after some pruning). The hashing scheme has several parameters: τ, T, k and $\widetilde{k}_l$ for $0 \le l \le T$. Let us show how to set them. First, we choose some $\tau > 1$ (we will set a concrete value later). Second, we choose

$$T = \left\lceil \frac{\tau c}{\sqrt{2}} - \frac{c}{2} \right\rceil + 1. \tag{3}$$

Third, we choose k to be smallest positive integer such that

$$\left(\frac{U(\tau c - 1)}{L} \right)^k \le \frac{1}{2n}, \tag{4}$$

where $U(\cdot)$ and L are from Theorem 6. Finally, for every $0 \le l \le T$ we set $\widetilde{k}_l$ to be the smallest positive integer such that for every $u, v \in \mathbb{R}^d$ with $\|p_1\|, \|p_2\| \in [c/2 + l - 1; c/2 + l + 1]$ and $\|p_1 - p_2\| \ge c$ we have

$$U(c)^k \cdot \Pr_{\widetilde{h} \sim \mathcal{H}_2^{\otimes \widetilde{k}_l}} \left[\widetilde{h}(u) = \widetilde{h}(v) \right] \le \frac{1}{3n}, \tag{5}$$

where $\mathcal{H}_2$ is a family from Theorem 15 for $\eta = 1/2 + l/c$.

It is immediate to see that, if the query algorithm outputs some point p, then $p \in P$ and p is within distance c from a query.

Auxiliary lemmas

Lemma 17. *After the preprocessing one has for every $1 \le i \le m$*

$$B_i = \bigcup_{0 \le l \le T} \widetilde{P}_{il}.$$

Proof. This follows from Jung's theorem (Theorem 10).

Indeed, after the lines 5–7 the diameter of B_i is at most τc. Thus, the radius of the smallest enclosing ball is at most $\tau c/\sqrt{2}$. It means that for any $l > T$ the set $\widetilde{P}_{il}$ is empty. $\qquad\square$

Lemma 18. *If $\mathcal{U}$ and $\mathcal{V}$ are two events with $\Pr[\mathcal{V}] < 1$, then*

$$\Pr[\mathcal{U} \vee \mathcal{V}] \geq \Pr[\mathcal{U} \mid \neg\mathcal{V}].$$

Proof.

$$
\begin{aligned}
\Pr\left[\mathcal{U} \vee \mathcal{V}\right] &= \Pr\left[\mathcal{V}\right] + \Pr\left[\mathcal{U} \mid \neg\mathcal{V}\right]\Pr\left[\neg\mathcal{V}\right] \\
&\geq \Pr\left[\mathcal{U} \mid \neg\mathcal{V}\right]\Pr\left[\mathcal{V}\right] + \Pr\left[\mathcal{U} \mid \neg\mathcal{V}\right]\Pr\left[\neg\mathcal{V}\right] \\
&= \Pr\left[\mathcal{U} \mid \neg\mathcal{V}\right]
\end{aligned}
$$

$\square$

Collision probabilities

Suppose that $q \in \mathbb{R}^d$ is a query and $p \in P$ is a data point. Let us introduce four events:

- $\mathcal{A}$ stands for "$h(p) = h(q)$";

- $\mathcal{B}$: "for every $p' \in P$ such that $\|p' - q\| > \tau c - 1$ we have $h(p') \neq h(q)$";

- $\mathcal{C}$: "we iterate through p in the line 30 of Algorithm 2";

- $\mathcal{D}$: "$B_{h(q)} \neq \emptyset$ and $\|q - s_{h(q)}\| \leq c$".

Lemma 19. *If $\|p - q\| \geq c$, then*
$$\Pr[\mathcal{C}] \leq 1/n. \tag{6}$$

Proof. Since $\mathcal{C}$ implies $\mathcal{A}$, we have

$$\Pr\left[\mathcal{C}\right] = \Pr\left[\mathcal{A}\right]\Pr\left[\mathcal{C} \mid \mathcal{A}\right]. \tag{7}$$

By Theorem 6 $\Pr\left[\mathcal{A}\right] \leq U(c)^k$. Moreover, if we denote

$$W_q = \left\{l \in \mathbb{Z}_+ \mid c/2 + l - 1 \leq \|q - u_{h(q)}\| \leq c/2 + l + 1\right\},$$

then

$$\Pr\left[\mathcal{C} \mid \mathcal{A}\right] = \sum_{l \in W_q} \Pr\left[p \in \widetilde{P}_{h(q)l} \wedge \widetilde{h}_{h(q)l}(p) = \widetilde{h}_{h(q)l}(q)\right].$$

Since $|W_q| \leq 3$ and due to (7) and (5) we have (6). $\qquad\square$

Lemma 20. *If $\|p - q\| \leq 1$, then*

$$\Pr[\mathcal{C} \vee \mathcal{D}] \geq n^{-\left(1 - \frac{1}{2\tau^2} + \frac{1}{2\tau^4}\right)\cdot\frac{1}{c^2} + O_\tau\left(\frac{1}{c^3}\right) + o_{c,\tau}(1)}.$$

Proof. Using Lemma 18, we get

$$\Pr\left[\mathcal{C} \vee \mathcal{D}\right] \geq \Pr\left[\mathcal{A} \wedge \mathcal{B} \wedge (\mathcal{C} \vee \mathcal{D})\right] = \Pr\left[\mathcal{A}\right]\Pr\left[\mathcal{B} \mid \mathcal{A}\right]\Pr\left[\mathcal{C} \vee \mathcal{D} \mid \mathcal{A} \wedge \mathcal{B}\right]$$
$$\geq \Pr\left[\mathcal{A}\right]\Pr\left[\mathcal{B} \mid \mathcal{A}\right]\Pr\left[\mathcal{C} \mid \mathcal{A} \wedge \mathcal{B} \wedge \neg\mathcal{D}\right]. \quad (8)$$

In the following three lemmas we lower bound the right-hand side of (8).

Lemma 21. $\Pr\left[\mathcal{A}\right] \geq L^k$

Proof. This follows immediately from Theorem 6. □

Lemma 22. $\Pr\left[\mathcal{B} \mid \mathcal{A}\right] \geq 1/2$

Proof.

$$\Pr\left[\neg\mathcal{B} \mid \mathcal{A}\right] \leq \sum_{\substack{p' \in P \\ \|p'-q\|>\tau c-1}} \Pr\left[h(p') = h(q) \mid h(p) = h(q)\right]$$

$$\leq \sum_{\substack{p' \in P \\ \|p'-q\|>\tau c-1}} \frac{\Pr\left[h(p') = h(q)\right]}{\Pr\left[h(p) = h(q)\right]} \leq n \cdot \left(\frac{U(\tau c - 1)}{L}\right)^k \leq 1/2,$$

where the penultimate inequality is due to Lemma 21, and the last one is due to (4). □

Lemma 23.
$$\Pr\left[\mathcal{C} \mid \mathcal{A} \wedge \mathcal{B} \wedge \neg\mathcal{D}\right] \geq \left(\frac{1}{3n \cdot U(c)^k}\right)^{\left(1-\frac{1}{2\tau^2}\right) \cdot \frac{1}{c^2} + O_\tau\left(\frac{1}{c^3}\right) + o_{c,\tau}(1)}$$

Proof. First, since we condition on $\mathcal{A} \wedge \mathcal{B}$, we have that $p \in B_{h(q)}$ after the preprocessing. Indeed, $\mathcal{A}$ implies that $p \in B_{h(q)}$ in the beginning of the preprocessing, and $\mathcal{B}$ together with $\|p - q\| \leq 1$ imply that this will be the case in the end as well.

Second, let us prove that $\|p - u_{h(q)}\| \geq c/2 - 1$ and $\|q - u_{h(q)}\| \geq c/2 - 1$. Indeed,

$$c \leq \|q - s_{h(q)}\| \leq \|q - u_{h(q)}\| + \|s_{h(q)} - u_{h(q)}\| \leq \|q - u_{h(q)}\| + \|p - u_{h(q)}\|$$
$$\leq 2 \cdot \|p - u_{h(q)}\| + \|p - q\| \leq 2 \cdot \|p - u_{h(q)}\| + 1,$$

where the first inequality follows from $\neg\mathcal{D}$, and the third inequality follows from the fact that p is not filtered from $B_{h(q)}$ and the definition of $s_{h(q)}$. As a result, we have $\|p - u_{h(q)}\| \geq (c-1)/2 \geq c/2 - 1$. Similarly, we prove $\|q - u_{h(q)}\| \geq c/2 - 1$.

As a result, we have that for some $0 \leq l \leq T$, $p \in \widetilde{P}_{h(q)l}$ and, moreover,

$$c/2 + l - 1 \leq \|q - u_{h(q)}\| \leq c/2 + l + 1,$$

so

$$\Pr\left[\mathcal{C} \mid \mathcal{A} \wedge \mathcal{B} \wedge \neg\mathcal{D}\right] \geq \Pr\left[\widetilde{h}_{h(q)l}(p) = \widetilde{h}_{h(q)l}(q)\right],$$

but due to Theorem 15, (5), (3) and the minimality of k_l this implies

$$\Pr\left[\mathcal{C} \mid \mathcal{A} \wedge \mathcal{B} \wedge \neg\mathcal{D}\right] \geq \left(\frac{1}{3n \cdot U(c)^k}\right)^{\left(1-\frac{1}{2\tau^2}\right) \cdot \frac{1}{c^2} + O_\tau\left(\frac{1}{c^3}\right) + o_{c,\tau}(1)}, \quad (9)$$

13

since we can apply Theorem 15 for

$$\eta = \frac{1}{2} + \frac{l}{c} \le \frac{1}{2} + \frac{T}{c} \le \frac{\tau}{\sqrt{2}} + O_\tau\left(\frac{1}{c}\right),$$

and as a result we have

$$\rho \le \left(1 - \frac{1}{4\eta^2}\right) \cdot \frac{1}{c^2} + O_\tau\left(\frac{1}{c^3}\right) + o_{c,\tau}(1) \le \left(1 - \frac{1}{2\tau^2}\right) \cdot \frac{1}{c^2} + O_\tau\left(\frac{1}{c^3}\right) + o_{c,\tau}(1),$$

which in turn implies (9). □

Now we can finish the proof of Lemma 20. Combining Lemmas 21, 22 and 23 we get

$$\Pr[\mathcal{C} \vee \mathcal{D}] \ge L^k \cdot \left(\frac{1}{n \cdot U(c)^k}\right)^{\left(1 - \frac{1}{2\tau^2}\right) \cdot \frac{1}{c^2} + O_\tau\left(\frac{1}{c^3}\right) + o_{c,\tau}(1)}. \tag{10}$$

It is known from [AI06] that
$$U(x) = L^{x^2 + o_x(1)},$$

so from (4) and the minimality of k we get

$$L^k = n^{-\frac{1}{(\tau c - 1)^2 - 1} + o_{c,\tau}(1)} = n^{-\frac{1}{\tau^2 c^2} + O_\tau\left(\frac{1}{c^3}\right) + o_{c,\tau}(1)}, \tag{11}$$

and

$$U(c)^k = n^{-\frac{c^2}{(\tau c - 1)^2 - 1} + o_{c,\tau}(1)} = n^{-\frac{1}{\tau^2} + O_\tau\left(\frac{1}{c}\right) + o_{c,\tau}(1)}. \tag{12}$$

Combining (10), (11) and (12) we get

$$\Pr[\mathcal{C} \vee \mathcal{D}] \ge n^{-\left(1 - \frac{1}{2\tau^2} + \frac{1}{2\tau^4}\right) \cdot \frac{1}{c^2} + O_\tau\left(\frac{1}{c^3}\right) + o_{c,\tau}(1)}.$$

□

The main result

Finally, we formulate and prove the main result.

Theorem 24. *There exists a data structure for $(c, 1)$-ANN for ℓ_2^d with*

- *preprocessing time $O_c(n^{2+\rho} + nd \log n)$,*
- *space $O_c(n^{1+\rho} + d \log n)$,*
- *query time $O_c(n^\rho + d \log n)$,*

where

$$\rho \le \frac{7/8}{c^2} + O\left(\frac{1}{c^3}\right) + o_c(1). \tag{13}$$

14

Proof. First, we analyze the running time of one two-level hash table.

The dimension reduction step takes time $O_c(nd \log n)$ by Theorem 9.

The preprocessing takes $O_c(n^{1+o_{c,\tau}(1)})$ time plus the time needed for the lines 5–7 and the line 9. It is straightforward to implement lines 5–7 in time $O_c(n^2 \log n)$ and one can use an algorithm from [GLS88] for finding u_i with running time $O_c(n \log^{O(1)} n)$.

The query algorithm takes time $O_c(n^{o_{c,\tau}(1)} + d \log n)$ in expectation due to Lemma 19. This is because in line 30 we iterate over at most 1 point on average that is not an approximate near neighbor.

By Lemma 20 the probability of finding an approximate near neighbor is at least

$$Q = n^{-\left(1 - \frac{1}{2\tau^2} + \frac{1}{2\tau^4}\right) \cdot \frac{1}{c^2} + O_\tau\left(\frac{1}{c^3}\right) + o_{\tau,c}(1)}.$$

In order to make the probability of success constant we build and query $1/Q$ independent copies of the two-level data structure. As a result we get the desired bounds with

$$\rho = \rho(\tau) \le \left(1 - \frac{1}{2\tau^2} + \frac{1}{2\tau^4}\right) \cdot \frac{1}{c^2} + O_\tau\left(\frac{1}{c^3}\right) + o_{\tau,c}(1).$$

The bound stated in (13) is obtained by setting $\tau = \sqrt{2}$.

Note that the above bound on the query time is *in expectation*, but it is also possible to modify the algorithm slightly to get a worst-case bound. The algorithm still iterates over $1/Q$ tables but stops after looking at $3/Q + 1$ points without finding an approximate near neighbor. The expected number of points the algorithm has to look at that are not an approximate near neighbor is at most $1/Q$. By Markov's inequality, with probability at least $2/3$, the algorithm doesn't look at more than $3/Q$ points that are not an approximate near neighbor. In each two-level table, the probability that the algorithm fails to find an approximate near neighbor is at most $1 - Q$. Thus, the probability it fails in all Q tables is at most $(1 - Q)^{1/Q} \le 1/e$. Overall, with probability at least $1 - 1/3 - 1/e$, the algorithm finds an approximate near neighbor without looking at more than $3/Q + 1$ points. □

Remark: if one is willing to have quasi-linear preprocessing time, then it is possible to modify Algorithm 2 slightly to achieve

$$\rho \le \frac{15/16}{c^2} + O\left(\frac{1}{c^3}\right) + o_c(1),$$

while having preprocessing time $O_c(n^{1+\rho} + nd \log n)$. The idea is to choose an arbitrary point $p \in B_i$ after the initial hashing and then remove from B_i points that are further from p than τc. After this filtering the algorithm is the same as before. We save in preprocessing time, since we no longer need to run lines 5–7.

The details are almost the same as in the proof of Theorem 24 and thus omitted.

5   Acknowledgments

References

[AI06]   Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS '2006)*, pages 459–468, 2006.

[And09]   Alexandr Andoni. *Nearest Neighbor Search: the Old, the New, and the Impossible*. PhD thesis, Massachusetts Institute of Technology, 2009.

[Bro98]   Andrei Z. Broder. Filtering near-duplicate documents. *Proceedings of the 1st International Conference on Fun with Algorithms (FUN '1998)*, 1998.

[Cla88]   Kenneth L. Clarkson. A randomized algorithm for closest-point queries. *SIAM Journal on Computing*, 17(4):830–847, 1988.

[CR93]   Andrea Califano and Isidore Rigoutsos. FLASH: a fast look-up algorithm for string homology. In *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology (ISMB '1993)*, pages 56–64, 1993.

[DG03]   Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures and Algorithms*, 22(1):60–65, 2003.

[Dub10]   Moshe Dubiner. Bucketing coding and information theory for the statistical high-dimensional nearest-neighbor problem. *IEEE Transactions on Information Theory*, 56(8):4166–4179, 2010.

[fmd13]   *Frontiers in Massive Data Analysis*. The National Academies Press, 2013.

[GLS88]   Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*. Springer-Verlag, 1988.

[GPY94]   Daniel H. Greene, Michal Parnas, and F. Frances Yao. Multi-index hashing for information retrieval. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS '1994)*, pages 722–731, 1994.

[HPIM12]   Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: towards removing the curse of dimensionality. *Theory of Computing*, 8(1):321–350, 2012.

[IM98]   Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the 30th ACM Symposium on the Theory of Computing (STOC '1998)*, pages 604–613, 1998.

[JL84]   William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Connecticut, 1982)*, volume 26 of *Contemporary Mathematics*, pages 189–206. 1984.

[KMS98]   David R. Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *Journal of the ACM*, 45(2):246–265, 1998.

[KWZ95]   Richard M. Karp, Orli Waarts, and Geoffrey Zweig. The bit vector intersection problem (preliminary version). In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS '1995)*, pages 621–630, 1995.

[LLR95]   Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[Mat02]   Jiří Matoušek. *Lectures on Discrete Geometry*. Springer, 2002.

[Mei93]   Stefan Meiser. Point location in arrangements of hyperplanes. *Information and Computation*, 106(2):286–303, 1993.

[MNP07]   Rajeev Motwani, Assaf Naor, and Rina Panigrahy. Lower bounds on locality sensitive hashing. *SIAM Journal on Discrete Mathematics*, 21(4):930–935, 2007.

[OWZ11]   Ryan O'Donnell, Yi Wu, and Yuan Zhou. Optimal lower bounds for locality sensitive hashing (except when q is tiny). In *Proceedings of Innovations in Computer Science (ICS '2011)*, pages 275–283, 2011.

[PRR95]   Ramamohan Paturi, Sanguthevar Rajasekaran, and John H. Reif. The light bulb problem. *Information and Computation*, 117(2):187–192, 1995.

[PTW08]   Rina Panigrahy, Kunal Talwar, and Udi Wieder. A geometric approach to lower bounds for approximate near-neighbor search and partial match. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS '2008)*, pages 414–423, 2008.

[PTW10]   Rina Panigrahy, Kunal Talwar, and Udi Wieder. Lower bounds on near neighbor search via metric expansion. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS '2010)*, pages 805–814, 2010.

[Val12]   Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '2012)*, pages 11–20, 2012.