Tight Lower Bounds for Data-Dependent Locality-Sensitive Hashing

Alexandr Andoni[*1] **and Ilya Razenshteyn**[2]

1 Columbia University, New York, USA
2 MIT CSAIL, Cambridge, USA

Abstract

We prove a tight lower bound for the exponent ρ for data-dependent Locality-Sensitive Hashing schemes, recently used to design efficient solutions for the c-approximate nearest neighbor search. In particular, our lower bound matches the bound of $\rho \leq \frac{1}{2c-1} + o(1)$ for the ℓ_1 space, obtained via the recent algorithm from [Andoni-Razenshteyn, STOC'15].

In recent years it emerged that data-dependent hashing is strictly superior to the classical Locality-Sensitive Hashing, when the hash function is data-*independent*. In the latter setting, the best exponent has been already known: for the ℓ_1 space, the tight bound is $\rho = 1/c$, with the upper bound from [Indyk-Motwani, STOC'98] and the matching lower bound from [O'Donnell-Wu-Zhou, ITCS'11].

We prove that, even if the hashing is data-dependent, it must hold that $\rho \geq \frac{1}{2c-1} - o(1)$. To prove the result, we need to formalize the exact notion of data-dependent hashing that also captures the complexity of the hash functions (in addition to their collision properties). Without restricting such complexity, we would allow for obviously infeasible solutions such as the Voronoi diagram of a dataset. To preclude such solutions, we require our hash functions to be succinct. This condition is satisfied by all the known algorithmic results.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Similarity search, high-dimensional geometry, LSH, data structures, lower bounds

Digital Object Identifier 10.4230/LIPIcs.SoCG.2016.9

1 Introduction

We study lower bounds for the high-dimensional nearest neighbor search problem, which is a problem of major importance in several areas, such as databases, data mining, information retrieval, computer vision, computational geometry, signal processing, etc. This problem suffers from the "curse of dimensionality" phenomenon: either space or query time are exponential in the dimension d. To escape this curse, researchers proposed *approximation* algorithms for the problem. In the (c, r)-approximate near neighbor problem, the data structure may return any data point whose distance from the query is at most cr, for an approximation factor $c > 1$ (provided that there exists a data point within distance r from the query). Many approximation algorithms are known for this problem: e.g., see surveys [22, 3, 1, 24].

An influential algorithmic technique for the approximate near neighbor search (ANN) is the *Locality Sensitive Hashing* (LSH) [13, 12]. The main idea is to hash the points so that the

[*] Work done in part while the first author was at the Simons Institute for the Theory of Computing, Berkeley University.

probability of collision is much higher for points that are close to each other (at distance $\leq r$) than for those which are far apart (at distance $> cr$). 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 essentially $O(n^{1+\rho}/p_1)$ extra space and $O(dn^\rho/p_1)$ query time, where $\rho = \frac{\log(1/p_1)}{\log(1/p_2)}$ [12]. The value of the exponent ρ thus determines the "quality" of the LSH families used.

Consequently, a lot of work focused on understanding the best possible value ρ for LSH, including the sequence of upper bounds [13, 9, 2] and lower bounds [18, 19]. Overall, they established the precise bounds for the best value of ρ: for ℓ_1 the tight bound is $\rho = \frac{1}{c} \pm o(1)$. In general, for ℓ_p, where $1 \leq p \leq 2$, the tight bound is $\rho = \frac{1}{c^p} \pm o(1)$.

Surprisingly, it turns out there exist *more efficient* ANN data structures, which step outside the LSH framework. Specifically, [5, 6] design algorithms using the concept of *data-dependent hashing*, which is a randomized hash family that itself adapts to the actual given dataset. In particular, the result of [6] obtains an exponent $\rho = \frac{1}{2c^p-1} + o(1)$ for the ℓ_p space, thus improving upon the best possible LSH exponent essentially by a factor of 2 for both ℓ_1 and ℓ_2 spaces.

Our result. Here we prove that the exponent $\rho = \frac{1}{2c^p-1}$ from [6] is essentially optimal even for data-dependent hashing, and cannot be improved upon. Stating the precise theorem requires introducing the precise model for the lower bound, which we accomplish below. For now, we state our main theorem informally:

▶ **Theorem 1** (Main, informal). *Any data-dependent hashing scheme for ℓ_1 that achieves probabilities p_1 and p_2 must satisfy*

$$\rho = \frac{\log 1/p_1}{\log 1/p_2} \geq \frac{1}{2c-1} - o(1),$$

as long as the description complexity of the hash functions is sufficiently small.

An immediate consequence is that $\rho \geq \frac{1}{2c^p-1} - o(1)$ for all ℓ_p with $1 \leq p \leq 2$, using the embedding from [17].

1.1 Lower Bound Model

To state the precise theorem, we need to formally describe what is data-dependent hashing. First, we state the definition of (data-independent) LSH, as well as define LSH for a fixed dataset P.

▶ **Definition 2** (Locality-Sensitive Hashing). We say that a hash family $\mathcal{H}$ over $\{0,1\}^d$ is (r_1, r_2, p_1, p_2)-*sensitive*, if for every $u, v \in \{0,1\}^d$ one has:

- if $\|u - v\|_1 \leq r_1$, then $\Pr_{h\sim\mathcal{H}}[h(u) = h(v)] \geq p_1$;
- if $\|u - v\|_1 > r_2$, then $\Pr_{h\sim\mathcal{H}}[h(u) = h(v)] \leq p_2$.

We now refine the notion of data-independent LSH, where we require the distribution to work only for a particular dataset P.

▶ **Definition 3** (LSH for a dataset P). A hash family $\mathcal{H}$ over $\{0,1\}^d$ is said to be (r_1, r_2, p_1, p_2)-*sensitive for a dataset $P \subseteq \{0,1\}^d$*, if:

- for every $v \in \{0,1\}^d$ and every $u \in P$ with $\|u - v\|_1 \leq r_1$ one has

$$\Pr_{h\sim\mathcal{H}}[h(u) = h(v)] \geq p_1;$$

- $\Pr_{\substack{h\sim\mathcal{H}\\u,v\sim P}}[h(u)=h(v) \text{ and } \|u-v\|_1 > r_2] \le p_2$.

Note that the second definition is less stringent than the first one: in fact, if all the points in a dataset are at distance more than r_2 from each other, then an LSH family $\mathcal{H}$ is also LSH for P, but not necessarily vice versa! Furthermore, in the second definition, we require the second property to hold only *on average* (in contrast to *every* point as in the first definition). This aspect means that, while Definition 3 is certainly necessary for an ANN data structure, it is not obviously *sufficient*. Indeed, the algorithm from [6] requires proving additional properties of their partitioning scheme, and in particular analyzes *triples* of points. Since here we focus on lower bounds, this aspect will not be important.

We are now ready to introduce data-dependent hashing.

1.1.1 Data-dependent hashing

Intuitively, a data-dependent hashing scheme is one where we can pick the family $\mathcal{H}$ *as a function of* P, and thus be sensitive for P. An obvious solution would hence be to choose $\mathcal{H}$ to consist of a single hash function h which is just the Voronoi diagram of the dataset P: it will be $(r, cr, 1, 0)$-sensitive for P and hence $\rho = 0$. However this does not yield a good data strcture for ANN since *evaluating* such a hash function on a query point q is as hard as the original problem!

Hence, ideally, our lower bound model would require that the hash function is *computationally efficient* to evaluate. We do not know how to formulate such a condition which would not make the question as hard as circuit lower bounds or high cell-probe lower bounds, which would be well beyond the scope of this paper.

Instead, we introduce a condition on the hash family that can be roughly summarized as "hash functions from the family are succinct". For a precise definition and discussion see below. For now, let us point out that all the known algorithmic results satisfy this condition.

Finally, we are ready to state the main result formally.

▶ **Theorem 4** (Main theorem, full). *Fix the approximation $c > 1$ to be a constant. Suppose the dataset P lives in the Hamming space $\{0,1\}^d$, where the dataset size $n = |P|$ is such that $d = \omega(\log n)$ as n tends to infinity. There exist distance thresholds r and $(c - o(1))r$ with the following property.*

Suppose there exist T hash functions $\{h_i\}_{1\le i\le T}$ over $\{0,1\}^d$ such that for every n-point dataset P there exists a distribution $\mathcal{H}_P$ over h_i's such that the corresponding hash family is $(r, (c - o(1))r, p_1, p_2)$-sensitive for P, where $0 < p_1, p_2 < 0.99$. For any such data-dependent scheme it must either hold that $\rho = \frac{\log 1/p_1}{\log 1/p_2} \ge \frac{1}{2c-1} - o(1)$ or that $\frac{\log T}{p_1} \ge n^{1-o(1)}$.

1.1.2 Interpreting Theorem 4

Let us explain the conditions and the conclusions of Theorem 4 in more detail.

We start by interpreting the conclusions. As explained above, the bound $\rho = \frac{\log 1/p_1}{\log 1/p_2} \ge \frac{1}{2c-1} - o(1)$ directly implies the lower bound on the query time $n^{\frac{1}{2c-1}-o(1)}$ for any scheme that is based on data-dependent hashing.

The second bound $\frac{\log T}{p_1} \ge n^{1-o(1)}$ is a little bit more mysterious. Let us now explain what it means precisely. The quantity $\log T$ can be interpreted as the *description complexity* of a hash function sampled from the family. At the same time, if we use a family with collision probability p_1 for close points, we need at least $1/p_1$ hash tables to achieve constant probability of success. Since in each hash table we evaluate at least one hash function, the

quantity $\frac{\log T}{p_1}$ can be interpreted as the lower bound for the *total space occupied by hash functions we evaluate during each query.* In all known constructions of (data-independent or data-dependent) LSH families [13, 9, 2, 23, 5, 6] the *evaluation time* of a single hash function is comparable to the space it occupies (for discussion regarding why is it true for [6], see Appendix A), thus, under this assumption, we can not achieve query time $n^{1-\Omega(1)}$, unless $\rho \geq \frac{1}{2c-1} - o(1)$. On the other hand, we can achieve $\rho = 0$ by considering a data-dependent hash family that consists only of the Voronoi diagram of a dataset (trivially, $p_1 = 1$ and $p_2 = 0$ for this case), thus the conclusion $\frac{\log T}{p_1} \geq n^{1-o(1)}$ can not be omitted in general[1]. Note that the "Voronoi diagram family" is very slow to evaluate: to locate a point we need to solve an instance of exact Nearest Neighbor Search, that is unlikely to be possible to do in strongly sublinear time. Thus, this family satisfies the above assumption "evaluation time is comparable to the space".

We now turn to interpreting the conditions. We require that $d = \omega(\log n)$. We conjecture that this requirement[2] is necessary and for $d = O(\log n)$ there is an LSH family that gives a better value of ρ (the improvement, of course, would depend on the hidden constant in the expression $d = O(\log n)$). In fact, some evidence towards this conjecture is given in a recent paper [7], where an improved data structure for ANN for the case $d = O(\log n)$ is presented, albeit by stepping outside of the pure (data-dependent) LSH framework.

1.2 Techniques and Related Work

There are two components to our lower bound, i.e., Theorem 4.

The first component is a lower bound for *data-independent* LSH for a *random dataset.* We show that in this case, we must have $\rho \geq \frac{1}{2c-1} - o(1)$. This is in contrast to the lower bound of [19], who achieve a higher lower bound but for the case when the (far) points are correlated. Our lower bound is closer in spirit to [18], who also consider the case when the far points are random uncorrelated. In fact, this component is a strengthening of the lower bound from [18].

We mention that, in [10], Dubiner has also considered the setting of a random dataset for a related problem – finding the closest pair in a given dataset P. Dubiner sets up a certain related "bucketing" model, in which he conjectures the lower bound, which would imply a $\rho \geq \frac{1}{2c-1}$ lower bound for data-independent LSH for a random set. Dubiner verifies the conjecture computationally and claims it is proved in a different manuscript.[3]

We also point out that, for the ℓ_2 case, the optimal data-independent lower bound $\rho \geq \frac{1}{2c^2-1} - o(1)$ follows from a recent work [4]. In fact, it shows almost exact trade-off between p_1 and p_2 (not only the lower bound on $\rho = \frac{\log(1/p_1)}{\log(1/p_2)}$). Unfortunately, the techniques there are really tailored to the Euclidean case (in particular, a powerful isoperimetric inequality of Feige and Schechtman is used [11]) and it is unclear how to extend it to ℓ_1, as well as, more generally, to ℓ_p for $1 \leq p < 2$.

Our second component focuses on the data-dependent aspect of the lower bound. In particular, we prove that if there exists a data-dependent hashing scheme for a random dataset with a better ρ, then in fact there is also a data-independent such hashing scheme.

---

[1] For the Voronoi diagram, $\log T \geq n$, since to specify it, one needs at least n bits.

[2] When ANN for the general dimension d is being solved, one usually first performs some form of the dimensionality reduction [14, 16, 8]. Since at this stage we do not want distances to be distorted by a factor more than $1 + o(1)$, the target dimension is precisely $\omega(\log n)$. So, the assumption $d = \omega(\log n)$ in Theorem 4 in some sense captures a *truly high-dimensional case.*

[3] This manuscript does not appear to be available at the moment of writing of the present paper.

To accomplish this, we consider the "empirical" average p_1 and p_2 for a random dataset, and prove it is close to the average p_1 and p_2, for which we can deduce a lower bound from the first component.

In terms of related work, we also must mention the papers of [20, 21], who prove cell-probe lower bounds for the same problem of ANN. In particular, their work utilizes the lower bound of [18] as well. Their results are however incomparable to our results: while their results are unconditional, our model allows us to prove a much higher lower bound, in particular matching the best algorithms.

2 Data-independent Lower Bound

In this section we prove the first component of Theorem 4. Overall we show a lower bound of $\rho \geq \frac{1}{2c-1} + o(1)$ for data-independent hash families for random datasets. Our proof is a strengthening and simplification of [18]. The final statement appears as Corollary 7. In the second component, we will use a somewhat stronger statement, Lemma 6).

For $u \in \{0,1\}^d$ and $0 \leq \alpha \leq 1/2$ define a random variable $N_\alpha(u)$ distributed over $\{0,1\}^d$ as follows: we set every bit of $N_\alpha(u)$ to the corresponding bit of u with probability $1-\alpha$ and with the remaining probability α we flip the bit.

The following lemma is proved via a combination of the averaging argument from [18] and an estimate using Hypercontractivity on the Boolean cube that can be found, e.g., in the proof of Lemma 3.6 of the arXiv version of [15].

▶ **Lemma 5.** *For every hash function $h\colon \{0,1\}^d \to \mathbb{Z}$ and every $0 \leq \alpha \leq 1/2$ one has:*

$$\Pr_{\substack{u\sim\{0,1\}^d \\ v\sim N_\alpha(u)}}[h(u)=h(v)] \leq \Pr_{u,v\sim\{0,1\}^d}[h(u)=h(v)]^{\frac{\alpha}{1-\alpha}}.$$

Proof. Let $A \subseteq \{0,1\}^d$ be a subset of the hypercube. Then, from the proof of Lemma 3.6 from the arXiv version of [15], we have

$$\Pr_{\substack{u\sim\{0,1\}^d \\ v\sim N_\alpha(u)}}[v \in A \mid u \in A] \leq \left(\frac{|A|}{2^d}\right)^{\frac{\alpha}{1-\alpha}}. \tag{1}$$

Low let us finish the proof using (1). For every $h\colon \{0,1\}^d \to \mathbb{Z}$ one has:

$$\begin{aligned}\Pr_{\substack{u\sim\{0,1\}^d \\ v\sim N_\alpha(u)}}[h(u)=h(v)] &= \sum_{k\in\mathbb{Z}} \Pr_{\substack{u\sim\{0,1\}^d \\ v\sim N_\alpha(u)}}[v \in h^{-1}(k) \mid u \in h^{-1}(k)] \cdot \left(\frac{|h^{-1}(k)|}{2^d}\right) \\ \leq \sum_{k\in\mathbb{Z}} \left(\frac{|h^{-1}(k)|}{2^d}\right)^{\frac{\alpha}{1-\alpha}} \cdot \left(\frac{|h^{-1}(k)|}{2^d}\right) &= \mathop{\mathrm{E}}_{u\sim\{0,1\}^d}\left[\left(\frac{|h^{-1}(h(u))|}{2^d}\right)^{\frac{\alpha}{1-\alpha}}\right] \\ &\leq \mathop{\mathrm{E}}_{u\sim\{0,1\}^d}\left[\frac{|h^{-1}(h(u))|}{2^d}\right]^{\frac{\alpha}{1-\alpha}} = \Pr_{u,v\sim\{0,1\}^d}[h(u)=h(v)]^{\frac{\alpha}{1-\alpha}},\end{aligned}$$

where the second step follows from (1), and the fourth step follows from the Jensen's inequality. ◀

We are now ready to prove the main lemma of this section, which will be used in the later section on data-dependent hashing. We need to introduce two more definitions. We

define "average p_1" as:

$$\zeta(c,d,\alpha,h) := \Pr_{\substack{u\sim\{0,1\}^d \\ v\sim N_\alpha(u)}}\left[h(u)=h(v)\middle|\|u-v\|_1 \le \frac{d}{2c}\right]$$

for $c>1$, positive integer d, $\alpha>0$ and a hash function $h\colon \{0,1\}^d \to \mathbb{Z}$. Similarly, we define the "average p_2" as

$$\eta(d,\beta,h) := \Pr_{u,v\sim\{0,1\}^d}\left[h(u)=h(v), \|u-v\|_1 > \left(\frac{1}{2}-\beta\right)\cdot d\right]$$

for positive integer d, $\beta>0$ and a hash function $h\colon \{0,1\}^d \to \mathbb{Z}$.

▶ **Lemma 6.** *Let $c>1$ be a fixed constant. Suppose that $\gamma=\gamma(d)>0$ is such that $\gamma=o(1)$ as $d\to\infty$. Then, there exist $\alpha=\alpha(d)$, $\beta=\beta(d)$ and $\rho=\rho(d)$ such that:*

$$\alpha = \frac{1}{2c} - o_{c,\gamma}(1),$$
$$\beta = o_\gamma(1),$$
$$\rho = \frac{1}{2c-1} - o_{c,\gamma}(1)$$

as $d\to\infty$ such that, for every d and every hash function $h\colon \{0,1\}^d\to\mathbb{Z}$, one has

$$\zeta(c,d,\alpha,h) \le \eta(d,\beta,h)^\rho + 2^{-\gamma\cdot d}.$$

Proof. We can choose $\alpha = \frac{1}{2c} - o_{c,\gamma}(1)$ such that

$$\Pr\left[\|N_\alpha(u)-u\|_1 \ge \frac{d}{2c}\right] < \frac{2^{-\gamma\cdot d}}{2}. \tag{2}$$

We can choose $\beta=o_\gamma(1)$ so that

$$\Pr_{u,v\sim\{0,1\}^d}\left[\|u-v\| < \left(\frac{1}{2}-\beta\right)\cdot d\right] < \frac{2^{-\gamma\cdot d}}{2}. \tag{3}$$

(Both follow from the standard Chernoff-type bounds.)

Now we apply Lemma 5 and get

$$\Pr_{\substack{u\sim\{0,1\}^d \\ v\sim N_\alpha(u)}}[h(u)=h(v)] \le \Pr_{u,v\sim\{0,1\}^d}[h(u)=h(v)]^\rho, \tag{4}$$

where

$$\rho = \frac{\alpha}{1-\alpha} = \frac{1}{2c-1} - o_{c,\gamma}(1).$$

Finally, we combine (4), (2) and (3), and get the desired inequality. ◀

The following corollary shows how the above lemma implies a lower bound on data-independent LSH.

▶ **Corollary 7.** *For every $c>1$ and $\gamma=\gamma(d)>0$ such that $\gamma=o(1)$ there exists $\beta=\beta(d)>0$ with $\beta=o_\gamma(1)$ such that if $\mathcal{H}$ is a data-independent $(d/(2c),(1/2-\beta)d,p_1,p_2)$-sensitive family, then*

$$p_1 \le p_2^{\frac{1}{2c-1}-o_{c,\gamma}(1)} + 2^{-\gamma\cdot d}.$$

Proof. We observe that for every $\alpha, \beta > 0$:

- $p_1 \leq \mathop{\mathrm{E}}_{h\sim\mathcal{H}}[\zeta(c,d,\alpha,h)]$;
- $p_2 \geq \mathop{\mathrm{E}}_{h\sim\mathcal{H}}[\eta(d,\beta,h)]$.

Now we apply Lemma 6 together with the following application of Jensen's inequality:

$$\mathop{\mathrm{E}}_{h\sim\mathcal{H}}[\eta(d,\beta,h)^\rho] \leq \mathop{\mathrm{E}}_{h\sim\mathcal{H}}[\eta(d,\beta,h)]^\rho,$$

since $0 < \rho \leq 1$. ◀

3 Data-Dependent Hashing

We now prove the second component of the main Theorem 4 proof. In particular we show that a very good data-dependent hashing scheme would refute Lemma 6 from the previous section.

3.1 Empirical Probabilities of Collision

For a particular dataset P, we will be interested in *empirical* probabilities p_1, p_2 – i.e., the equivalents of ζ, η for a given set P – defined as follows. Let $0 < \delta(d) < 1/3$ be some function. Let P be a random set of points from $\{0,1\}^d$ of size $2^{\delta(d)\cdot d}$. The *empirical* versions of ζ and η with respect to P are:

$$\widehat{\zeta}(c,d,\alpha,h,P) := \Pr_{\substack{u\sim P\\ v\sim N_\alpha(u)}}\left[h(u)=h(v)\middle|\|u-v\|_1 \leq \frac{d}{2c}\right]$$

$$\widehat{\eta}(d,\beta,h,P) := \Pr_{u,v\sim P}\left[h(u)=h(v), \|u-v\|_1 > \left(\frac{1}{2}-\beta\right)\cdot d\right].$$

We now want to prove that, for a random dataset P, the empirical ζ, η are close to the true averages. For this we will need the following auxiliary lemma.

▶ **Lemma 8.** *Let M be an $n \times n$ symmetric matrix with entries from $[0;1]$ and average ε. Let M' be a principal $n^\delta \times n^\delta$ submatrix of M sampled uniformly with replacement. Then, for every $\theta > 0$, the probability that the maximum of the average over M' and θ does not lie in $[1/2;2]\cdot\max\{\varepsilon,\theta\}$ is at most $n^\delta \cdot 2^{-\Omega(\theta n^\delta)}$.*

Proof. We need the following version of Bernstein's inequality.

▶ **Lemma 9.** *Suppose that $X_1,\ldots,X_n$ are i.i.d. random variables that are distributed over $[0;1]$. Suppose that $\mathrm{E}[X_i] = \varepsilon$. Then, for every $0 < \theta < 1$, one has*

$$\Pr\left[\max\left\{\frac{1}{n}\sum_{i=1}^n X_i, \theta\right\} \in \left[\frac{1}{2};2\right]\cdot\max\{\varepsilon,\theta\}\right] \geq 1 - 2^{-\Omega(\theta n)}.$$

We just apply Lemma 9 and take union bound over the rows of M'. ◀

The following two lemmas are immediate corollaries of Lemma 9 and Lemma 8, respectively.

▶ **Lemma 10.** *For every $c > 1$, $\alpha > 0$, positive integer d, $\theta > 0$ and a hash function $h\colon \{0,1\}^d \to \mathbb{Z}$, one has*

$$\Pr_P\left[\max\{\widehat{\zeta}(c,d,\alpha,h,P),\theta\} \in [1/2;2]\cdot\max\{\zeta(c,d,\alpha,h),\theta\}\right] \geq 1 - 2^{-\Omega\left(\theta\cdot 2^{\delta(d)\cdot d}\right)}.$$

▶ **Lemma 11.** *For every $\beta > 0$, positive integer d, $\theta > 0$ and a hash function $h\colon \{0,1\}^d \to \mathbb{Z}$, one has*

$$\Pr_P\left[\max\{\widehat{\eta}(d,\beta,h,P),\theta\} \in [1/2;2]\cdot\max\{\eta(d,\beta,h),\theta\}\right] \geq 1 - 2^{\delta(d)\cdot d}\cdot 2^{-\Omega\left(\theta\cdot 2^{\delta(d)\cdot d}\right)}.$$

3.2 Proof of Theorem 4

We are finally ready to complete the proof of the main result, Theorem 4.

Let us first assume that $p_1 = o(1)$ and then show how to handle the general case. Suppose that $n = |P| = 2^{\delta\cdot d}$. By the assumption of Theorem 4, $\delta = o(1)$. Let $\{h_1, h_2, \ldots, h_T\}$ be a set of hash functions. We can assume that $\frac{\log T}{p_1} \leq n^{1-\Omega(1)}$, since otherwise we are done. Let us fix $\gamma = \gamma(d) > 0$ such that $\gamma = o(1)$ and $2^{-\gamma d} \ll p_1^{\omega(1)}$. We can do this, since if $p_1 = 2^{-\Omega(d)}$, then $1/p_1 = n^{\omega(1)}$, and the desired statement is true. Then, by Lemma 6, there is $\alpha = \alpha(d)$ and $\beta = \beta(d) = o_\gamma(1)$ such that for every $1 \leq i \leq T$

$$\zeta(c,d,\alpha,h_i) \leq \eta(d,\beta,h_i)^{\frac{1}{2c-1}-o_{c,\gamma}(1)} + 2^{-\gamma d}. \tag{5}$$

Let us choose $\theta > 0$ such that

$$T\cdot 2^{\delta\cdot d} \ll 2^{\theta\cdot 2^{\delta\cdot d}}$$

and $\theta \ll p_1$. We can do it since, by the above assumption, $\frac{\log T}{p_1} \leq n^{1-\Omega(1)} = 2^{\delta\cdot d(1-\Omega(1))}$.

Then, from Lemma 10 and Lemma 11 we get that, with high probability over the choice of P, one has for every $1 \leq i \leq T$:

- $\max\{\widehat{\zeta}(c,d,\alpha,h_i),\theta\} \in [1/2;2]\cdot\max\{\zeta(c,d,\alpha,h_i),\theta\}$;
- $\max\{\widehat{\eta}(d,\beta,h_i),\theta\} \in [1/2;2]\cdot\max\{\eta(d,\beta,h_i),\theta\}$.

Suppose these conditions hold and assume there exist a distribution $\mathcal{D}$ over $[T]$ such that the corresponding hash family is $(d/2c, (1/2-\beta(d))d, p_1, p_2)$-sensitive for P. Then,

$$\begin{aligned} p_1 \leq \mathop{\mathrm{E}}_{i\sim\mathcal{D}}[\widehat{\zeta}(c,d,\alpha,h_i)] \leq \mathop{\mathrm{E}}_{i\sim\mathcal{D}}[\max\{\widehat{\zeta}(c,d,\alpha,h_i),\theta\}] &\leq 2\cdot\mathop{\mathrm{E}}_{i\sim\mathcal{D}}[\max\{\zeta(c,d,\alpha,h_i),\theta\}] \\ &\leq 2\cdot\mathop{\mathrm{E}}_{i\sim\mathcal{D}}[\zeta(c,d,\alpha,h_i)] + \theta. \end{aligned} \tag{6}$$

Similarly,

$$\begin{aligned} p_2 \geq \mathop{\mathrm{E}}_{i\sim\mathcal{D}}[\widehat{\eta}(d,\beta,h_i)] \geq \mathop{\mathrm{E}}_{i\sim\mathcal{D}}[\max\{\widehat{\eta}(d,\beta,h_i),\theta\}] - \theta &\geq \frac{1}{2}\cdot\mathop{\mathrm{E}}_{i\sim\mathcal{D}}[\max\{\eta(d,\beta,h_i),\theta\}] - \theta \\ &\geq \frac{1}{2}\cdot\mathop{\mathrm{E}}_{i\sim\mathcal{D}}[\eta(d,\beta,h_i)] - \theta. \end{aligned} \tag{7}$$

Averaging (5) and applying Jensen's inequality, we have

$$\mathop{\mathrm{E}}_{i\sim\mathcal{D}}[\zeta(c,d,\alpha,h_i)] \leq \mathop{\mathrm{E}}_{i\sim\mathcal{D}}[\eta(d,\beta,h_i)]^{\frac{1}{2c-1}-o(1)} + 2^{-\gamma(d)\cdot d}. \tag{8}$$

Thus, substituting (6) and (7) into (8)

$$\frac{p_1-\theta}{2} \leq (2(p_2+\theta))^{\frac{1}{2c-1}-o(1)} + 2^{-\gamma d},$$

which proves the theorem, since $\theta \ll p_1$, $2^{-\gamma d} \ll p_1^{\omega(1)}$, and $p_1 = o(1)$.

Now let us deal with the case $p_1 = \Omega(1)$. This can be reduced to the case $p_1 = o(1)$ by choosing a slowly-growing super constant k and replacing the set of T functions with the set of T^k *tuples* of length k. This replaces p_1 and p_2 with p_1^k and p_2^k, respectively. In the same time, we choose k so that $T' = T^k$ still satisfy the hypothesis of the theorem. Then, we just apply the above proof.

References

1. Alexandr Andoni. *Nearest Neighbor Search: the Old, the New, and the Impossible*. PhD thesis, Massachusetts Institute of Technology, 2009.
2. 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'06)*, pages 459–468, 2006.
3. Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008.
4. Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. Practical and optimal LSH for angular distance. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS'15)*, 2015.
5. Alexandr Andoni, Piotr Indyk, Huy L. Nguyen, and Ilya Razenshteyn. Beyond locality-sensitive hashing. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'14)*, pages 1018–1028, 2014.
6. Alexandr Andoni and Ilya Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *Proceedings of the ACM Symposium on the Theory of Computing (STOC'15)*, 2015.
7. Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *Proceedings of the 27th ACM-SIAM Symposium on Discrete Algorithms (SODA'16)*, 2016.
8. Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures and Algorithms*, 22(1):60–65, 2003.
9. Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the 20th ACM Symposium on Computational Geometry (SoCG'04)*, pages 253–262, 2004.
10. Moshe Dubiner. Bucketing coding and information theory for the statistical highdimensional nearest-neighbor problem. *IEEE Transactions on Information Theory*, 56(8):4166–4179, 2010.
11. Uriel Feige and Gideon Schechtman. On the optimality of the random hyperplane rounding technique for MAX CUT. *Random Structures and Algorithms*, 20(3):403–440, 2002.
12. 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.
13. 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'98)*, pages 604–613, 1998.
14. 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.
15. Subhash Khot and Nisheeth K. Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative-type metrics into l_1. *J. ACM*, 62(1):8:1–8:39, 2015.
16. Eyal Kushilevitz, Rafail Ostrovky, and Yuval Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM Journal on Computing*, 30(2):457–474, 2000.
17. Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.
18. Rajeev Motwani, Assaf Naor, and Rina Panigrahy. Lower bounds on locality sensitive hashing. *SIAM Journal on Discrete Mathematics*, 21(4):930–935, 2007.

19 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'11)*, pages 275–283, 2011.

20 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'08)*, pages 414–423, 2008.

21 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'10)*, pages 805–814, 2010.

22 Hannan Samet. *Foundations of Multidimensional and Metric Data Structures*. Elsevier, 2006.

23 Kengo Terasawa and Yuzuru Tanaka. Spherical LSH for approximate nearest neighbor search on unit hypersphere. In *Algorithms and Data Structures, 10th International Workshop, WADS 2007, Halifax, Canada, August 15-17, 2007, Proceedings*, pages 27–38, 2007.

24 Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. *CoRR*, abs/1408.2927, 2014.

A Upper Bounds Are in the Model

In this section, we show how the data-dependent hash family from [6] fits into the model of the lower bound from Section 1.1.

Let us briefly recall the hash family construction from [6]. For simplicity, assume that all points and queries lie on a sphere of radius $R \gg cr$. First, consider a *data-independent* hash family from [5, 6], called *Spherical LSH*. It gives a good exponent $\rho \leq \frac{1}{2c^2-1} + o(1)$ for distance thresholds r vs $\sqrt{2}R$ (the latter corresponds to a typical distance between a pair of points from the sphere). The main challenge that arises is how to handle distance thresholds r vs cr, where the latter may be much smaller than $\sqrt{2}R$. Here comes the main insight of [6].

We would like to process the dataset so that the distance between a typical pair of *data points* is around $\sqrt{2}R$, so to apply Spherical LSH. To accomplish this, we remove all the clusters of radius $(\sqrt{2}-\varepsilon)R$ that contain lots of points (think of $\varepsilon > 0$ being barely sub-constant). We will treat these clusters separately, and will focus on the remainder of the pointset for now. So for the remainder of the pointset, we just apply the Spherical LSH to it. This scheme turns out to satisfy the definition of the data-dependent hash family for the remaining points and for distance thresholds r vs. cr; in particular, the hash function is sensitive for the remaining set only! The intuition is that, for any potential query point, there is only a small number of data points within distance $(\sqrt{2}-\varepsilon)R$ – otherwise, they would have formed yet another cluster, which we would have removed – and the larger distances are handled well by the Spherical LSH. Thus, *for a typical pair of data points*, Spherical LSH is sensitive for P (see Definition 3).

How does [6] deal with the clusters? The main observation is that one can enclose such a cluster in a ball of radius $(1-\Omega(\varepsilon^2))R$, which intuitively makes our problem a little bit simpler (after we reduce the radius enough times, the problem becomes trivial). To answer a query, we query *every cluster*, as well as *one part* of the remainder (partitioned by the Spherical LSH). This can be shown to work overall, in particular, we can control the overall branching and depth of the recursion (see [6] for the details).

For both the clusters, as well as the parts obtained from the Spherical LSH, the algorithm recurses on the obtained point subsets. The overall partitioning scheme from [6] can be seen as a tree, where the root corresponds to the whole dataset, and every node either corresponds to a cluster or to an application of the Spherical LSH. One nuance is that, in

different parts of the tree the Spherical LSH partitioning obtains different p_1, p_2 (depending on R). Nonetheless, each time it holds that $p_1 \geq p_2^\rho$ for $\rho \leq \frac{1}{2c^2-1} + o(1)$. Hence, a node terminates as a leaf once its accumulated p_2 (product of p_2's of "Spherical LSH" nodes along the path from the root) drops below $1/n$.

We now want to argue that the above algorithm can be recast in the framework of data-dependent hashing as per Definition 3. We consider a subtree of the overall tree that contains the root and is defined as follows. Fix a parameter $l = n^{-o(1)}$ (it is essentially the target p_2 of the partition). We perform DFS of the tree and cut the tree at any "Spherical LSH" node where the cumulative p_2 drops below l. This subtree gives a partial partition of the dataset P as follows: for "Spherical LSH" nodes we just apply the corresponding partition, and for cluster nodes we "carve" them in the random order. It turns out that if we choose $l = n^{-o(1)}$ carefully, the partition will satisfy Definition 3 and the preconditions of Theorem 4. In particular, the description complexity of the resulting hash function is $n^{o(1)}$.

Let us emphasize that, while Definition 3 is certainly necessary for an ANN data structure based on data-dependent hashing, it is not *sufficient*. In fact, [6] prove additional properties of the above partitioning scheme, essentially because the "p_2 property" is "on average" one (thus, we end up having to understand how this partitioning scheme treats *triples* of points).