Approximate Nearest Neighbor Search via Locality Sensitive Hashing

Rishav Gupta, Aditya Morolia, Saswata Mukherjee, Zhan Yu

April 6, 2025

1 Introduction

We study the *approximate nearest neighbor* (ANN) problem.

Definition 1.1 ((c, r)-approximate nearest neighbor problem)**.** *Consider the metric space $(X, \text{dist}(\cdot, \cdot))$. Given a set $S \subseteq X$ of n points and an $r \in \mathbb{R}$, and a failure probability f, construct a data structure Q such that upon receiving a query $y \in X$, if $\exists x \in S$ such that $\text{dist}(x, y) \leq r$, return any point $x' \in S$ such that $\text{dist}(x', y) \leq c \cdot r$, with probability at least $1 - f$.*

An important technique to solve ANN is using *locality sensitive hashing* (LSH), which are functions that map close points to the same value and far points to different values with high probability.

Definition 1.2 (LSH Family)**.** *A family of functions $\mathcal{H} = \{h : X \to \mathbb{Z}\}$ is (r, cr, p_1, p_2)-LSH if for all $x, y \in X$:*

- $\text{dist}(x, y) \leq r \Rightarrow \Pr_{h \sim \mathcal{H}}[h(x) = h(y)] \geq p_1$.
- $\text{dist}(x, y) > cr \Rightarrow \Pr_{h \sim \mathcal{H}}[h(x) = h(y)] \leq p_2$.

Given some (r, cr, p_1, p_2)-LSH family with $p_1 \geq p_2$ and we would like to boost the probability p_1 close to 1 and diminish the probability p_2 close to 0. Then we can solve (c, r)-approximate nearest neighbor problem in space $\tilde{\mathcal{O}}(n^{1+\rho})$ and query time $\tilde{\mathcal{O}}(n^{\rho})$ [IM98]. Here ρ is an important parameter of the LSH family defined as $\rho = \frac{\log(1/p_1)}{\log(1/p_2)}$, which determines the "quality" of the LSH families used. Constructing the LSH family with smaller ρ yields more efficient ANN algorithms. Consequently, significant effort has been dedicated to determining the optimal value of ρ for LSH.
We first study a lower bound on the parameter ρ for any LSH family presented in [MNP08]. They show that, $\rho \geq \frac{0.462}{c^s}$ for any (r, cr, p_1, p_2)-LSH family, when $X = \{0, 1\}^d$ and the distance function is induced by the ℓ_s norm. This work was followed by an improved lower bound of [OWZ14], which showed $\rho \geq \frac{1}{c^s}$ when the (far) points are correlated. We then present an interesting observation of [AINR14], that this lower bound does not translate to a lower bound for ANN. This is based on the observation that the dataset is assumed to be correlated (structured) in the lower bound of [OWZ14], which can be exploited to construct better, *data dependent* hash functions. For ℓ_2 metric space, they achieved $\rho \leq (7/8c^2)$.

2 Preliminaries

Theorem 2.1. *[KMS98] For every $t > 0$*

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

where $\mathcal{N}(0, 1)$ is the normal distribution with mean 0 and variance 1.

Theorem 2.2. *[JL84] For every $d \in \mathbb{N}$ and $\varepsilon, \delta > 0$, there exists a distribution over linear maps $A : \mathbb{R}^d \to \mathbb{R}^{\log(1/\delta)/\varepsilon^2}$ so that for every $x \in \mathbb{R}^d$, one has $\Pr_A[\|Ax\| \in (1 \pm \varepsilon)\|x\|] \geq 1 - \delta$. Moreover this map can be sampled in time $O(d \log(1/\delta)/\varepsilon^2)$*

Theorem 2.3 (Jung's theorem). *Every subset of $\mathbb{R}^d$ of diameter Δ can be enclosed by a ball of radius $\Delta/\sqrt{2}$.*

3 Lower Bound for LSH

In this section, we present a lower bound for the ρ parameter of LSH on the ℓ_s metric space as proven in [MNP08]. Note that for binary vectors, the ℓ_1-distance is equivalent to the Hamming distance. Also, for any $s \geq 1$, we can relate ℓ_s-distance to ℓ_1-distance as $\|x - y\|_s = \|x - y\|_1^{1/s}$ for any $x, y \in \{0,1\}^d$ so proving results for the Hamming metric space will give us lower bounds for the general case.
First we redefine the ρ parameter associated with the ℓ_s metric space as $d \to \infty$:

$$\rho_s = \lim_{d \to \infty} \sup \Big\{ \frac{\log(1/p_1)}{\log(1/p_2)} : \exists (r, cr, p_1, p_2)\text{-LSH family } \mathcal{H} \text{ on } (\{0,1\}^d, \|\cdot\|_s) \Big\}.$$

The lower bound for LSH can be stated as follows.

Theorem 3.1 (Lower bound for LSH). *For every (r, cr, p_1, p_2)-LSH family with $c \geq 1$ on the $(\{0,1\}^d, \|\cdot\|_s)$ metric space,*

$$\rho_s \geq \frac{e^{\frac{1}{c^s}} - 1}{e^{\frac{1}{c^s}} + 1} \geq \frac{e-1}{e+1} \cdot \frac{1}{c^s} \geq \frac{0.462}{c^s}.$$

Note that the function $f(x) = \frac{e^x - 1}{e^x + 1}$ is concave, which means that $f(t \cdot x) \geq t \cdot f(x)$ for $t \in [0,1]$. Let $t = \frac{1}{c^s}$ and $x = 1$, the second inequality in Theorem 3.1 holds as

$$f\big(\frac{1}{c^s}\big) = \frac{e^{\frac{1}{c^s}} - 1}{e^{\frac{1}{c^s}} + 1} \geq \frac{1}{c^s} \cdot f(1) = \frac{e-1}{e+1} \cdot \frac{1}{c^s}.$$

Now we sketch the proof of Theorem 3.1. Following the above observation, we simply need to show that $\rho_s \geq \frac{e^{1/c^s} - 1}{e^{1/c^s} + 1}$. We break down the proof into two main lemmas, the Hashing Lemma and the Random Walk Lemma. First we prove the Hashing Lemma, which shows that if we choose a random point $x \in \{0,1\}^d$, then the size of the hash bucket

$$H^{-1}(H(x)) \coloneqq \{y \in \{0,1\}^d : h(x) = h(y)\} \text{ for } h \sim \mathcal{H}^1$$

is small.

Lemma 3.2 (Hashing Lemma). *Let $\mathcal{H}$ be an (r, R, p_1, p_2)-LSH family on the Hamming cube $(\{0,1\}^d, \|\cdot\|_1)$, fix a point $x \in \{0,1\}^d$ and let $R < d/2$, then*

$$\mathbb{E}|H^{-1}(H(x))| \leq 2^d (p_2 + e^{-\frac{1}{d}(\frac{d}{2} - R)^2}),$$

We include a proof Appendix A for completeness. Next we need the Random Walk Lemma, which shows the upper bound the probability of r-step random walks from a random point in a hash bucket that end up in the same hash bucket. We need the following definition.

Definition 3.3 (Random walk on Hamming cubes). *On a Hamming cube with vertices $V = \{0,1\}^d$ and edges $E = \{(u,v) \in V \times V : \|u - v\|_1 = 1\}$, a one-step random walk from a vertex $x \in V$ transits to any of the n neighboring vertices with equal probability:*

$$\Pr(u \to v) = \begin{cases} \frac{1}{n}, & \text{if } (u,v) \in E, \\ 0, & \text{otherwise.} \end{cases}$$

[1] We follow the notation from the paper, but clarify that H is a random variable over $\mathcal{H}$, and $H^{-1}H(x)$ is a random variable that denotes a sampled hash bucket to which x can belong to.

Lemma 3.4 (Random Walk Lemma). *Let r be an odd integer. Given $\emptyset \neq B \subseteq \{0,1\}^d$, consider the random variable $Q_B \in \{0,1\}^d$ defined as follows: choose a point $z \in B$ uniformly at random, and perform r-steps of the standard random walk on the Hamming cube starting from z (choose y uniformly from the set of all strings which have hamming distance 1, with z). The point thus obtained will be denoted Q_B. Then,*

$$\Pr\left[Q_B \in B\right] \leq \left(\frac{|B|}{2^d}\right)^{\frac{e^{2r/d}-1}{e^{2r/d}+1}}.$$

We refer the reader to Appendix B for the proof.

Lemma 3.5 (Main Proposition). *Let H be a (r, R, p_1, p_2)-sensitive hash family on the Hamming cube $(\{0,1\}^d, \|\cdot\|_1)$. Assume that r is an odd integer and that $R < \frac{d}{2}$. Then,*

$$p_1 \leq \left(p_2 + e^{-\frac{1}{d}(\frac{d}{2}-R)^2}\right)^{\frac{e^{2r/d}-1}{e^{2r/d}+1}}.$$

Proof. Note that $\Pr[H(W_r(x) = H(x)] \geq p_1$ by the first property of LSH. Taking expectation over the uniform probability measure on $\{0,1\}^d$, we deduce that

$$\begin{aligned}
p_1 &\leq \mathbb{E}_{x\in\{0,1\}^d} \Pr_{H\sim\mathcal{H},W_r}[H(W_r(x)) = H(x)] \\
&= \mathbb{E}_H \Pr_{x,W_r}\left[x \in \{0,1\}^d : W_r(x) \in H^{-1}(H(x))\right] \\
&= \mathbb{E}_H \sum_{k\in\mathbb{N}} \Pr_{x,W_r}\left[x \in \{0,1\}^d : W_r(x) \in H^{-1}(H(x)) \wedge H(x) = k\right] \\
&= \mathbb{E}_H \sum_{k\in\mathbb{N}} \sum_{x\in H^{-1}(k)} \frac{1}{2^d} \Pr_{W_r}\left[W_r(x) \in H^{-1}(k)\right] \\
&= \mathbb{E}_H \sum_{k\in\mathbb{N}} \frac{|H^{-1}(k)|}{2^d} \sum_{x\in H^{-1}(k)} \frac{1}{H^{-1}(k)} \Pr_{W_r}\left[W_r(x) \in H^{-1}(k)\right] \\
&= \mathbb{E}_H \sum_{k\in\mathbb{N}} \frac{|H^{-1}(k)|}{2^d} \cdot \Pr_{W_r}\left[Q_{H^{-1}(k)} \in H^{-1}(k)\right] \\
&\leq \mathbb{E}_H \sum_{k\in\mathbb{N}} \frac{|H^{-1}(k)|}{2^d} \cdot \left(\frac{|H^{-1}(k)|}{2^d}\right)^{\frac{e^{2r/d}-1}{e^{2r/d}+1}} && \text{(Random Walk Lemma)} \\
&= \mathbb{E}_H \mathbb{E}_{x\in\{0,1\}^d}\left[\frac{|H^{-1}(H(x))|}{2^d}\right]^{\frac{e^{2r/d}-1}{e^{2r/d}+1}} \\
&\leq \mathbb{E}_{x\in\{0,1\}^d}\left[\frac{\mathbb{E}_H[|H^{-1}(H(x))|]}{2^d}\right]^{\frac{e^{2r/d}-1}{e^{2r/d}+1}} && \text{(Jensen Inequality)} \\
&\leq \left(p_2 + e^{-\frac{1}{d}(\frac{d}{2}-R)^2}\right)^{\frac{e^{2r/d}-1}{e^{2r/d}+1}} && \text{(Hashing Lemma)}
\end{aligned}$$

□

Then choosing $R \approx \frac{d}{2} - \sqrt{d \log d}$ and $r \approx R/c$ yields the lower bound $\rho_s \geq \frac{e^{1/c^s}-1}{e^{1/c^s}+1}$ with $s = 1$; for general $s > 1$, the inequality follows from the relationship $\|x-y\|_s = \|x-y\|_1^{1/s}$ over $\{0,1\}^d$.

4 Data Dependent Hashing

Till now we have seen lower bounds on ρ for LSH, in context of approximate nearest neighbor search. For the rest of the section, assume $X = \mathbb{R}^d$ and the distance is induced by the ℓ_2 norm. In this section, when we say $\|\cdot\|$, we mean the ℓ_2 norm.
In 2014, [AINR14] has given an approach of *data-dependent hashing* or *two level hashing* based approach for $(c, 1)$-approximate nearest neighbor problem in Euclidean space.

Theorem 4.1. *[AINR14] Given n points in $\mathbb{R}^d$, in ℓ_2 norm, there is an algorithm that solves $(c, 1)$-ANN problem with*

1. ***Pre-processing time:*** $\mathcal{O}_c(n^{2+\rho} + nd \log n)$
2. ***Query time:*** $\mathcal{O}_c(n^{\rho} + d \log n)$
3. ***Space:*** $\mathcal{O}_c(n^{1+\rho} + d \log n)$

where $\rho \leq 7/(8c^2) + O(1/c^3) + o_c(1)$.

Intuitively in this approach at first we are going to sample from some hash family to create hash buckets, that is our first level hashing. And for each hash bucket we independently sample hash functions, depending on the points on the bucket, which is second level hashing. Here we shall see the main ideas of [AINR14].
For the first level hashing the construction of [AI06] has been used, where in ℓ_2 norm, $\rho = 1/c^2 + o_c(1)$ has been achieved. Below we state the formal statement of the theorem.

Theorem 4.2. *[AI06] For every sufficiently large d and n, there is a hash family $\mathcal{H}$ for ℓ_2^d so that,*

1. *$h \leftarrow \mathcal{H}$ can be sampled in time, stored in space and computed in time $t^{\mathcal{O}(t)} \log n + \mathcal{O}(dt)$, for $t = \log^{2/3} n$.*
2. *For any two points u, v, the collision probability $p(.)$ of $\mathcal{H}$ only depends on $\|u - v\|$, where,*
   I. $p(1) \geq L$, where $L = \frac{A}{2\sqrt{t}(1+\epsilon+8\epsilon^2)^{t/2}}$.
   II. For all $c > 1$, $p(c) \leq U(c)$, where $U(c) = \frac{2}{(1+c^2\epsilon)^{t/2}}$.

Here $A \in (0, 1)$ is some absolute constant, and $\epsilon = \Theta(t^{-1/2}) = \Theta(\log^{-1/3} n)$.

Remark 4.3. *The hash family $\mathcal{H}$ is data independent hashing in the following sense that, for any two points in the input, their collision probability only depends on the distance between them. More precisely if distance between two points are $c > 1$, the collision probability is a function on c, which is $U(c)$.*

For the second level hashing they have come up with a new technique, called *Gaussian hashing*. In particular, when all the data points are in a spherical shell of radius ηc and width $\mathcal{O}(1)$, they have got an improvement on the value of ρ, compared to the work [AI06]. We shall see formal statement and proof idea in Section 4.1.

4.1 Gaussian Locality Sensitive Hashing

We now describe the Gaussian LSH scheme of [AINR14]. We wish to come up with a hashing scheme that can give us an advantage over [AI06] for structured data. The structure that we impose is very natural: that the dataset has a bounded diameter. Given this, we can invoke Theorem 2.3 to claim that the points lie on a thin spherical shell of radius $\mathcal{O}(c)$ and width $\mathcal{O}(1)$. Formally speaking, they prove the following theorem. In what follows, we give a proof sketch of the theorem.

Theorem 4.4. *[AINR14] for sufficiently large c, every $\nu > 1/2$ and $1/2 \leq \eta < \nu$, there exists a $(1, c, p_1, p_2)$-sensitive LSH family $\mathcal{H}_G$ for*

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

in ℓ_2 norm so that,

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

2. *One can sample $h \leftarrow \mathcal{H}_G$, in time $\exp(o(d))$, store in space $\exp(o(d))$ and compute in time $\exp(o(d))$.*

Proof. We begin by first constructing such a family on a spherical surface. In what follows, imagine that all the input points lie on a hypersphere, and we want to be able to partition them efficiently into buckets. We do so by proving the following theorem.

Theorem 4.5. *[AINR14] 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*

$$\rho = \frac{\ln(1/p_1)}{\ln(1/p_2)} = \left(1 - \frac{1}{4\eta^2}\right) \cdot \frac{1}{c^2} + \textit{lower order terms}$$

Proof. Let $\varepsilon > 0$ be a positive parameter such that $\varepsilon = o(1), \varepsilon = \omega(d^{-1/2})$. Then, the following algorithm describes how to sample $h \sim \mathcal{H}$.

Algorithm 1 Gaussian partitioning

$\mathcal{P} \leftarrow \emptyset$
while $\bigcup \mathcal{P} \neq \eta c \cdot S^{d-1}$ **do** ▷ Eventually, $\mathcal{P}$ will be a partition of $\eta c \cdot S^{d-1}$
  Sample $w \sim \mathcal{N}(0,1)^d$
  $S \leftarrow \left\{u \in \eta c \cdot S^{d-1} \mid \langle u, w\rangle \geq \eta c \cdot \varepsilon\sqrt{d}\right\} \setminus \bigcup \mathcal{P}$
  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

Now our goal is as follows. There are two points $u, v \in \eta c \cdot S^{d-1}$ with angle α between them. What is the collision probability over the randomness of the hash function? This probability is exactly equal to the probability that they land in the same bucket, given that one of them lands in a particular bucket, over the randomness of the buckets.

$$\Pr_{h\sim\mathcal{H}}[h(u) = h(v)] = \frac{\Pr_{w\sim\mathcal{N}(0,1)^d}[\langle u, w\rangle \geq \eta c \cdot \varepsilon\sqrt{d} \wedge \langle v, w\rangle \geq \eta c \cdot \varepsilon\sqrt{d}]}{\Pr_{w\sim\mathcal{N}(0,1)^d}[\langle u, w\rangle \geq \eta c \cdot \varepsilon\sqrt{d} \vee \langle v, w\rangle \geq \eta c \cdot \varepsilon\sqrt{d}]} \tag{4.1}$$

We first wish to get a handle on the term highlighted in red. Observe that, $\|u\| = \eta c$. Say $\widehat{u} := \frac{u}{\|u\|}$. Then

$$\langle u, w\rangle \geq \eta c\varepsilon\sqrt{d} \implies \langle \widehat{u}, w\rangle \geq \varepsilon\sqrt{d} \implies \sum_j \widehat{u}_j w_j \geq \varepsilon\sqrt{d}.$$

Recall that the coordinates $w_j \sim \mathcal{N}(0,1)$. Thus, the LSH is a convex combination of independent Gaussian samples, which is another Gaussian random variable (say X). By writing $v = (\widehat{u}\cos\alpha + \widehat{u}^{\perp}\sin\alpha)\,\|v\|$, we can replace the second inequality with a similar Gaussian random variable Y[2]. This gives us

$$\Pr_{h\sim\mathcal{H}}[h(u) = h(v)] = \Theta(1) \cdot \frac{\Pr_{X,Y\sim\mathcal{N}(0,1)}[X \geq \varepsilon\sqrt{d} \wedge \varepsilon\sqrt{d}\cos\alpha \cdot X - \sin\alpha \cdot Y \geq \varepsilon\sqrt{d}]}{\Pr_{X\sim\mathcal{N}(0,1)}[X \geq \varepsilon\sqrt{d}]} \tag{4.2}$$

$$= \Theta(\varepsilon\sqrt{d}) \cdot \frac{\Pr_{X,Y\sim\mathcal{N}(0,1)}[X \geq \varepsilon\sqrt{d}\cos\alpha \cdot X - \sin\alpha \cdot Y \geq \varepsilon\sqrt{d}]}{e^{-\varepsilon^2 d/2}}. \quad \text{(using Theorem 2.1)} \tag{4.3}$$

[2]Here we have abused notation a little and treated a random variable the same as a sample from the distribution.

Lemma 4.6. *[AINR14] We can derive the following bounds on the numerator.*

- ***(For bounding collision probability of far points.)***

$$\Pr_{X,Y\sim\mathcal{N}(0,1)}\left[X \geq \varepsilon\sqrt{d}\cos\alpha \wedge X - \sin\alpha \cdot Y \geq \varepsilon\sqrt{d}\right] = \mathcal{O}\left(\frac{e^{-\varepsilon^2 d\cdot(1+\tan^2\frac{\alpha}{2})/2}}{\varepsilon\sqrt{d}}\right) \tag{4.4}$$

- ***(For bounding collision probability of close points.)*** *If $0 \leq \alpha < \alpha_0$ for some constant $0 < \alpha_0 < \pi/2$, then*

$$\Pr_{X,Y\sim\mathcal{N}(0,1)}\left[X \geq \varepsilon\sqrt{d}\cos\alpha \wedge X - \sin\alpha \cdot Y \geq \varepsilon\sqrt{d}\right] = \Omega\left(\frac{e^{-\varepsilon^2 d\cdot(1+\tan^2\frac{\alpha_0}{2})/2}}{\varepsilon^2 d\cdot\tan\frac{\alpha_0}{2}}\right). \tag{4.5}$$

Using Lemma 4.6 we can bound the collision probability in Eq. (4.1) as follows.

Lemma 4.7. *[AINR14] 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

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

Using Lemma 4.7 for the angles that correspond to distances 1 and c, one gets a value of ρ as claimed.

□

Now all we have to do is extend this construction to work with strips of $\mathcal{O}(1)$ width. This can be done using the following two ideas.

Normalize points before hashing them. This allows us to use the construction above as it is for hashing. But we need to ask: does it mess up the collision probability by a lot? The answer is no (for the distance regimes we care about). This can be easily seen by considering the following identity for any two vectors u, v,

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

Using this, one can check that for $u, v \in \{x \in \mathbb{R}^d \mid \|x\| \in [\eta c - 1, \eta c + 1]\}$:

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

Run the loop for some fixed (say $\exp(o(d))$) times. This works because of an ε-net argument. Consider approximating the volume of continuous spherical strip with a discrete subset. Identify spheres of very small radius ε with it's center. How many such points should be present in the discrete set to cover the entire volume? The answer is the ratio of the volume of the entire shell to the volume of an ε-ball. This discrete set is called an ε-net. It turns out that running the loop on line 2 $\exp(o(d))$ times implies that the probability that the buckets cover the shell is at least $1 - \exp(-d)$.
This concludes the proof.

□

4.2 Two Level Hashing Algorithm

Now using the tools we have developed we are ready to give an two level hashing algorithm (or data dependent hashing algorithm) for $(c, 1)$-ANN. At first let us look at the **overview of the algorithm**.
Say we have given n many input points fro $\mathbb{R}^d$ and S be the set of inputs.

- In the very first step we assume that we can reduce the dimension of the space $\mathcal{O}_c(\log d)$, using Theorem 2.2. For this we pay negligible amount of cost in the error and problem will be reduced to $(c-1, 1)$-ANN. But the advantage is, we can now perform all the arithmetic operations in $\mathcal{O}_c(\log n)$ time and all values of order $\exp(o(d))$ is now $n^{o_c(1)}$.
- Without loss of generality, we assume we are solving $(c, 1)$-ANN in $\mathcal{O}_c(\log n)$ dimension. In the first step we sample $h \sim \mathcal{H}^{\otimes k}$, for some suitable k, where $\mathcal{H}$ is the family defined in Theorem 4.2. To get improved bound on ρ, we introduced a new parameter τ and get the following relation between distance and collision probability.

Distance	1	c	τc
Collision probability	$n^{-1/(\tau c)^2}$	n^{-1/τ^2}	n^{-1}

- We get hash buckets $B_1, \ldots, B_m$ and with high probability, diameter of each B_i is $\leq \tau c$. Imposing Theorem 2.3 we can assume each B_i is a ball of radius $\tau c/\sqrt{2}$.
- Focus on some fixed B_i. Say, u_i is it's center. We shift origin to u_i to make the center of B_i to be $\mathbf{0}$. Say $s_i \in B_i$ is the nearest point to u_i. Now we can break the ball B_i in a inner sphere P_i' of diameter $\leq c-1$ and outer spherical shells of radius $\mathcal{O}(c)$ and width $\mathcal{O}(1)$. Say they are $P_{i0}, \ldots, P_{iT}$, for $T = \lceil \frac{\tau c}{\sqrt{2}} - \frac{c}{2} \rceil + 1$.
- Sample $h'_{i0}, \ldots, h'_{iT} \sim \mathcal{H}_G^{k'}$ for each of the spherical shells independently. In this case we get relation between distance and collision probabilities as follows,

Distance	1	c
Collision probability	$n^{-(1-\Omega_\tau(1))(1-1/\tau^2)/c^2}$	n^{-1+1/τ^2}

- As we sampled two types of hash function independently, we get,

Distance	1	c
Collision probability	$n^{-(1-\Omega_\tau(1))/c^2}$	n^{-1}

- So, $\rho \approx \frac{1-\Omega_\tau(1)}{c^2}$. Putting $\tau = \sqrt{2}$, we get the above bound that $\rho \leq (7/8c^2)$.

In next page we state the formal algorithm. Finally set k to be the smallest positive integer so that $\frac{(U(\tau c-1))^k}{L^k} \leq \frac{1}{n}$. And choose k'_j to be smallest positive integer so that for every u, v with $\|u\|, \|v\| \in [c/2 + j - 1, c/2 + j + 1]$ and $\|u - v\| \geq c$,

$$U(c)^k \Pr_{h \sim \mathcal{H}_G^{\otimes k'_j}}[h(u) = h(v)] \leq \frac{1}{3n}$$

Say given $p \in S$ some data point and q be some query point.

- $\mathcal{A}_1$: We iterate through p fo line 27 of Algorithm 2.
- $\mathcal{A}_2$: $B_{h(q)} \neq \phi$ and $\|q - s_{h(q)}\| \leq c$.

Consider the following two lemmas.

Lemma 4.8. *[AINR14] If $\|p - q\| \geq c$ then $\Pr[\mathcal{A}_1] \leq n^{-1}$.*

Lemma 4.9. *[AINR14] When* $\|p-q\| \leq 1$, $\Pr[\mathcal{A}_1 \vee \mathcal{A}_2] \leq n^{-\left(1-\frac{1}{2\tau^2}+\frac{1}{2\tau^4}\right)\frac{1}{c^2}+\mathcal{O}_\tau(\frac{1}{c^3})+o_{c,\tau}(1)}$.

Intuitively the above two lemmas say that,

- When $\|p-q\| > c$, with high probability, we shall not find p while iterating in hash buckets inside any of the outer shells.
- When $\|p-q\| \leq 1$, we shall hit p sometimes during the run of the algorithm with high probability.

Together with the lemmas and putting $\tau = \sqrt{2}$ we achieve the desired upper bound on τ.

Algorithm 2 Two Level Hashing

Pre-processing (S, τ, T, k, k'_j)

Sample $h \sim \mathcal{H}^{\otimes k}$ where $\mathcal{H}$ is from Theorem 4.2.
Get $B_1, \ldots, B_m$ where $B_i = \{x : h(x) = i\}$.
for $i = 1, \ldots, m$ **do**
  while $\exists\, u, v \in B_i$ s.t. $\|u - v\| > \tau c$ **do**
    $B_i \leftarrow B_i \setminus \{u, v\}$
  end while
  if $B_i \neq \phi$ **then**
    $u_i \leftarrow$ the center of smallest enclosing ball of B_i.
    $s_i \in B_i$ be the nearest point to u_i.
    for $j = 0, \ldots, T$ **do**
      $P_{ij} := \{p - u_i : c/2 + j - 1 \leq \|p - u_i\| \leq c/2 - j + 1\}$
      Sample $h'_{ij} \sim \mathcal{H}_G^{\otimes k'_j}$ for $\eta = 1/2 + j/c$ from Theorem 4.4.
      $B'_{ijr} := \{x \in P_{ij} : h'_{ij}(x) = r\}$
    end for
  end if
end for

Query(q)

$i \leftarrow h(q)$.
if $B_i = \phi$ **then**
  Return $\perp$
end if
if $\|q - s_i\| \leq c$ **then**
  Return s_i
end if
for $j = 0, \ldots, T$ **do**
  if $c/2 + j - 1 \leq \|q - u_i\| \leq c/2 - j + 1$ **then**
    $r \leftarrow h'_{ij}(q - u_i)$
    for $p \in B'_{ijr}$ **do**
      if $\|q - (p + u_i)\| \leq c$ **then**
        Return $p + u_i$
      end if
    end for
  end if
end for
Return $\perp$

5 Discussion and Future Work

Can we achieve even better parameters than what we have seen so far? Along this line, [BDGL16] introduced the idea of *locality sensitive filtering* using random product codes (RPC). The idea is to pick an efficiently list decodeable RPC over the surface of a sphere, and identify each hash bucket as the set of all vectors that decode to a particular code word. They show that this can achieve better ρ parameter. Furthermore, recently [KL21] show that this is optimal, when the data set is distributed on the surface of a sphere. Further, [AR15] showed a lower bound for data-dependent LSH. They showed that even if the hashing is data-dependent, it must hold that $\rho \geq \frac{1}{2c-1} - o(1)$.

On the side of conditional hardness for ANN, [Rub18] showed a SETH based lower bound for the problem. Specifically they show that unless the strong exponential time hypothesis (SETH) is false, $\forall \delta > 0, \exists \varepsilon > 0$ such that computing $(1+\varepsilon)$ approximation to bichromatic closest pair (the batch version of ANN) requires $\Omega\left(n^{2-\delta}\right)$ time. This implies a near linear time queries are essential for ANN with polynomial processing time.

In practice, recently the idea of hierarchical navigable small world graphs [MY20] has gained a lot of attention, and has proved superior than LSH based techniques. The idea here is to build increasingly dense proximity graphs with the data points as a set of vertices and perform random walks on them to converge to the approximately closest point. We believe that there are lots of cool theoretical ideas to be exploited in this framework.

References

[AI06] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *2006 47th annual IEEE symposium on foundations of computer science (FOCS'06)*, pages 459–468. IEEE, 2006.

[AINR14] Alexandr Andoni, Piotr Indyk, Huy L. Nguyen, and Ilya Razenshteyn. Beyond locality-sensitive hashing. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, page 1018–1028, USA, 2014. Society for Industrial and Applied Mathematics.

[AR15] Alexandr Andoni and Ilya Razenshteyn. Tight lower bounds for data-dependent locality-sensitive hashing, 2015.

[BDGL16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, page 10–24, USA, 2016. Society for Industrial and Applied Mathematics.

[IM98] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, page 604–613, New York, NY, USA, 1998. Association for Computing Machinery.

[JL84] William B. Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into hilbert space. *Contemporary mathematics*, 26:189–206, 1984.

[KL21] Elena Kirshanova and Thijs Laarhoven. Lower bounds on lattice sieving and information set decoding. In *Advances in Cryptology – CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part II*, page 791–820, Berlin, Heidelberg, 2021. Springer-Verlag.

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

[MNP08] Rajeev Motwani, Assaf Naor, and Rina Panigrahy. Lower bounds on locality sensitive hashing. *SIAM Journal on Discrete Mathematics*, 21(4):930–935, 2008.

[MY20] Yu A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(4):824–836, April 2020.

[OWZ14] Ryan O'Donnell, Yi Wu, and Yuan Zhou. Optimal lower bounds for locality-sensitive hashing (except when q is tiny). *ACM Trans. Comput. Theory*, 6(1), March 2014.

[Rub18] Aviad Rubinstein. Hardness of approximate nearest neighbor search. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 1260–1268, New York, NY, USA, 2018. Association for Computing Machinery.

A Proof Of the Hashing Lemma

Lemma A.1 (Hashing Lemma)**.** *Let H be an (r, R, p_1, p_2)-LSH family on the Hamming cube $(\{0,1\}^d, \|\cdot\|_1)$, fix a point $x \in \{0,1\}^d$ and let $R < d/2$, then*

$$\mathbb{E}|H^{-1}(H(x))| \leq 2^d(p_2 + e^{-\frac{1}{d}(\frac{d}{2}-R)^2}),$$

Proof. We can divide all points in the hash bucket $H^{-1}(H(x))$ into two parts: (1) $\{u : \|u-x\|_1 \le R\}$ and (2) $\{u : \|u-x\|_1 > R\}$. For each point x in part (1), we have $\Pr_H[H(x) = H(u)] \le 1$; while for each point x in part (2), we have $\Pr_H[H(x) = H(u)] \le p_2$ by definition of LSH. Thus we simply write

$$\begin{aligned}\mathbb{E}|H^{-1}(H(x))| &= \sum_{u\in\{0,1\}^d} \Pr_H[H(u) = H(x)] \\ &\le \left|\left\{u \in \{0,1\}^d : \|u-x\|_1 \le R\right\}\right| + p_2 \cdot \left|\left\{u \in \{0,1\}^d : \|u-x\|_1 > R\right\}\right| \\ &= \sum_{k=0}^{\lfloor R\rfloor} \binom{d}{k} + p_2 \cdot \sum_{k=\lfloor R\rfloor+1}^{d} \binom{d}{k}.\end{aligned}$$

Assumes that $R < \frac{d}{2}$, the inequality in the Hashing Lemma follows from the estimation of binomial coefficients,

$$\sum_{k\le \frac{d}{2}-a} \binom{d}{k} \le 2^d \cdot e^{-\frac{a^2}{d}}.$$

□

B Proof of the Random Walk Lemma

Lemma B.1 (Random Walk Lemma)**.** *Let r be an odd integer. Given $\emptyset \neq B \subseteq \{0,1\}^d$, consider the random variable $Q_B \in \{0,1\}^d$ defined as follows:*
Choose a point $z \in B$ uniformly at random, and perform r-steps of the standard random walk on the Hamming cube starting from z.(Choose y uniformly from the set of all strings which have hamming distance 1 from z) The point thus obtained will be denoted Q_B. Then,

$$\Pr\left[Q_B \in B\right] \le \left(\frac{|B|}{2^d}\right)^{\frac{e^{2r/d}-1}{e^{2r/d}+1}}.$$

We will use the following facts and definitions from the Fourier analysis on the Hamming cube to prove the above lemma.

Definition B.2 (Fourier Basis)**.** *Given $S \subset \{0,1\}^d$ we define the following function $W_S : \{0,1\}^d \to \{-1,1\}$ as*

$$W_S(u) = (-1)^{\sum_{j\in S} u_j}$$

Definition B.3 (Inner Product)**.** *For any $f, g : \{0,1\}^d \to \mathbb{R}$ we will define an inner product between them as*

$$\langle f, g\rangle = \frac{1}{2^d} \sum_{u\in\{0,1\}^d} f(u)g(u)$$

- We define, $\hat{f}(S) := \langle f, W_S\rangle$
- We can see that $\langle W_{S_1}, W_{S_2}\rangle = 0$ for $S_1 \neq S_2$.
- Using this fact, we get that W_S form an orthogonal basis hence we get the following decomposition

  $$f = \sum_{S\subset[d]} \hat{f}(S)W_S,$$

  which implies

  $$\langle f, g\rangle = \sum_{S\subseteq[d]} \hat{f}(S)\hat{g}(S).$$

We will asumme the following fact for every $f : \{0,1\}^d \to \mathbb{R}$, which follows from the Bonami-Beckner inequality

$$\sum_{S\subseteq[d]} \varepsilon^{2|S|}\widehat{f}(S)^2 \leq \left(\frac{1}{2^d}\sum_{u\in\{0,1\}^d} f(u)^{1+\varepsilon^2}\right)^{\frac{2}{1+\varepsilon^2}}.$$

Specializing to the indicator of $B \subseteq \{0,1\}^d$ we get that

Definition B.4 (Bonami-Becker Inequality)**.**

$$\sum_{S\subseteq\ [d]} \varepsilon^{2|S|}\widehat{\mathbf{1}_B}(S)^2 \leq \left(\frac{|B|}{2^d}\right)^{\frac{2}{1+\varepsilon^2}}.$$

(Proof of Random Walk Lemma). For this treat the functions from $f : \{0,1\}^d \to \mathbb{R}$ as a vector in $\mathbb{R}^{2^d}$ where $f_i = f(\text{binary}(i))$. Let P be the transition matrix of the standard random walk on $\{0,1\}^d$, i.e. $P_{uv} = 1/d$ if u and v differ in exactly one coordinate, $P_{uv} = 0$ otherwise. By a direct computation we have that for every $S \subseteq \{1,\ldots,d\}$,

$$PW_S = \left(1 - \frac{2|S|}{d}\right) W_S.$$

Since, W_S is an eigenvector of P with eigenvalue $1 - \frac{2|S|}{d}$. The probability that the random walk starting form a random point in B ends up in B after r steps equals.

$$\begin{aligned}
\Pr\left[Q_B \in B\right] &= \frac{1}{|B|}\sum_{a,b\in B}(P^r)_{ab} \\
&= \frac{2^d}{|B|}\left\langle P^r\mathbf{1}_B, \mathbf{1}_B\right\rangle \\
&= \frac{2^d}{|B|}\sum_{S\subseteq\{1,\ldots,d\}} \widehat{\mathbf{1}_B}(S)^2\left(1 - \frac{2|S|}{d}\right)^r \\
&\leq \frac{2^d}{|B|}\sum_{\substack{S\subseteq\{1,\ldots,d\}\\ |S|\leq d/2}} \widehat{\mathbf{1}_B}(S)^2\left(1 - \frac{2|S|}{d}\right)^r && \text{(Since r is odd drop the negative terms)}
\end{aligned}$$

Now for $|S| \leq d/2$ we can apply the fact that $e^{-x} \geq (1-x)$ to get $e^{-\frac{2|S|r}{d}} \geq (1 - \frac{2|S|}{d})^r$, since both quantities are positive, for $|S| > d/2$ the quantity $(1 - \frac{2|S|}{d})$ is negative hence its odd power would be negative and hence it is less than $e^{-\frac{2|S|r}{d}}$

$$\begin{aligned}
\Pr\left[Q_B \in B\right] &\leq \frac{2^d}{|B|}\sum_{S\subseteq\{1,\ldots,d\}} \widehat{\mathbf{1}_B}(S)^2 \cdot e^{-2r|S|/d} \\
&\leq \frac{2^d}{|B|}\cdot\left(\frac{|B|}{2^d}\right)^{\frac{2}{1+e^{-2r/d}}} && \text{(Bonami-Becker Inequality)} \\
&= \left(\frac{|B|}{2^d}\right)^{\frac{1-e^{-2r/d}}{1+e^{-2r/d}}}.
\end{aligned}$$

□