Smooth Tradeoffs between Insert and Query Complexity in Nearest Neighbor Search

Michael Kapralov
IBM Watson

December 5, 2014

Abstract

Locality Sensitive Hashing (LSH) has emerged as the method of choice for high dimensional similarity search, a classical problem of interest in numerous applications. LSH-based solutions require that each data point be inserted into a number A of hash tables, after which a query can be answered by performing B lookups. The original LSH solution of [IM98] showed for the first time that both A and B can be made sublinear in the number of data points. Unfortunately, the classical LSH solution does not provide any tradeoff between insert and query complexity, whereas for data (respectively, query) intensive applications one would like to minimize insert time by choosing a smaller A (respectively, minimize query time by choosing a smaller B). A partial remedy for this is provided by Entropy LSH [Pan06], which allows to make either inserts or queries essentially constant time at the expense of a loss in the other parameter, but no algorithm that achieves a smooth tradeoff is known.

In this paper, we present an algorithm for performing similarity search under the Euclidean metric that resolves the problem above. Our solution is inspired by Entropy LSH, but uses a very different analysis to achieve a smooth tradeoff between insert and query complexity. Our results improve upon or match, up to lower order terms in the exponent, best known data-oblivious algorithms for main memory LSH for the Euclidean metric.

1 Introduction

Similarity search is a classical problem of interest to numerous applications in data-mining such as duplicate detection, content-based search [KG09, LJW+07], collaborative filtering [DDGR07], pattern classification [CH67], clustering [Ber02]. In the similarity search problem the algorithm is given a database of objects to preprocess and is then required to find, for each query object q, the object in the database that is closest to q in some metric. In these applications, objects in the database are usually represented by high dimensional feature vectors, resulting in a nearest neighbor search problem in $\mathbb{R}^d$ under an appropriate metric. In this paper, we consider the Euclidean metric, or ℓ_2.

For the exact nearest neighbor problem a family of tree-based approaches have been developed such as K-D trees [Ben75], cover trees [BKL06], navigating nets [KL04], R-trees [Gut84], and SR-trees [KS97]. However, the performance of these techniques degrades very fast with the dimensionality of the problem (known as the 'curse of dimensionality') and in fact degrades to a linear scan of the data quite quickly [WSB98]. Since the exact version of the nearest neighbor problem suffers from the 'curse of dimensionality', substantial attention has been devoted to the *Approximate Nearest Neighbor Problem*. In this problem, instead of reporting the closest point to the query q, the algorithm only needs to return a point that is at most a factor $c > 1$ further away from q than its nearest neighbor in the database. Specifically, let $D = \{p_1, \ldots, p_N\}$ denote a database of points, where $p_i \in \mathbb{R}^d, i = 1, \ldots, N$. In the Euclidean *c-Approximate Nearest Neighbor* problem one is required to report, for each query q, a point $\hat{p} \in D$ such that

$$||q - \hat{p}||_2 \leq c \cdot \min_{p \in D} ||q - p||_2.$$

One can see [IM98, KOR98] that this problem reduces, with a slight overhead in space and time, to the so-called (c, r)-*Near Neighbor* problem ((c, r)-NN for short). In the (c, r)-NN problem the goal is to return a data point within distance cr of the query point q if a data point within distance r of q exists. The simple reduction from the c-Approximate Near Neighbor problem to the (c, r)-NN problem can be obtained by considering a sequence of geometrically increasing radii r, which increases the space and time requirement only by a logarithmic factor.

Locality Sensitive Hashing (LSH) has emerged as the method of choice for the (c, r)-NN problem [IM98]. LSH is based on a special hashing scheme such that similar points have a higher chance of getting mapped to the same buckets than distant points. Then for each query, the nearest neighbor among the data points mapped to the same bucket as the query point is returned as the search result. We now describe the LSH approach formally. We start with the definition of a *locality sensitive family* of hash functions:

Definition 1 *Let the space $\mathbb{R}^d$ be equipped with a norm $||\cdot||$, let $r \geq 0$ be a distance threshold, let $c > 1$. A family of hash functions $\mathcal{H} = \{h : \mathbb{R}^d \to \mathcal{U}\}$ is said to be a (r, cr, p_1, p_2)-LSH family if for all $x, y \in \mathbb{R}^d$,* **(1)** *if $||x - y|| \leq r$, then $\mathbf{Pr}_{h \in \mathcal{H}}[h(x) = h(y)] \geq p_1$; and* **(2)** *if $||x - y|| \geq cr$, then $\mathbf{Pr}_{h \in \mathcal{H}}[h(x) = h(y)] \leq p_2$.*

Points $x, y \in \mathbb{R}^d$ are called *near* points in the former case and *far* points in the latter. An LSH family can be used to obtain the following solution to the (c, r)-NN problem:

Theorem 2 ([IM98]) *Let $\mathcal{H}$ denote a (r, cr, p_1, p_2)-LSH family for a norm $||\cdot||$ on $\mathbb{R}^d$. Then $\mathcal{H}$ can be used to solve the (c, r)-NN problem with norm $||\cdot||$ on a database D on N points using space $dN^{1+\rho+o(1)}$ and query time $dN^{\rho+o(1)}$ as long as $p_1 \geq N^{-o(1)}$, where $\rho = \frac{\log(1/p_1)}{\log(1/p_2)}$.*

The algorithm that yields Theorem 2 hashes the points in the database into N^ρ/p_1 hash tables, resulting in space usage $dN^\rho/p_1 = dN^{\rho+o(1)}$. Then for each query lookups are performed in N^ρ/p_1 tables, resulting in the stated query time. As seen from Theorem 2, the ratio $\rho = \frac{\log(1/p_1)}{\log(1/p_2)}$ governs the quality of the solution provided by LSH, so constructing LSH families with the smallest possible ρ is crucial. The original paper [IM98] exhibited an LSH familty for the Hamming cube with $\rho \leq 1/c$. For the Euclidean metric, which we consider in this paper, a simple and practical LSH familty achieving $\rho \leq \frac{1}{c}$ was constructed in [DIIM04]. An LSH familty achieving $\rho = \frac{1}{c^2} + o(1)$ was constructed in [AI, And09]. This dependence of ρ on c is optimal [MNP06, OWZ11]. The LSH scheme of [AI] implements the *ball-carving* approach. First, the points of the database are projected to a smaller *reduced space* $\mathbb{R}^n$, where $n \ll \log N$. Then the reduced space $\mathbb{R}^n$ is covered by randomly shifted grids of Euclidean balls, and each point in the reduced space is hashed to the lexicographically first ball that covers it. We will use a similar approach as our basic hashing scheme in this paper (see Algorithm 1 in section 3).

The main disadvantage of the conventional LSH indexing scheme is the relatively large number (in practice, up to hundreds [GIM99]) of hash tables required for good search quality. This imposes large space requirements and insert time. To mitigate the space inefficiency, Panigrahy [Pan] introduced the Entropy LSH scheme, which uses only $\tilde{O}(1)$ hash tables as opposed to $N^{1/c}$:

Theorem 3 ([Pan]) *There exists a data structure for solving the (c, r)-NN problem under the ℓ_2 metric in $\mathbb{R}^d$ that uses $\tilde{O}(N)$ space and $\tilde{O}(N^{2.06/c})$ query time for sufficiently large $c > 1$.*

This result guarantees extremely small space (or, insert time) at the expense of an increase of query time from $N^{1/c}$ of [IM98] to $N^{2.06/c}$. Alternatively, [Pan] also showed that the query time can be made very efficient (constant time) at the expense of larger space requirements:

Theorem 4 ([Pan]) *There exists a data structure for solving the (c, r)-NN problem under the ℓ_2 metric in $\mathbb{R}^d$ that uses $\tilde{O}(N^{1/(1-2.06/c)})$ space and polylogarithmic query time.*

In [And09] an algorithm inspired by Entropy LSH is given that achieves $\tilde{O}(n)$ space and query time $N^{O(1/c^2)}$ for the same setting.

As seen from the above, despite its efficiency, the classical LSH solution does not provide any tradeoff between the insert and query complexity. However, for data (respectively, query) intensive workloads one would like to minimize one parameter, even if it entails an increase in the other. Entropy LSH provides a partial remedy for this, allowing to make either inserts or queries extremely efficient ($N^{o(1)}$ time per point) at the expense of a loss in the other parameter, but no algorithm that achieves a *smooth tradeoff* is known. In this paper we provide the first algorithm for nearest neighbor search that achieves a smooth tradeoff between insert and query complexity, improving upon or matching known results for parameter settings that algorithms were known for before.

Our results As before, we denote the database of points by $D = \{p_1, \ldots, p_N\}$. Our main result is

Theorem 5 *Let $\alpha \in [0, 1]$ be a constant. Let $c \geq 1$ be the desired approximation ratio, and assume that $c^2 \geq 3(1-\alpha)^2 - \alpha^2 + \delta$ for an arbitrarily small constant $\delta > 0$. Then there exists a data structure for the (c, r)-NN problem under ℓ_2 with $dN^{\alpha^2\rho_\alpha + o(1)}$ insert time[1], $dN^{(1-\alpha)^2\rho_\alpha + o(1)}$ query time and space $dN^{1+\alpha^2\rho_\alpha(1+o(1))}$, where $\rho_\alpha = \frac{4}{c^2+(1-\alpha)^2-3\alpha^2}$. Furthermore, setting $\alpha = 0$ results in a data structure with space dN, and $\alpha = 1$ results in a single probe data structure. The success probability is $1 - o(1)$ for any fixed query.*

We note that the constraint on the approximation ratio c is only nontrivial when one is interested in very low query complexity, i.e. α is close to 1. When $\alpha = 1$, our condition constrains c to be strictly larger than $\sqrt{3}$. This constraint is inherent to the approach, and is also inherent in Entropy LSH (see Theorem 4 above, where c is required to be larger than a constant).

Note that setting $\alpha = 0$, we get a data structure with space dN, $dN^{o(1)}$ insert time and $dN^{\frac{4}{c^2+1}}$ query time. Prior to our work, the best known scheme with linear space and $O(1/c^2)$ dependence of the exponent was due to [And09], where $dN^{O(1/c^2)}$ dependence was achieved with unspecified constant in the $O(\cdot)$ notation. Setting $\alpha = 1/2$, we obtain a data structure with $dN^{1/c^2+o(1/c^2)}$ insert and query time, as well as $dN^{1+1/(c^2-1/2)+o(1)}$ space, matching, up to lower order terms, the best known exponent of $1/c^2$ obtained in [AI, And09]. Setting $\alpha = 1$, we obtain a single probe data structure with $dN^{4/(c^2-3)+o(1)}$ insert time that succeeds with probability $1 - o(1)$. The query time is $dN^{o(1)}$ and the space is $dN^{1+4/(c^2-3)+o(1)}$.

It is interesting to note that, unlike Entropy LSH, which requires knowledge of the distance between near points up to $1 \pm o(1)$ factor, our scheme only needs an *upper bound* on the distance between near points, similarly to the classical construction of [IM98]. Also, interestingly, our approach yields success probability $1 - o(1)$ even in the linear space or single probe regime, as opposed to $\Omega(1/\log N)$ success probability provided by Entropy LSH [Pan].

Our techniques. At a high level, our algorithm is a natural interpolation between two extremes of Entropy LSH (Theorem 3 and Theorem 4). The algorithm is parameterized by a $\alpha \in [0, 1]$, which governs the tradeoff between the insert and query complexity. Our data structure uses exactly one random hash function, which is selected at initialization. In order to insert a data point p into the data structure, we project p to a reduced space $\mathbb{R}^n$ using a dimensionality reduction matrix S, generate $A = N^{\alpha^2\rho_\alpha(1+o(1))}$ perturbations $Sp + u^i, i = 1, \ldots, A$ of p, and insert p into buckets that these perturbations $Sp + u^i$ hash to. The magnitude of the perturbation u^i is proportional to α. Similarly, given a query q, we generate $B = N^{(1-\alpha)^2\rho_\alpha(1+o(1))}$ perturbations $Sq + v^j, j = 1, \ldots, B$ of q, examine buckets that these

[1]Note that our expressions for runtime have a extra *additive* $o(1)$ term in the exponent. This term is due to the time needed to evaluate our hash function. On the other hand, the space complexity only suffers from a *multiplicative* $1 + o(1)$ loss in the exponent. The latter loss is zero at the extreme points, where $\alpha = 0$ or $\alpha = 1$, yielding strictly linear space ($\alpha = 0$) and single probe ($\alpha = 1$) data structures with success probability $1 - o(1)$ respectively.

perturbations $Sq + v^j$ hash to, and return the closest point found (see Fig. 2) . The magnitude of the perturbation v^j is proportional to $1 - \alpha$.

While our algorithm is inspired by [Pan], our analysis is fundamentally different. In [Pan] correctness of hashing schemes is argued using an entropy based approach. One shows that for a random hash function and two near points p, q the conditional entropy I of $\mathbf{h}(p)$ given $\mathbf{h}$ and q is small, and then generates about 2^I samples from this conditional distribution. This many samples are sufficient to ensure a collision with nontrivial probability. In [And09] Andoni achieves the nearly optimal $O(1/c^2)$ dependence of the exponent on the approximation parameter c at the expense of introducing a more complex framework that still relies on entropy considerations.

In this paper, we take a more direct approach to analyzing our algorithm, avoiding entropy-based arguments altogether. In order to achieve correctness, we need to prove two statements: lower bounds on collision probability for (perturbations of)near points, and upper bounds on collision probability for (perturbations of)far points. For the first claim, we need to prove that for a given pair of near points p, q, with high probability over the choice of the hash function $\mathbf{h}$ and the perturbations $u^i, i = 1, \ldots, A, v^j, j = 1, \ldots, B$ (see Fig. 2) at least one of $Sp + u^i$ collides with at least one of $Sq + v^j$ under our hash function. This claim turns out to be rather delicate: we cannot prove that a fixed perturbation $Sp + u^i$ is likely to collide with at least one of the perturbations $Sq + v^j$ since this is simply not true. One can prove, for example, that a given perturbation $Sp + u^i$ collides with at least one of the $Sq + v^j$'s with nontrivial probability, but that does not lead to the result since such events for different u^i's are dependent (via the q^j's). Instead, we define a point $z := (1 - \alpha)p + \alpha q$ lying on the line segment between p and q, and show that **(a)** at least one of $Sp + u^i$'s collides with Sz under $\mathbf{h}$ with probability $1 - o(1)$ and **(b)** at least one of $Sq + v^j$'s collides with z with probability $1 - o(1)$. A union bound over the failure events for these two claims yields the result.

The proof of the upper bound on the collision probability is also somewhat subtle, and requires a careful setting of parameters. The main issue is that we need to argue about the probability that the pair of points $Sp + u^i, Sq + v^j$ collide under hashing. This probability depends on the distribution of the vector $(Sp + u^i) - (Sq + v^j)$, which is not particularly simple, for example, when u^i and v^j are sampled uniformly from a ball of fixed radius (the most convenient setting for the first claim). To remedy this, we sample the perturbations u^i, v^j from balls whose radii are sampled from an appropriate distribution, so that u^i, v^j are vectors of independent Guassians. Since $S(p-q)$ is Gaussian, this ensures that the vector $(Sp + u^i) - (Sq + v^j) = S(p - q) + u^i - v^j$ is a vector of independent Gaussians, making analysis a manageable task.

Related work The problem of proving lower bounds for nearest neighbor search has received a lot of attention in the literature. The results of [MNP06, OWZ11] Euclidean metric show that $\rho = 1/c^2 + o(1)$ achieved by LSH functions of [AI] is best possible up to lower order terms. The results of [PTW08, PTW] show that that single-probe algorithms for (c, r)-NN under the Euclidean metric must use $N^{1+\Omega(1/c^2)}$ space (these lower bounds hold for the cell probe model). In a recent paper [AINR14] showed how to use LSH functions in a more efficient way than Theorem 2 to obtain better space and query time, namely $N^{1+(7/8)/c^2}$ space and $N^{(7/8)/c^2}$ query time for large c. Unlike previous works, their approach is *data-dependent*: the family of LSH functions is chosen carefully as a function of the database as opposed to sampled uniformly at random from a fixed distribution. It would be very interesting to see if similar analysis can be used to improve our tradeoffs.

Organization We give some definitions and relevant results from probability and high dimensional geometry in section 2. The algorithm is presented in section 3. The analysis is presented in section 4. An outline of the analysis is given in section 4.1, some technical lemmas are presented in section 4.2. Upper and lower bounds on collision probability of far and near points respectively are given in sections 4.3 and 4.4 respectively, and are put together in section 4.5 to obtain a proof of Theorem 5. Proofs omitted from the main body of the paper are given in the Appendix.

2 Preliminaries

Definition 6 *The Gamma distribution with shape parameter $k > 0$ and scale $\theta > 0$, denoted by $\Gamma(k, \theta)$, has the pdf $\frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-x/\theta}$.*

Claim 7 *[BGMN05] Let $X \sim \frac{1}{\sqrt{\pi}} e^{-|x_i|^2}$. Then $|X|^2 \sim \Gamma(1/2, 1)$.*

We will define the distribution $\mathcal{D} \sim 2\Gamma(n/2 + 1, 1)$, where n is the dimension of our reduced space (see section 3 below). This distribution will be used extensively throughout the algorithm. For any real t we write $t \cdot \mathcal{D}$ to denote the

distribution of $t \cdot X$, where $X \sim \mathcal{D}$. We write $\mathcal{N}(0, I_n)$ to denote the Gaussian distibution on $\mathbb{R}^n$ with covariance matrix I_n.

Claim 8 *Let $X \geq 0$ be a random variable. For any event $\mathcal{E}$ one has $\mathbf{E}[X|\mathcal{E}] \leq \frac{1}{\mathbf{Pr}[\mathcal{E}]}\mathbf{E}[X]$.*

In what follows for a vector x we write $||x||$ or $||x||_2$ to denote the ℓ_2-norm of x. We write $\mathbb{B}_R(x)$ to denote the ℓ_2 ball of radius R around x, and write $|\mathbb{B}_R(x)|$ to denote the volume of $\mathbb{B}_R(x)$. Let $C(u, r)$ denote the volume of the spherical cap at distance u from the center of a ball $\mathbb{B}_r(0)$. Let $I(u, r) = \frac{C(u,r)}{|\mathbb{B}_r(0)|}$ be the relative cap volume.

Lemma 9 *Let $a, w \in \mathbb{R}^n$. Let $r, R \in \mathbb{R}^+$ be parameters, and suppose that $R > r$. Let $d = ||a - w||_2$. Let $x = \frac{r^2+d^2-R^2}{2d}$. If $x > 0$, then*

$$I(x) \leq \frac{|\mathbb{B}_r(a) \cap \mathbb{B}_R(w)|}{|\mathbb{B}_r(a)|} \leq 2I(x).$$

We will use

Lemma 10 *[[AI], Lemma 2.1] For any $n \geq 2$ and $0 \leq u \leq r$ one has $\frac{C'}{\sqrt{n}}\left(1 - \left(\frac{u}{r}\right)^2\right)^{n/2} \leq I(u, r) \leq \left(1 - \left(\frac{u}{r}\right)^2\right)^{n/2}$, where C' is an absolute constant.*

3 The algorithm

In this section we describe our algorithm. We denote data points by $p \in \mathbb{R}^d$, and query points by $q \in \mathbb{R}^d$. The number of points in the database is denoted by N, as before. In the preprocessing stage for each point p we perform the following operation K times independently (we index the independent repetitions by $l = 1, \ldots, K$). First, we project p down to dimension $n \ll \log N$ using a dimensionality reduction matrix S_l. Then we perturb $S_l p$ by an appropriately chosen vector of independent Gaussians (the magnitude of the perturbation is proportional α). Finally, we hash the perturbed points using a ball-carving approach described below (see BASICHASH). Thus, for each perturbation of p and for each l we obtain a hash value. We concatenate these values and hash point p into the corresponding bucket. The query phase is analogous, the only difference is the magnitude of the perturbations (specified below). We now describe these steps in details.

Dimensionality reduction. Let $S \in \mathbb{R}^{(K \cdot n) \times d}$ denote a matrix of independent Gaussians of unit variance. We partition the rows of S into K blocks, corresponding to K hash functions. Thus $S_l \in \mathbb{R}^{n \times d}$ is the dimensionality reduction matrix for the l-th hash function. Since entries of S are chosen as Gaussians with unit variance, for any $p, q \in \mathbb{R}^d$ we have $S_l(p - q) \sim ||p - q||_2 \cdot \mathcal{N}(0, I_n)$. We will also write $Sp \in R^{K \cdot n}$ for $p \in \mathbb{R}^d$ to denote the concatenation of $S_l p$'s.

Perturbation. Let d_{near} denote the distance between near points in the original space $\mathbb{R}^d$ (this corresponds to the radius r in the (c, r)-NN problem). For each l the perturbations of a projected data point p are of the form $S_l p + u_l^i$, $i = 1, \ldots, A$, where $u_l^i \sim \alpha d_{near} \cdot \mathcal{N}(0, I_n)$. Note that by Lemma 14 this is the same as first sampling a radius $r_{p,l}$ so that $r_{p,l}^2 \sim (\alpha d_{near})^2 \cdot \mathcal{D}$, and then sampling u_l^i uniformly from the ball $\mathbb{B}_{r_{p,l}}(0)$ (see Algorithm 2). The perturbation for query points is analogous: for each l the perturbations of a projected query point q are obtained as $S_l q + v_l^j$, $j = 1, \ldots, B$, where $v_l^j \sim (1 - \alpha) d_{near} \cdot \mathcal{N}(0, I_n)$ (see Algorithm 3).

Ball-carving. We use the ball-carving approach of [AI] as the basic scheme, i.e. we first project point in $\mathbb{R}^d$ down to smaller dimension $n = o(\log N)$, and then perform ball-carving in the reduced space $\mathbb{R}^n$. We will use ball-carving with ℓ_2 balls as the basic scheme. The (expected) distance between near points after dimensionality reduction will be at most $d_{near}\sqrt{n}$ for $d_{near} = n^{-1/4}$, the distance between far points will be at least $c \cdot d_{near}\sqrt{n}$. We will be carving with Euclidean balls of radius R_l, for $l = 1, \ldots, K$. The radii R_l are sampled independently from the distribution $R_l^2 \sim \mathcal{D}$ at the beginning of the algorithm and passed as a parameters to all functions (see Algorithms 1, 2 and 3).

We now describe how the reduced space $\mathbb{R}^n$ is covered by Euclidean balls. For each $l = 1, \ldots, K$ let $\mathcal{U}_l$ denote a subset of $[0, \sqrt{\pi} n R_l]^n$ of size

$$T = (C \log N) \cdot \frac{(\sqrt{\pi} n R_l)^n}{\text{vol}(\mathbb{B}_{R_l}(0))} = (C \log N) \cdot \frac{(\sqrt{\pi} n R_l)^n}{\pi^{n/2}\Gamma(n/2 + 1) R_l^n} = (C \log N) \cdot \frac{n^n}{(n/2)!} \quad (1)$$

sampled uniformly at random. Here $C > 0$ is an appropriately large constant, and N is the number of points in the database. Note that T is integer as long as n is even and $(C \log N) \geq 1$ is an integer, which we assume from now on. Let $\mathcal{G} := (\sqrt{\pi} n R) \cdot \mathbb{Z}^n$ denote an infinite grid of scaled integer points. Our basic hashing function BASICHASH, for each l, will map the input point to one of the balls of radius R_l centered at shifts $u + \mathcal{G}$ of the grid, for $u \in \mathcal{U}_l$ (note that the balls centered at different grid points do not overlap) . More precisely, for each $l \in [1 : K]$ the centers of the balls are given by

$$\mathcal{W}_l := \mathcal{U}_l + \mathcal{G},$$

where we use the notation $S_1 + S_2 = \{a + b : a \in S_1, b \in S_2\}$ for $S_1, S_2 \subset \mathbb{R}^n$. We refer to points in $\mathcal{W}_l$ as *centers*. First note that for any l a ball of radius R_l around any point $x \in \mathbb{R}^d$ contains exactly $C \log N$ centers in $\mathcal{W}_l$ in expectation by choice of parameters. We will need the fact that with high probability over the choice of $\mathcal{U}_l$ all perturbed points below have about $C \log N$ centers in the ball of radius R_l around them. We now make this precise. Fix a constant $\alpha \in [0, 1]$. Let p, q denote a query and its near point. Define the event

$$\mathcal{E}^*(p, q) := \left\{ \mathbb{B}_{R_l}(S_l p + u_l^i) \cap \mathcal{W}_l \leq 2C \log N \text{ and } \mathbb{B}_{R_l}(S_l q + v_l^j) \cap \mathcal{W}_l \leq 2C \log N \text{ for all } i, j, l \right\}. \tag{2}$$

One has

Claim 11 *Let $\alpha \in [0, 1]$ be a constant. Let $c \geq 1$ be the desired approximation ratio, and assume that $c^2 \geq 3(1-\alpha)^2 - \alpha^2 + \delta$ for an arbitrarily small constant $\delta > 0$, as in Theorem 5. Suppose that the constant C in* (1) *is sufficienlty large. Then for any pair of points p, q one has* $\mathbf{Pr}[\mathcal{E}^*(p, q)] \geq 1 - 1/N$.

Proof: Recall that by assumption of Theorem 5 one has $c^2 \geq 3(1 - \alpha)^2 - \alpha^2 + \delta$ for a constant $\delta > 0$, so the number of perturbations is always bounded by $N^{8/\delta}$, say.

Fix l. Then by standard concentration inequalities on has

$$\mathbf{Pr}[|\mathbb{B}_{R_l}(S_l p + u_l^i) \cap \mathcal{W}_l| > 2C \log N] < N^{-8/\delta - 2}$$

for any i, and

$$\mathbf{Pr}[|\mathbb{B}_{R_l}(S_l q + v_l^i) \cap \mathcal{W}_l| > 2C \log N] < N^{-8/\delta - 2}$$

for any j as long as the constant C is sufficiently large. Now the claim follows by a union bound over all $l = 1, \ldots, K$ and at most $2N^{8/\delta}$ perturbations of p and q. ■

The basic hash function simply returns such a center if it is unique, and a uniformly random element of a large universe otherwise:

Algorithm 1 ℓ_2-ball carving LSH for ℓ_2: hashing data points

1: **procedure** BASICHASH($x, R, n, \mathcal{W}$)
2: **If** $|\mathcal{W}^* \cap \mathbb{B}_R(x)| \neq 1$ **return** $UNIF([0, 1]^n)$ ▷ If no center falls into $\mathbb{B}_R(x)$, return a random element of a large universe
3: **return** a uniformly random element of $\mathcal{W}^* \cap \mathbb{B}_R(x)$
4: **end procedure**

The function BASICHASH can be evaluated in time $n^{O(n)}$, which we will ensure to be $N^{o(1)}$ below (see Claim 20 in Appendix A for the runtime bound).

Algorithm 2 ℓ_2-ball carving LSH for ℓ_2: hashing data points

procedure HASHDATA($p, \alpha, S, K, \{R_l\}_{l=1}^K, B, n, , \{\mathcal{U}_l\}_{l=1}^K, d_{near}$) ▷ We have $d_{near} = n^{-1/4}$ and $R_l \approx \sqrt{n}$
  for $l = 1, \ldots, K$ **do**
    Sample $r_{p,l}$ from distribution given by $r_{p,l}^2 \sim (\alpha d_{near})^2 \cdot \mathcal{D}$.
  end for
  for $j = 1, \ldots, A$ **do** ▷ Generating A points around p
    for $l = 1, \ldots, K$ **do**
      $u_l^i \leftarrow UNIF(\mathbb{B}_{r_{p,l}}(0))$ ▷ So that $u_l^i \sim (\alpha d_{near})^2 \cdot \mathcal{N}(0, I_n)$
      $h_l \leftarrow$ BASICHASH$(S_l p + u_l^i, R_l, n, \mathcal{U}_l)$
    end for
    PUT$(\mathbf{h}, p)$ ▷ Insert $\langle \mathbf{h}, p \rangle$ into hash table
  end for
end procedure

Queries are performed as follows:

Algorithm 3 ℓ_2-ball carving LSH for ℓ_2: query

procedure QUERY($q, \alpha, S, K, \{R_l\}_{l=1}^K, A, n, \{\mathcal{U}_l\}_{l=1}^K, d_{near}$) ▷ We have $d_{near} = n^{-1/4}$ and $R_l \approx \sqrt{n}$
  $T \leftarrow \emptyset$ ▷ Candidate points
  for $l = 1, \ldots, K$ **do**
    Sample $r_{q,l}$ from distribution given by $r_{q,l}^2 \sim ((1-\alpha) d_{near})^2 \cdot \mathcal{D}$.
  end for
  for $i = 1, \ldots, B$ **do** ▷ Generating B points around q
    for $l = 1, \ldots, K$ **do**
      $v_l^i \leftarrow UNIF(\mathbb{B}_{r_{q,l}}(0))$ ▷ So that $v_l^j \sim ((1-\alpha) d_{near})^2 \cdot \mathcal{N}(0, I_n)$
      $h_l \leftarrow$ BASICHASH$(S_l q + v_l^i, R_l, n, \mathcal{U}_l)$
    end for
    If $|T| > N^{(1-\alpha)^2 \rho_\alpha + o(1)}$ **then break**
    $T \leftarrow T \cup$ GET$(\mathbf{h})$ ▷ Retrieve at most $N^{(1-\alpha)^2 \rho_\alpha + o(1)}$ points from hash bucket
  end for
  return closest point to q in T
end procedure

Note that the main difference between Algorithms 2 and 3 is the magnitude of the perturbations to projected points (see Fig. 1). We gather useful properties of random variables used in Algorithms 2 and 3 below.

Claim 12 *Let p be a data point, let q be a query point such that $p - q = \lambda d_{near}$. Let $i \in [A], j \in [B], l \in [K]$. Then the vector $(S_l p + u_l^i) - ((S_l q + v_l^j)$ is uniformly random in the ball $\mathbb{B}_{r'}(0)$, where*

$$(r')^2 \sim (\lambda^2 + \alpha^2 + (1-\alpha)^2) d_{near}^2 \cdot \mathcal{D}.$$

and $(S_l p + u_l^i) - (S_l q + v_l^j) \sim \sqrt{\lambda^2 + \alpha^2 + (1-\alpha)^2} d_{near} \cdot \mathcal{N}(0, I_n)$.

Proof: Recall that u_l^i is sampled uniformly at random from $\mathbb{B}_{r_{p,l}}(0)$, where $r_{p,l}^2 \sim (\alpha d_{near})^2 \cdot \mathcal{D}$. By Corollary 15 we thus have that $u_l^i \sim \alpha d_{near} \cdot \mathcal{N}(0, I_n)$. Similarly, $v_l^j \sim (1-\alpha) d_{near} \cdot \mathcal{N}(0, I_n)$. Also, $S_l p - S_l q \sim \gamma d_{near} \cdot \mathcal{N}(0, I_n)$ by the choice of S and 2-stability of the Gaussian disrtibution. Thus, we have

$$(S_l p + u_l^i) - (S_l q + v_l^j) \sim \sqrt{||p-q||_2^2 + \alpha^2 d_{near}^2 + (1-\alpha)^2 d_{near}^2} \cdot \mathcal{N}(0, I_n).$$

The distribution of $(r')^2$ follows by Corollary 15. ■

Claim 13 *Let $x \in \mathbb{R}^d$. Let $w_l :=$ BASICHASH$(S_l x, R_l, n, \mathcal{U}_l)$, and condition on $\mathcal{E}^*$. Then w_l is a uniformly random point in $\mathbb{B}_{R_l}(x)$. In particular, $w_l - x \sim \mathcal{N}(0, I_n)$.*

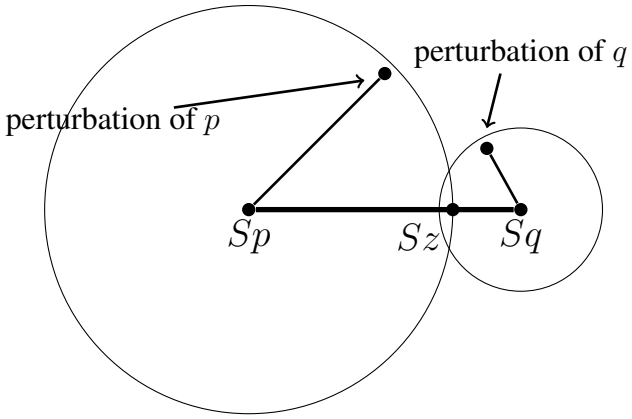


Figure 1: Perturbations of data points and query points in projected space. The radii of balls around projected points Sp and Sq are about $\alpha||Sp - Sq||$ and $(1-\alpha)||Sp - Sq||$ respectively. The radii are sampled from a distribution independently, so they do not add up to $||Sp - Sq||$ in general.

Proof: By conditioning on $\mathcal{E}^*$ the output of BASICHASH is a center in $\mathcal{W}_l$. Since BASICHASH outputs a uniformly random center that falls into $\mathbb{B}_{R_l}(x)$, the first claim follows. Further, since $R_l^2 \sim \mathcal{D}$ by definition of R_l, we have that $w_l - x \sim \mathcal{N}(0, I_n)$ by Corollary 15. ■

In the next section we give the analysis of our algorithms, resulting in a proof of Theorem 5. We give a glossary of main parameters here for convenience of the reader.

Glossary of parameters

- α – parameter governing balance between insert and query complexity. $\alpha \in [0, 1]$ is assumed to be an absolute constant.
- N – number of points in the database
- d – dimension of the original space
- d_{near} – distance between near points in the original space (the r in (c, r)-NN problem)
- n – dimension of the reduced space, i.e. the space in which BASICHASH performs ball-carving
- K – number of hash functions to concatenate
- R_l – radius of balls used for carving in the reduced space for the l-th hash function, $l = 1, \ldots, K$. R_l is sampled from the distribution $R_l^2 \sim \mathcal{D}$.
- $\mathcal{W} = \bigcup_{l=1}^{K} \mathcal{W}_l$ – set of centers in reduced space that points are hashed to. The sets $\mathcal{W}_l$ are inifiinite, since $\mathcal{W}_l = \mathcal{U}_l + \mathcal{G}$ ($\mathcal{U}_l$ are the shifts), and they are represented implicitly by $\mathcal{U}_l$.
- $S_l \in \mathbb{R}^{n \times d}, l = 1, \ldots, K$ – dimensionality reduction matrices for the l independent hash functions (each S_l is a matrix of independent unit variance Gaussians).
- r_l – radius of the small balls around projected points $S_l p, S_l q$ that we sample perturbed points from (see Fig. 1).
- $C \log N$ – expected number of centers $w \in \mathcal{W}$ that belong to the ball $\mathbb{B}_R(a)$ for a typical point $a \in \mathbb{R}^n$ in the reduced space.

4 Analysis

In this section we give the formal analysis of our algorithm. We start with an outline in section 4.1, then present some technical lemmas in section 4.2. Upper and lower bounds on collision probability of far and near points respectively are given in sections 4.3 and 4.4 respectively, and are put together in section 4.5 to obtain a proof of Theorem 5.

4.1 Proof outline

We now give intuition for the analysis. Recall that algorithm first projects a pair of points $p, q \in \mathbb{R}^d$ to $\mathbb{R}^n$ for a slowly growing parameter n. Then at preprecessing time $A \approx N^{\alpha^2/c^2}$ random points in a ball of radius proportional to α around data point p are generated. Similarly, at query time $B \approx N^{(1-\alpha)^2/c^2}$ random points in a ball of radius proportional to $1-\alpha$ around q are generated (see Algorithm 2 and 3 respectively).These points are then hashed using BASICHASH: the reduced space $\mathbb{R}^n$ is covered with a sufficient number of Euclidean balls, and a point is hashed to a uniformly random ball in our collection that contains it. This is repeated K times independently, and the outputs are concatenated to obtain the final hash function.

In order to establish our result, we need to prove lower bounds on collision probability for (perturbations of) near points, and upper bounds on collision probability for (perturbations of) far points.

Near points. Given a pair of points p, q such that $||p-q|| \leq d_{near}$, we need to argue that with probability at least $1-o(1)$ over the choice of the hash function $\mathbf{h}$ and perturbations u^i, v^j (see Fig. 1) one has $\mathbf{h}(Sp+u^i) = \mathbf{h}(Sq+v^j)$ for at least one pair i, j. This claim turns out to be rather delicate: we cannot prove that a fixed perturbation $p+u^i$ is likely to collide with at least one of the perturbations $q+v^j$ since this is simply not true. One can prove, for example, that a given perturbation $p+u^i$ collides with at least one of the $q+v^j$'s with nontrivial probability, but that does not lead to the result since such events for different u^i's are dependent (via the q^j's). Instead, we define the point $z = (1-\alpha)p+\alpha q$ on the line segment joining p and q, and argue that **(1)** $\mathbf{h}(Sq+v^j) = \mathbf{h}(Sz)$ for at least one j and **(2)** $\mathbf{h}(Sp+u^i) = \mathbf{h}(Sz)$ for at least one i with probability $1-o(1)$ over the choice of $\mathbf{h}$ and $\{u^i\}, \{v^j\}$. Steps **(1)** and **(2)** are similar, so we outline the proof of **(1)** only.

We first note that $\mathbf{h}(Sq+v^j) = \mathbf{h}(Sz)$ if and only if $\mathbf{h}_l(Sq+v^j) = \mathbf{h}_l(Sz)$ for all l, i.e. the collision should happen in every independent repetition. Now fix l, and let $w_l \in \mathcal{W}_l$ denote the center that $S_l z$ hashes to under BASICHASH. By definition of BASICHASH our point $S_l q + v_l^j$ collides with $S_l z$ if and only if two conditions are satisified: **(a)** $w_l \in \mathbb{B}_{R_l}(S_l q + v_l^j)$ and **(b)** w_l is chosen as the center to be hashed to in line 3 of BASICHASH. Condition **(b)** is easy to handle, so we describe our approach to ensuring **(a)**.

Note that for a fixed hash function $\mathbf{h}$ and fixed $i = 1, \ldots, B$ we have

$$\mathbf{Pr}_{v^j}\left[w_l \in \mathbb{B}_{R_l}(S_l q + v_l^j) \text{ for all } l = 1, \ldots, K\right] = \prod_{l=1}^{K} \mathbf{Pr}_{v^j}\left[w_l \in \mathbb{B}_{R_l}(S_l q + v_l^j)\right]$$

by independence of perturbations v^j across different repetitions. On the other hand, since the perturbed query point $S_l q + v_l^j$ is uniformly random in the ball $\mathbb{B}_{r_{q,l}}(S_l q)$, we have

$$\mathbf{Pr}_{v^j}\left[w_l \in \mathbb{B}_{R_l}(S_l q + v_l^j)\right] = \frac{|\mathbb{B}_{r_{q,l}}(S_l q) \cap \mathbb{B}_{R_l}(w_l)|}{|\mathbb{B}_{r_{q,l}}(S_l q)|}. \tag{3}$$

Denote the rhs by ξ_l, and note that ξ_l is a random variable that depends on the projection matrix S_l, random shifts $\mathcal{U}_l$ and radius R_l used in repetition l. In section 4.4 below we prove a concentration result for $\prod_{l=1}^{K} \xi_l$, implying that the lhs of (3) is quite tightly concentrated. More precisely, we prove that $\sum_{l=1}^{K} \ln \xi_l$ is within a $1+o(1)$ factor of its expectation with probability $1-o(1)$, which is sufficient for our purposes. Note that concentration is not immediate, since the rhs of (3) may in general take arbitrarily small values. However, a careful choice of parameters allows to control the variance and prove concentration.

The details of this argument are presented in section 4.4. The argument requires a good estimate for

$$\mathbf{E}[\ln \xi_l] = \mathbf{E}\left[\frac{|\mathbb{B}_{r_{q,l}}(S_l q) \cap \mathbb{B}_{R_l}(w_l)|}{|\mathbb{B}_{r_{q,l}}(S_l q)|}\right].$$

Such an estimate is provided by Lemma 16 below, one of our two main technical lemmas.

Far points Consider a pair of far points $p, q \in \mathbb{R}^d$, i.e. $||p-q|| \geq c \cdot d_{near}$. We need to prove that the perturbations of p and q in projected space, i.e. $S_l p + u_l^i$ and $S_l q + v_l^j$, are unlikely to collide under all K hash functions. Fix l and let $w_l \in \mathcal{W}_l$ denote the center that $S_l q + v_l^j$ hashes to. We need to upper bound the probability that $S_l p + u_l^i$ belongs to the ball $\mathbb{B}_{R_l}(w_l)$. This quantity depends on the distribution of

$$(S_l p + u_l^i) - (S_l q + v_l^j), \tag{4}$$

which is in general quite complicated if radii $r_{p,l}, r_{q,l}$ of the small balls that perturbations are sampled from are fixed. This is the reason why we sample these radii from the (scaled) distribution $\mathcal{D}$ – this sampling ensures that u_l^i, v_l^j are simply vectors of independent Gaussians. But $S_l(p-q)$ also is a vector of independent Gaussians by the choice of S. Thus, (4) is just a vector of independent Gaussians. Finally, again by Lemma 14, (4) is a uniformly random vector in a ball of radius r that satisfies $r^2 \sim \gamma^2 \mathcal{D}$ for some $\gamma > 0$. Thus, all we are interested in is the expectation of

$$\frac{|\mathbb{B}_r(a) \cap \mathbb{B}_R(w_l)|}{|\mathbb{B}_r(a)|},$$

where $r^2 \sim \gamma^2 \cdot \mathcal{D}$, $a = S_l q + v_l^j$, $w_l - a \sim \mathcal{N}(0, I_n)$, and $R^2 = ||w-a||^2 + Y, Y \sim 2\Gamma(1,1)$. A bound on this quantity is provided by Lemma 17. The details of the analysis outlined above are provided in section 4.3.

In the rest of this section we state our main technical lemmas in section 4.2, prove the upper bound on collision probability in section 4.3 and prove the lower bound in section 4.4. We then put these results together in section 4.5 to obtain a proof of Theorem 5.

4.2 Technical lemmas

The follows lemma will be very useful for our analysis:

Lemma 14 *Let $X_i \sim \frac{1}{\sqrt{\pi}} e^{-x^2}$, $i = 1, \ldots, n$. Let Y be exponential with mean 1. Let $R = (X_1^2 + \ldots + X_n^2 + Y)^{1/2}$. Then $(X_1, \ldots, X_n)$ is uniformly distributed in the ball $R \cdot \mathbb{B}(0)$.*

The proof of Lemma 14 is given in Appendix C. Note that the random variables X_i for $p = 2$ in Lemma 14 are Gaussian with variance $1/2$. Since we work with unit norm Gaussians, we need to introduce appropriate scaling:

Corollary 15 *Let $X_i \sim N(0,1)$, $i = 1, \ldots, n$. Let $Y \sim 2\Gamma(1,1)$. Let $R = (X_1^2 + \ldots + X_n^2 + Y)^{1/2}$, so that $R^2 \sim \mathcal{D}$. Then $(X_1, \ldots, X_n)$ is uniformly distributed in the ball $R \cdot \mathbb{B}(0)$.*

The following two lemmas will be our main tool in bounding collision probability for near and far points in section 4.3 and 4.4 below.

Lemma 16 (Lower bounds on collision probability in reduced space) *Let $a, b \in \mathbb{R}^n$ such that $a - b \sim \gamma' \cdot \mathcal{N}(0, I_n)$. Let $r^2 \sim \gamma^2 \cdot \mathcal{D}$, where $\gamma = o_n(1)$, $\gamma' \leq \gamma$. Let $w - a \sim \mathcal{N}(0, I_n)$, and let $R^2 = ||w-a||^2 + Y, Y \sim 2\Gamma(1,1)$(see Fig. 2(a) for an illustration). Let*

$$\xi := \frac{|\mathbb{B}_r(b) \cap \mathbb{B}_R(w)|}{|\mathbb{B}_r(b)|}.$$

Then there exists an event $\mathcal{E}$ with $\mathbf{Pr}[\bar{\mathcal{E}}] \leq e^{-\gamma^2 n}$ such that $\mathbf{E}[\ln \xi | \mathcal{E}] \geq -\frac{1}{2}\gamma^2 n(1 + o(1))$. Furthermore, one has $\ln \xi > -n$ conditional on $\mathcal{E}$.

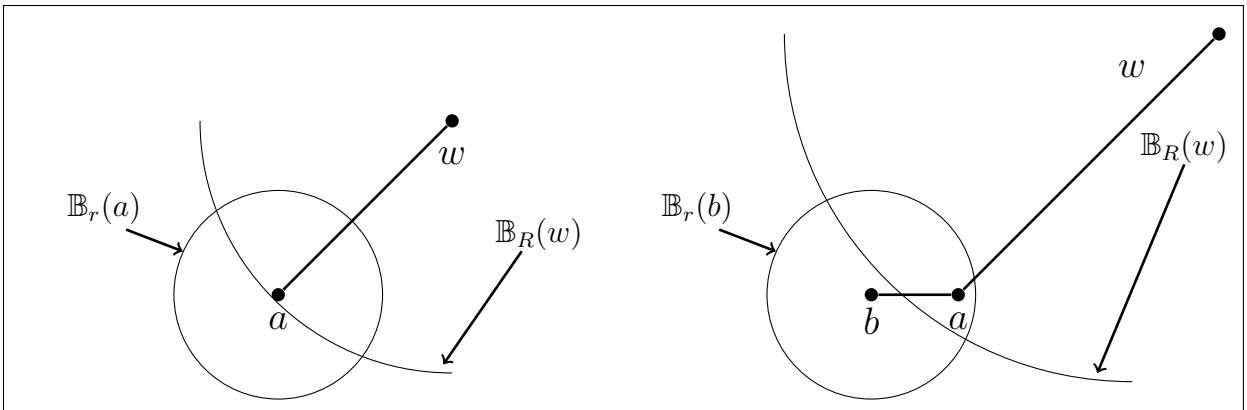
The proof of this lemma is given in Appendix B. The next lemma will be useful for upper bounding collision probability:

Lemma 17 *Let $a \in \mathbb{R}^n$. Let $r^2 \sim \gamma^2 \cdot \mathcal{D}$, where γ is such that $\gamma^2 n = \omega(1)$. Let $w - a \sim \mathcal{N}(0, I_n)$, and let $R^2 = ||w-a||^2 + Y, Y \sim 2\Gamma(1,1)$ (see Fig. 2(b) for an illustration). Let*

$$\xi := \frac{|\mathbb{B}_r(a) \cap \mathbb{B}_R(w)|}{|\mathbb{B}_r(a)|}.$$

Then $\mathbf{E}[\xi] \leq 2exp\left(-\frac{1}{8}\gamma^2 n(1 - O(\gamma))\right)$.

The proof of this lemma is given in Appendix B. We note this is quite similar to the upper bound on the probability p_2 that two points at distance $2r$ from each other collide under hashing proved in [AI] (see equation (2) in [AI]). Indeed, since b is uniformly random in a ball of radius r as above, $a - b$ can be viewed as the Gaussian projection of a fixed length vector $x - y$ in $\mathbb{R}^d$. We are thus intrested in the probability that w, the point that a hashes to, is within distance R of b. This is up to a factor of 2 the quantity studied in [AI], with the minor difference that the radius R of the balls that we are carving with is sampled from a distribution rather than fixed.



(a) Illustration for Lemma 16. The boundary of the ball $\mathbb{B}_R(w)$ is shown.

(b) Illustration for Lemma 17. The boundary of the ball $\mathbb{B}_R(w)$ is shown.

4.3 Upper bound on collision probability for far points

In this and the next section we state the main lemmas of our analysis (the proofs are deferred to the appendix due to space constraints), and then put them together to obtain a proof of Theorem 5 in section 4.5.

We gather some of the random variables used in Algorithms 2 and 3 here for convenience of the reader. Recall that $S \in \mathbb{R}^{(K\cdot n)\times d}$ is a matrix of independent Gaussians of unit variance used for dimensionality reduction. For each $l = 1, \ldots, K$ we have

1. $r_{p,l}^2 \sim (\alpha d_{near})^2 \cdot \mathcal{D}$, $r_{q,l}^2 \sim ((1-\alpha)d_{near})^2 \cdot \mathcal{D}$.
2. $u^i = (u_1^i, \ldots, u_K^i) \in \mathbb{R}^{K\cdot n}, i = 1, \ldots, A$, $v^j = (v_1^j, \ldots, v_K^j) \in \mathbb{R}^{K\cdot n}, j = 1, \ldots, B$ be sampled by choosing u_l^i independent uniformly random in $\mathbb{B}_{r_{p,l}}(0)$, and v_l^j uniformly at random in $\mathbb{B}_{r_{q,l}}(0)$.

Also, $R_l, l = 1, \ldots, K, R_l^2 \sim \mathcal{D}$ are the radii of the balls that BASICHASH uses for carving. $\mathcal{U}_l, l = 1, \ldots, K$ are the shifts of the grids $\mathcal{G}$, and $\mathcal{W}_l = \mathcal{U}_l + \mathcal{G}$ is the set of centers of balls used for carving.

Lemma 18 *Let $\alpha \in [0,1]$ be a constant. Let $c > 1$ denote the desired approximation ratio, and suppose that $K = n^{\Theta(1)}$. Let $d_{near} = n^{-1/4}$. Let $p, q \in \mathbb{R}^d$ be a pair of far points, i.e. $||p-q||_2 \geq cd_{near}$. Consider an invocation of* HASHDATA$(p, \alpha, S, K, \{R_l\}_{l=1}^K, B, n, \{\mathcal{U}_l\}_{l=1}^K, d_{near})$ *and* QUERY$(q, \alpha, S, K, \{R_l\}_{l=1}^K, A, n, \{\mathcal{U}_l\}_{l=1}^K, d_{near})$.

Then for each $i \in [1:A], j \in [1:B]$ one has

$$\mathbf{Pr}[\mathbf{h}(Sp + u^i) = \mathbf{h}(Sq + v^j)] \leq e^{-(c^2+\alpha^2+(1-\alpha)^2)\cdot d_{near}^2(1-o(1))nK/8}.$$

The proof of the Lemma is given in Appendix B.

4.4 Lower bound for near points

Lemma 19 *Let $\alpha \in [0,1]$ be a constant. Let $c > 1$ denote the desired approximation ratio, and suppose that $K = n^{\Theta(1)}$, $Kd_{near}^2 = n^{\Omega(1)}, d_{near}^2 n = n^{\Omega(1)}, n = \omega(1), n = o(\log N)$. Let $p, q \in \mathbb{R}^d$ be a pair of near points, i.e. $||p-q||_2 \leq d_{near}$. Consider an invocation of* HASHDATA$(p, \alpha, S, K, \{R_l\}_{l=1}^K, B, n, \{\mathcal{U}_l\}_{l=1}^K, d_{near})$ *and* QUERY$(q, \alpha, S, K, \{R_l\}_{l=1}^K, A, n, \{\mathcal{U}_l\}_{l=1}^K, d_{near})$. *Then*

$$\mathbf{Pr}[\exists i \in [1:A], j \in [1:B] \text{ s.t. } \mathbf{h}(Sp + u^i) = \mathbf{h}(Sq + v^j)] = 1 - o(1)$$

as long as

$$A \geq (C \log N)^{2K} e^{(1+o(1))\alpha^2 d_{near}^2 nK/2},$$

and

$$B \geq (C \log N)^{2K} e^{(1+o(1))(1-\alpha)^2 d_{near}^2 nK/2}.$$

If $\alpha = 0$ then setting $A = 1$ and B as above is sufficient. Similarly, if $\alpha = 1$, setting A as above and $B = 1$ is sufficient.

The proof of the Lemma is given in Appendix B.

4.5 Putting it together

We can now give the proof of our main theorem:

Proof of Theorem 5:

Let $K' = d_{near}^2 nK/2$, and recall that $d_{near} = n^{-1/4}$. We choose the scaling so that $||p-q||_2 \leq d_{near}$ for near points and $||p-q||_2 \geq c d_{near}$ for far points. We choose

$$K' := \frac{4}{c^2 + \alpha^2 - 3(1-\alpha)^2} \ln N,$$

and set K, n as follows. Recal that $K' = d_{near}^2 nK/2 = n^{1/2}K/2$, i.e. $n^{1/2}K = 2K'$. We let $K = (2K')^{3/4}$ and $n = (2K')^{1/2}$. Note that we have $K = n^{\Theta(1)}$, as required by Lemma 18. Furtheremore, we have $K d_{near}^2 = K n^{-1/2} = (2K')^{3/4-1/4} = (2K')^{1/2} = \omega(1)$ and $d_{near} n = n^{1/2} = n^{\Omega(1)}$, as required by Lemma 19. We now verify that the factor $(2C \log N)^{2K}$ that arises in Lemma 19 is $N^{o(1)}$, and that the BASICHASH function can be computed in time $N^{o(1)}$ for our choice of parameters.

For the first claim, note that we have $K = O(\log^{3/4} N)$, and hence

$$(2C \log N)^K = e^{O(\log^{3/4} N \log\log N)} = N^{o(1)}. \tag{5}$$

Also, we have

$$n^{O(n)} = (\log N)^{O(\log^{1/2} N)} = N^{o(1)}, \tag{6}$$

implying that BASICHASH can be computed in time $N^{o(1)}$ by Claim 20.

Let

$$A = (2C \log N)^{2K} e^{K'\alpha^2(1+\epsilon)} = N^{\alpha^2 \rho_\alpha (1+o(1))}$$
$$B = (2C \log N)^{2K} e^{K'(1-\alpha)^2(1+\epsilon)} = N^{(1-\alpha)^2 \rho_\alpha (1+o(1))},$$

where $\epsilon = o(1)$ as in Lemma 19. When $\alpha = 0$, we set $A = 1$ and B as above, and when $\alpha = 1$, we set $B = 1$ and A as above, in accordance with Lemma 19. The space and insert and query complexity are immediate. Correctness is guaranteed by Lemma 19 if pruning were not done in lines 11 and 12 of Algorithm 3. We now argue correctness formally.

Consider a query q. We now show that the number of collisions of perturbations $q + v^j$ of q with perturbations $p' + u^{j'}$ of points p' that are **far** from q is bounded by $N^{(1-\alpha)^2\rho_\alpha + o(1)}$. By Lemma 18 the expected number of such collisions is bounded by

$$ABNe^{-\frac{1}{4}(c^2+\alpha^2+(1-\alpha)^2)K'(1-o(1))}. \tag{7}$$

Indeed, this is because there are N points in the database, each of which is inserted into the hash table A times, and the near neighbor query for point q performs B lookups. The expected query time is bounded by B plus (7). The latter term equals

$$\begin{aligned}
ABNe^{-\frac{1}{4}(c^2+\alpha^2+(1-\alpha)^2)K'(1-o(1))} &= (2C\log N)^{4K} \cdot Ne^{(\alpha^2+(1-\alpha)^2-\frac{1}{4}(c^2+\alpha^2+(1-\alpha)^2))K'(1-o(1))} \\
&= N^{1+o(1)} e^{-\frac{1}{4}(c^2-3\alpha^2-3(1-\alpha)^2)K'} \\
&= N^{o(1)} \cdot e^{\frac{1}{4}(c^2+(1-\alpha)^2-3\alpha^2)K'} \cdot e^{-\frac{1}{4}(c^2-3\alpha^2-3(1-\alpha)^2)K'} \\
&= N^{o(1)} \cdot e^{(1-\alpha)^2 K'} = N^{(1-\alpha)^2\rho_\alpha + o(1)}.
\end{aligned}$$

Thus, by Markov's inequality the number of collisions of perturbations $q + v^j$ of q with perturbations $p' + u^{j'}$ of points p' that are far from q is bounded by $N^{(1-\alpha)^2\rho_\alpha+o(1)}$ with probability $1 - o(1)$, so the pruning step does not prune away a near point if it exists, and correctness follows. ■

References

[AI] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS'06*.

[AINR14] Alexandr Andoni, Piotr Indyk, Huy L. Nguyen, and Ilya Razenshteyn. Beyond locality-sensitive hashing. *SODA*, 2014.

[And09] Alexandr Andoni. *Nearest Neighbor Search: the Old, the New, and the Impossible*. Ph.D. Thesis, MIT, 2009.

[Ben75] J. Bentley. Multidimensional binary search trees used for associative searching. In *Comm. ACM*, 1975.

[Ber02] P. Berkhin. A survey of clustering data mining techniques. Springer, 2002.

[BGMN05] Franck Barthe, Olivier Guédon, Shahar Mendelson, and Assaf Naor. A probabilistic approach to the geometry of the l_p^n-ball. *The Annals of Probability*, 33:480–513, 2005.

[BKL06] A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbors. In *ICML*, 2006.

[CH67] T. Cover and P. Hart. Nearest neighbour pattern classification. In *IEEE Trans. on Inf. Theory*, 1967.

[DDGR07] A. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: Scalable online collaborative filtering. In *WWW*, 2007.

[DIIM04] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Symposium on Computational Geometry*, pages 253–262, 2004.

[GIM99] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *VLDB*, 1999.

[Gut84] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *SIGMOD*, 1984.

[IM98] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, 1998.

[KG09] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search. In *ICCV*, 2009.

[KL04] R. Krauthgamer and J. Lee. Navigating nets:simple algorithms for proximity search. In *SODA*, 2004.

[KOR98] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search of approximate nearest neighbor in high dimensional spaces. In *STOC*, 1998.

[KS97] N. Katayama and S. Satoh. The sr-tree: an index structure for high-dimensional nearest neighbor queries. In *SIGMOD*, 1997.

[LJW+07] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li. Multi-probe lsh: Efficient indexing for high-dimensional similarity search. In *VLDB*, 2007.

[MNP06] R. Motwani, A. Naor, and R. Panigrahy. Lower bounds on locality sensitive hashing. In *SCG '06: Proceedings of the twenty-second annual symposium on Computational geometry*, pages 154–157, 2006.

[OWZ11] Ryan O'Donnell, Yi Wu, and Yuan Zhou. Optimal lower bounds for locality sensitive hashing (except when q is tiny). *ITCS*, 2011.

[Pan] Rina Panigrahy. Entropy based nearest neighbor search in high dimensions. In *SODA'06*.

[PTW] R. Panigrahy, K. Talwar, and U. Wieder. Lower bounds on near neighbor search via metric expansion. *FOCS'10*.

[PTW08] Rina Panigrahy, Kunal Talwar, and Udi Wieder. A geometric approach to lower bounds for approximate near-neighbor search and partial match. *FOCS*, pages 414–423, 2008.

[RR91] S. T. Rachev and L. Ruschendorf. Approximate independence of distributions on spheres and their stability properties. *The Annals of Probability*, 19(3):1311–1337, 07 1991.

[SZ90] G. Schechtman and J. Zinn. On the volume of intersection of two l_p balls. *Proc. Amer. Math. Soc.*, 110:217–224, 1990.

[WSB98] R. Weber, H. Schek, and S. Blott. A quantititative analysis and performance study for similarity search methods in high dimensional spaces. In *VLDB*, 1998.

A Omitted proofs from section 3

Claim 20 BASICHASH *can be implemented to run in expected time* $(C \log n)n^{O(n)}$.

Proof: In order to implement BASICHASH it is sufficient, given input point x, to form a list $\{w_1, \ldots, w_Q\}$ of centers in $\mathcal{W}_l$ that belong to $\mathbb{B}_{R_l}(x)$, and output a uniformly random such center. To form the list, we consider the $T = (C \log N)n^{O(n)}$ shifted grids $u + \mathcal{G}$, $u \in \mathcal{U}_l$. For a fixed shift u, it is sufficient to round $x - u$ to the closest grid point (this can be done in $O(n)$ time), and check if this grid point is within Euclidean distance R_l of x. Thus, BASICHASH can be implemented in stated time. ■

B Omitted proofs from section 4

We will need

Claim 21 *Let* $X \sim \Gamma(k, 1)$ *for some* $k \geq 1$. *Then for any* $\delta \in (0, 1)$ *one has*

$$\mathbf{Pr}[X \notin (1 \pm \delta)\mathbf{E}[X]] < e^{-\Omega(\delta^2 k)}.$$

Proof of Lemma 9: Note that $\mathbb{B}_r(a) \cap \mathbb{B}_R(w)$ is the union of a spherical cap of $\mathbb{B}_r(a)$ and a spherical cap of $\mathbb{B}_R(w)$. Since $R > r$, the volume of the first spherical cap is at least the volume of the second one, so we concentrate on bounding this volume. Let x be the distance from p to the plane that defines the smaller cap. Then $r^2 - x^2 = R^2 - (d - x)^2$. Since $R^2 - (d - x)^2 = R^2 - d^2 + 2dx - x^2$, this implies $x = \frac{r^2 + d^2 - R^2}{2d}$ as required.

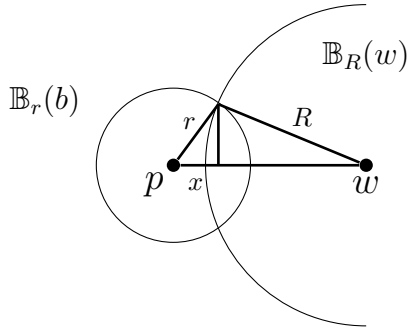


Figure 2: Intersection of $\mathbb{B}_r(a)$ and $\mathbb{B}_R(w)$.

■

Proof of Lemma 16: By Lemma 9

$$\xi \geq I\left(\frac{||c-b||^2 + r^2 - R^2}{2||c-b||}, r\right).$$

First note that

$$\begin{aligned} ||c-b||^2 - R^2 &= ||c-a||^2 + ||a-b||^2 + 2\langle c-a, a-b\rangle - R^2 \\ &= (||c-a||^2 - R^2) + ||a-b||^2 + 2||a-b|| \cdot \langle c-a, \frac{a-b}{||a-b||}\rangle \\ &= -Y + ||a-b||^2 + 2||a-b|| \cdot Z, \end{aligned}$$

where $Z = \langle c-a, \frac{a-b}{||a-b||}\rangle \sim N(0,1)$ is Gaussian. Thus, by Lemma 10

$$\xi \geq \frac{C'}{\sqrt{n}}\left(1 - \left(\frac{-Y + r^2 + ||a-b||^2 + 2||a-b|| \cdot Z}{2||c-b||r}\right)^2\right)^{n/2} \tag{8}$$

for a constant $C' > 0$.

We now show that all quantities involved in (8) are tightly concentrated, and use this fact to prove the claimed result. By Claim 21 with $\delta = \Theta(\gamma)^2$ one has

1. $\mathbf{Pr}[r \notin (1 + O(\gamma))\gamma\sqrt{n}] < e^{-\gamma^2 n}$
2. $\mathbf{Pr}[||a - b|| \notin (1 + O(\gamma))\gamma'\sqrt{n}] < e^{-\gamma^2 n}$
3. $\mathbf{Pr}[||c - b|| \notin (1 + O(\gamma))\sqrt{1 + \gamma^2}\sqrt{n}] < e^{-\gamma^2 n}$

Before we prove the claim of the lemma, we show that

$$\frac{-Y + r^2 + ||a - b||^2 + 2||a - b|| \cdot Z}{2||c - b||r} \leq 100\gamma \tag{9}$$

with high probability. Let $\mathcal{E} = \{| - Y + r^2 + ||a - b||^2 + 2||a - b|| \cdot Z| < 100\gamma||c - b||r\}$.

By concentration results 1, 2, 3 above, we have

$$\begin{aligned}
\mathbf{Pr}[\bar{\mathcal{E}}] &\leq e^{-\gamma^2 n} + \mathbf{Pr}[| - Y + 2\gamma^2 n + 4\gamma\sqrt{n} \cdot Z| < 50\gamma^2 n] \\
&\leq e^{-\gamma^2 n} + \mathbf{Pr}[| - Y + 4\gamma\sqrt{n} \cdot Z| < 40\gamma^2 n] \\
&\leq e^{-\gamma^2 n} + \mathbf{Pr}[Y > 20\gamma^2 n] + \mathbf{Pr}[4\gamma\sqrt{n} \cdot |Z| > 20\gamma^2 n] \\
&\leq e^{-\gamma^2 n} + e^{-\Omega(\gamma^2 n)} + e^{-\Omega(\gamma^2 n)} = e^{-\Omega(\gamma^2 n)}.
\end{aligned} \tag{10}$$

We now get a proof of the last claim of the lemma by noting that

$$\ln \xi \geq (n/2) \ln \left(1 - \left(\frac{-Y + r^2 + ||a - b||^2 + 2||a - b|| \cdot Z}{2||c - b||r}\right)^2\right) - O(\ln n) \geq -O(\gamma)n,$$

where we used the bound $\ln(1 - x) \geq -(1 + O(\gamma))x$ for $x = O(\gamma)$.

We condition on $\mathcal{E}$ in what follows, so we have

$$\begin{aligned}
\xi &\geq \frac{C'}{\sqrt{n}} \left(1 - \left(\frac{-Y + r^2 + ||a - b||^2 + 2||a - b|| \cdot Z}{2||c - b||r}\right)^2\right)^{n/2} \\
&\geq \frac{C'}{\sqrt{n}} \exp\left(-(1 + O(\gamma))(n/2) \left(\frac{-Y + r^2 + ||a - b||^2 + 2||a - b|| \cdot Z}{2||c - b||r}\right)^2\right).
\end{aligned}$$

Taking logarithms of both sides, we get

$$\begin{aligned}
\ln \xi &\geq -(1 + O(\gamma))(n/2) \left(\frac{-Y + r^2 + ||a - b||^2 + 2||a - b|| \cdot Z}{2||c - b||r}\right)^2 - O(\ln n) \\
&\geq -(1 + O(\gamma)) \frac{1}{8(1 + \gamma^2)\gamma^2 n} \left(-Y + r^2 + ||a - b||^2 + 2||a - b|| \cdot Z\right)^2 - O(\ln n).
\end{aligned} \tag{11}$$

By (11) we now have

$$\mathbf{E}[\ln \xi | \mathcal{E}] \geq -(1 + O(\gamma)) \frac{1}{8(1 + \gamma^2)\gamma^2 n} \mathbf{E}[\left(-Y + r^2 + ||a - b||^2 + 2||a - b|| \cdot Z\right)^2 | \mathcal{E}] - O(\ln n), \tag{12}$$

and thus it is sufficient to upper bound

$$\mathbf{E}[\left(-Y + r^2 + ||a - b||^2 + 2||a - b|| \cdot Z\right)^2 | \mathcal{E}] \leq \frac{1}{\mathbf{Pr}[\mathcal{E}]} \mathbf{E}[\left(-Y + r^2 + ||a - b||^2 + 2||a - b|| \cdot Z\right)^2]. \tag{13}$$

[2]This choice of δ is not unique in that any choice of $\delta = \Omega(\gamma)$ would have been sufficient to obtain the claimed result, with this choice affecting the precise form of the $o(1)$ term in the result of the lemma. We choose $\delta = \Theta(\gamma)$ to minimize the amount of notation.

For the latter quantity we have using independence of $Y, Z, a-b, r$

$$\begin{aligned}&\mathbf{E}_{Y,Z,a-b,r}[\left(-Y+r^2+||a-b||^2+2||a-b||\cdot Z\right)^2]\\&\leq \mathbf{E}[r^4+||a-b||^4+2r^2||a-b||^2]+o(r^4)\\&\leq 4r^4+o(r^4),\end{aligned}$$

where we used that $\gamma' \leq \gamma$. Substituting this into (12), using (13) and (10), we get

$$\mathbf{E}[\ln \xi|\mathcal{E}] \geq -\frac{1}{2}\gamma^2 n(1+o(1))$$

and $\mathbf{Pr}[\mathcal{E}] \geq 1-e^{-\gamma^2 n}$, as desired. ■

Proof of Lemma 17: By Lemma 9 and Lemma 10 one has

$$\xi \leq 2I\left(\frac{r^2+||a-c||^2-R^2}{2||a-c||}, r\right) = 2I\left(\frac{r^2-Y}{2||a-c||}, r\right) \leq 2\exp\left(-(n/2)\left(\frac{r^2-Y}{2r||a-c||}\right)^2\right)$$

as long as $Y < r^2$. If $Y > r^2$, we upper bound the probability of collision by 1. This corresponds to a negligible $e^{-r^2/2}$ term.

By Claim 21 one has $||a-c|| \in (1\pm O(\gamma))n$ with probability at least $1-e^{-\gamma^2 n}$. Also, by Claim 21 one has $r \in (1\pm O(\gamma))\gamma n > 8$ with probability at least $1-e^{-\gamma^2 n}$. We condition on this event (call it $\mathcal{E}$) in what follows. We get

$$\begin{aligned}\frac{1}{2}\int_0^{r^2}\exp\left(-\frac{n}{8r^2||a-c||^2}\left(r^2-y\right)^2\right)e^{-y/2}dy &\leq \frac{1}{2}\int_0^{r^2}\exp\left(-\frac{(1-O(\gamma))}{8r^2}\left(r^2-y\right)^2\right)e^{-y/2}dy\\&\leq \frac{1}{2}\int_0^{r^2}\exp\left(-\frac{(1-O(\gamma))}{8r^2}\left(r^2-y\right)^2\right)e^{-(1-\delta)y/2}dy\\&= \frac{1}{2}\int_0^{\infty}\exp\left(-\frac{1-O(\gamma)}{8r^2}\left(r^4+2r^2y+y^2\right)\right)dy\\&= \frac{1}{2}\int_0^{\infty}\exp\left(-\frac{1-O(\gamma)}{8r^2}\left(r^2+y\right)^2\right)dy\end{aligned}$$

We now show that the value of this integral is essentially upper bounded by the value of the integrand at the left endpoint, i.e. when $y=0$:

$$\begin{aligned}\int_0^{\infty}\exp\left(-\frac{1-O(\gamma)}{8r^2}\left(r^2+y\right)^2\right)dy &= \exp\left(-\frac{(1-O(\gamma))r^2}{8}\right)\int_0^{\infty}\exp\left(-\frac{1-O(\gamma)}{8r^2}\left[\left(r^2+y\right)^2-r^4\right]\right)dy\\&= \exp\left(-\frac{(1-O(\gamma))r^2}{8}\right)\int_0^{\infty}\exp\left(-\frac{1-O(\gamma)}{8r^2}\left[2yr^2+y^2\right]\right)dy\end{aligned}$$

We now show that the integral in the last line is bounded by a constant. First note that $r \geq 8$ by our conditioning. We have

$$\int_0^{\infty}\exp\left(-\frac{1-O(\gamma)}{8r^2}\left[2yr^2+y^2\right]\right)dy \leq \int_0^{\infty}\exp\left(-y\right)dy \leq 1$$

Since we conditioned on an event of probability $1-e^{-\gamma^2 n} < 9/10$ for sufficiently large $\gamma^2 n$, the result follows. ■

We now give a proof of Lemma 18. We restate the lemma here for convenience of the reader.

Lemma 18 *Let $\alpha \in [0,1]$ be a constant. Let $c > 1$ denote the desired approximation ratio, and suppose that $K = n^{\Theta(1)}$. Let $d_{near} = n^{-1/4}$. Let $p, q \in \mathbb{R}^d$ be a pair of far points, i.e. $||p-q||_2 \geq cd_{near}$. Consider an invocation of* HASHDATA$(p, \alpha, S, K, \{R_l\}_{l=1}^K, B, n, \{\mathcal{U}_l\}_{l=1}^K, d_{near})$ *and* QUERY$(q, \alpha, S, K, \{R_l\}_{l=1}^K, A, n, \{\mathcal{U}_l\}_{l=1}^K, d_{near})$.

Then for each $i \in [1:A], j \in [1:B]$ one has

$$\mathbf{Pr}[\mathbf{h}(Sp+u^i) = \mathbf{h}(Sq+v^j)] \leq e^{-(c^2+\alpha^2+(1-\alpha)^2)\cdot d_{near}^2(1-o(1))nK/8}.$$

Proof: Recall that $\mathbf{h}$ is a concatenation of K independent hash functions. Fix $l \in [1:K]$ and consider the l-th hash function. We will prove that for each $i \in [1:A], j \in [1:B]$ and each $l \in [1:K]$ one has

$$\mathbf{Pr}[\mathbf{h}_l(Sp+u^i) = \mathbf{h}_l(Sq+v^j)] \leq e^{-(c^2+\alpha^2+(1-\alpha)^2)(1-o(1))d_{near}^2 n/8}, \tag{14}$$

where $\mathbf{h}_l$ stands for the l-th component of l. The result will then follow by independence of $\mathbf{h}_l$ for different l. Let $a = S_l p + u_l^i, b = S_l q + v_l^j$. By Claim 12 we have

$$a - b \sim \gamma \cdot \mathcal{N}(0, I_n),$$

where $\gamma = \sqrt{||p-q||_2^2 + \alpha^2 d_{near}^2 + (1-\alpha)^2 d_{near}^2}$. We also have by Claim 12 that b is a uniformly random point in the ball $\mathbb{B}_r(a)$, where $r^2 \sim \gamma^2 \cdot \mathcal{D}$.

Let $\mathbb{B}_{R_l}(a) \cap \mathcal{W}_l = \{w_1, \ldots, w_Q\}$. Each w_q is uniformly random in $\mathbb{B}_{R_l}(a)$. Let w_q be the uniformly random element of $\mathbb{B}_{R_l}(a) \cap \mathcal{W}_l$ that a hashes to (by line 3 of BASICHASH). By Claim 13 $w_q - a \sim \mathcal{N}(0, I_n)$, so we are in the setting of Lemma 17, i.e. we are interested in upper bounding

$$\xi = \frac{|\mathbb{B}_r(a) \cap \mathbb{B}_{R_l}(w)|}{|\mathbb{B}_r(a)|}.$$

By Lemma 17 we have

$$\mathbf{E}[\xi] \leq 2\text{exp}\left(-\frac{1}{8}\gamma^2 n(1+O(\gamma))\right),$$

where the expectation is over the choice of centers $\mathcal{W}_l$, dimensionality reduction matrix S_l and perturbations u_l^i, v_l^j. Let $q \in [1:Q]$ denote the (uniformly random) center that a hashes to (by line 3 of BASICHASH). We have

$$\begin{aligned}\mathbf{Pr}[w_q \in \mathbb{B}_{R_l}(b) | w_q \in \mathbb{B}_{R_l}(a)] &\leq 2\text{exp}\left(-\frac{1}{8}\gamma^2 n(1+O(\gamma))\right) \\ &\leq 2\text{exp}\left(-\frac{1}{8}(c^2+\alpha^2+(1-\alpha)^2)d_{near}^2 n(1-o(1))\right)\end{aligned} \tag{15}$$

where the probability is over $\mathcal{W}_l, S_l, u_l^i, v_l^j$. This yields the required bound when $Q \neq 0$. It remains to note that if $\mathbb{B}_{R_l}(a) \cap \mathcal{W}_l = \emptyset$, then a is hashed to a uniformly random element from an arbitrarily large universe, so the collision probability can easily be made polynomially small in the number of input points by choosing the universe to be poly(N) size. Since $n = o(\log N)$ by assumption of the lemma, a $1/\text{poly}(N)$ term is smaller than the rhs of (15), and (14) follows by a union bound over these two cases. The result of the lemma now follows by independence of the hashing process for different $l = 1, \ldots, K$. ■

We now give a proof of Lemma 19. We restate the lemma here for convenience of the reader.

Lemma 19 *Let $\alpha \in [0,1]$ be a constant. Let $c > 1$ denote the desired approximation ratio, and suppose that $K = n^{\Theta(1)}$, $Kd_{near}^2 = n^{\Omega(1)}, d_{near}^2 n = n^{\Omega(1)}, n = \omega(1), n = o(\log N)$. Let $p, q \in \mathbb{R}^d$ be a pair of near points, i.e. $||p-q||_2 \leq d_{near}$. Consider an invocation of* HASHDATA$(p, \alpha, S, K, \{R_l\}_{l=1}^K, B, n, \{\mathcal{U}_l\}_{l=1}^K, d_{near})$ *and* QUERY$(q, \alpha, S, K, \{R_l\}_{l=1}^K, A, n, \{\mathcal{U}_l\}_{l=1}^K, d_{near})$. *Then*

$$\mathbf{Pr}[\exists i \in [1:A], j \in [1:B] \text{ s.t. } \mathbf{h}(Sp+u^i) = \mathbf{h}(Sq+v^j)] = 1 - o(1)$$

as long as

$$A \geq (C \log N)^{2K} e^{(1+o(1))\alpha^2 d_{near}^2 nK/2},$$

and

$$B \geq (C \log N)^{2K} e^{(1+o(1))(1-\alpha)^2 d_{near}^2 nK/2}.$$

If $\alpha = 0$ then setting $A = 1$ and B as above is sufficient. Similarly, if $\alpha = 1$, setting A as above and $B = 1$ is sufficient.

We prove Lemma 19 in a sequence of steps. We start by giving an outline of the proof, then prove every step formally and put them together below. We need to prove that if a sufficiently large number of perturbations $u^i, v^j, i = 1, \ldots, A, j = 1, \ldots, B$ are used, then at least one pair of perturbed points collides under $\mathbf{h}$, i.e.

$$\mathbf{Pr}[\exists i \in [1:A], j \in [1:B] \text{ s.t. } \mathbf{h}(Sp+u^i) = \mathbf{h}(Sq+v^j)] = 1 - o(1)$$

as long as A, B are sufficiently large. Here the probability is over the choice of the hash function $\mathbf{h}$ (which consists of dimensionality reduction matrices S_l and centers $\mathcal{W}_l$, $l = 1, \ldots, K$) as well as the choice of perturbations u^i, v^j. Also, each perturbed point choose a center in $\mathcal{W}$ to hash to in line 3 of BASICHASH independently uniformly at random. Our argument proceeds as follows. First, we define the point (see Fig. 1)

$$z = (1 - \alpha)p + \alpha q.$$

We then follow these three steps:

(1) Prove that if the number of perturbations B of the *query point* q is sufficiently large, then at least one of them will collide with z under $\mathbf{h}$:

$$\mathbf{Pr}[\exists j \in [1 : B] \text{ s.t. } \mathbf{h}(Sq + v^i) = \mathbf{h}(Sz)] = 1 - o(1).$$

(2) Prove that if the number of perturbations A of the *data point* p is sufficiently large, then at least one of them will collide with the z under $\mathbf{h}$:

$$\mathbf{Pr}[\exists i \in [1 : A] \text{ s.t. } \mathbf{h}(Sp + u^i) = \mathbf{h}(Sz)] = 1 - o(1).$$

(3) Conclude, using the union bound, that at least one perturbation of p is very likely to collide with at least one perturbation of q, obtaining the result.

In what follows we give the argument for **Step 1.** and **Step 2** (which are symmetric, so we will only give the details for **Step 1**). Thus, we are now interested in proving that

$$\mathbf{Pr}[\exists j \in [1 : B] \text{ s.t. } \mathbf{h}(Sq + v^j) = \mathbf{h}(Sz)] = 1 - o(1), \tag{16}$$

where the probability is over the choice of the hash function $\mathbf{h}$ (i.e. matrices S_l and centers $\mathcal{W}_l$, $l = 1, \ldots, K$), and the choice of perturbations v^j. Let $w_l \in \mathcal{W}_l$ denote the element that $S_l z$ hashes to, i.e. BASICHASH$(S_l z, R_l, n, \mathcal{W}_l)$ (this is well-defined if the ball of radius R_l around $S_l z$ contains a center in $\mathcal{W}_l$ – see below).

Our argument now proceeds in three steps:

Step 1a. Prove that with probability $1 - o(1)$ over the choice of the dimensionality reduction matrix S and centers $\mathcal{W}$ one has

$$\mathbf{Pr}_{v^j}[w_l \in \mathbb{B}_{R_l}(S_l q + v_l^j) \text{ for all } l = 1, \ldots, K] \geq e^{-K \frac{1}{2}(1-\alpha)^2 d_{near}^2 n(1+o(1))}.$$

Denote this event by $\mathcal{E}^a$. Note that this is by itself not sufficient for (16), since for each $l = 1, \ldots, K$ BASICHASH chooses a uniformly random center from $\mathcal{W}_l \cap \mathbb{B}_{R_l}(x)$ to output. However, for any fixed pair of points p, q if the event $\mathcal{E}^*(p, q)$ occurs (see (2)), then all balls $\mathbb{B}_{R_l}(S_l q + v^j)$ around perturbations of q and p contain at most $2C \log N$ centers. Thus, conditional on the high probability event $\mathcal{E}^*(p, q)$, a given perturbation $S_l q + v^j$ is reasonably likely (probability at least $1/(2C \log N)$) to get hashed to w_l after all. By independence of these perturbations, we should be able to argue that at least one of them will get hashed to w_l with overwhelming probability, as long as the number of trials is sufficiently large. In **Step 1b** below we argue that the number of independent trials is large, and in **Step 1c** below show that this number of trials is sufficient.

Step 1b. Prove that, conditional on $\mathcal{E}^a$, with probability $1 - o(1)$ over the choice of perturbations $v^j, j = 1, \ldots, B$ there exists a set $J \subseteq [B]$ of size at least $(2C \log N)^{2K}/2$ such that

$$w_l \in \mathbb{B}_{R_l}(S_l q + v_l^j) \text{ for all } l = 1, \ldots, K$$

and all $j \in J$. Call this event $\mathcal{E}^b$.

Step 1c. Prove that conditional on $\mathcal{E}^*(p, q) \wedge \mathcal{E}^a \wedge \mathcal{E}^b$, with probability $1 - o(1)$ over the choices of centers to hash to in BASICHASH there exists $j \in J$ such that $\mathbf{h}(Sq + v^j) = \mathbf{h}(Sz)$.

We now give the details of **Steps 1a-1c**.

Step 1a. is given by

Lemma 22 *Let $\alpha \in [0,1]$ be a constant. Let $c > 1$ denote the desired approximation ratio, and suppose that $K = n^{\Theta(1)}$, $Kd_{near}^2 = n^{\Omega(1)}, d_{near}^2 n = n^{\Omega(1)}, n = \omega(1), n = o(\log N)$. Let $p, q \in \mathbb{R}^d$ be a pair of near points, i.e. $||p-q||_2 \leq d_{near}$. Let $z = (1-\alpha)p + \alpha q$.*

Let $w_l \in \mathcal{W}_l$ denote the element that $S_l z$ hashes to, i.e. BASICHASH$(S_l z, R_l, n, \mathcal{W}_l)$. *Then with probability $1 - o(1)$ over the choice of the dimensionality reduction matrix S and centers $\mathcal{W}$ one has for all j*

$$\mathbf{Pr}_{v^j}[w_l \in \mathbb{B}_{R_l}(S_l q + v_l^j) \text{ for all } l = 1, \ldots, K] \geq e^{-K\frac{1}{2}(1-\alpha)^2 d_{near}^2 n(1+o(1))}.$$

Similarly, for all i

$$\mathbf{Pr}_{u^i}[w_l \in \mathbb{B}_{R_l}(S_l p + u_l^i) \text{ for all } l = 1, \ldots, K] \geq e^{-K\frac{1}{2}\alpha^2 d_{near}^2 n(1+o(1))}.$$

Proof: We prove the first claim (the proof of the second is analogous).

If $\alpha = 1$, then the claim is obvious, since $v_l^j = 0$ for all j and l. Thus, we assume that $\alpha \neq 1$ in what follows, i.e. α is bounded away from 1 since α is a constant. Fix $j \in [1:B]$, and let $v := v^j$ to simplify notation. First note that for any fixed S and $\mathcal{W}$ one has by independence of perturbations in different coordinates

$$\mathbf{Pr}_v[w_l \in \mathbb{B}_{R_l}(S_l q + v_l) \text{ for all } l = 1, \ldots, K] = \prod_{l=1}^{K} \mathbf{Pr}_{v_l}[w_l \in \mathbb{B}_{R_l}(S_l q + v_l)].$$

Recall that for each $l = 1, \ldots, K$ w_l is the center that $S_l z$ is hashed to. By Claim 13 we have $S_l z - w_l \sim \mathcal{N}(0, I_n)$. Also,

$$S_l z - S_l q \sim (1-\alpha)||p-q||_2 \mathcal{N}(0, I_n) \tag{17}$$

by 2-stability of the Gaussian distribution. Recall that $S_l q + v_l$ is a uniformly random point in $\mathbb{B}_{r_{q,l}}(S_l q)$, so the probability that w_l is at distance at most R from it is given by

$$\xi_l = \frac{|\mathbb{B}_{r_{p,l}}(S_l q) \cap \mathbb{B}_{R_l}(w_l)|}{|\mathbb{B}_{r_{p,l}}(S_l q)|}. \tag{18}$$

We are in the setting of Lemma 16, where we have $\gamma' = (1-\alpha)||p-q||_2$ by (17) and $\gamma = (1-\alpha)d_{near}$ by definition of $r_{p,l}$. Note that $\gamma' \leq \gamma$ as required by Lemma 16 since p and q are near points by assumption. Thus for each l there exists an event $\mathcal{E}_l^1$ with $\mathbf{Pr}[\mathcal{E}_l^1] \geq 1 - e^{-\Omega((1-\alpha)^2 d_{near}^2 n)}$ such that

$$\mathbf{E}[\ln \xi_l | \mathcal{E}_l^1] \geq -\frac{1}{2}(1-\alpha)^2 d_{near}^2 n(1+o(1))$$

and $|\ln \xi| \leq n$ conditional on $\mathcal{E}_l^1$. Let $\mathcal{E}^1 := \bigwedge_{l=1}^K \mathcal{E}_l^1$. Note that $\mathbf{Pr}[\bar{\mathcal{E}}_1] \leq K e^{-\Omega(\gamma^2 n)} = n^{O(1)} e^{-n^{\Omega(1)}} = o(1)$ (we used that $K = n^{\Theta(1)}$ and $d_{near}^2 n = n^{\Omega(1)}$ by the assumptions of the lemma). By Chernoff bounds we have for any $\epsilon > 0$

$$\begin{aligned} \mathbf{Pr}\left[\sum_{l=1}^{K} \ln \xi_l \in -(1\pm\epsilon)K\frac{1}{2}(1-\alpha)^2 d_{near}^2 n(1+o(1)) \,\middle|\, \mathcal{E}^1\right] &< e^{-\Omega(\epsilon^2 K \frac{1}{2}(1-\alpha)^2 d_{near}^2)} \\ &\leq e^{-\Omega(\epsilon^2 K(1-\alpha)^2 d_{near}^2)}. \end{aligned} \tag{19}$$

We have

$$K(1-\alpha)^2 d_{near}^2 = \omega(1) \tag{20}$$

by assumption of the lemma, so the rhs of (19) is $o(1)$. Now note that by (19) and (20) there exists a setting of $\epsilon = o(1)$ such that with probability $1 - o(1)$ over $\mathcal{W}, S$, conditional on $\mathcal{E}^1$

$$\sum_{l=1}^{K} \ln \xi_l \in -(1\pm\epsilon)K\frac{1}{2}(1-\alpha)^2 d_{near}^2 n(1+o(1)),$$

implying that with probability at least $1 - o(1)$ over the choice of the centers $\mathcal{W}$ and the dimensionality reduction matrix S one has for each j

$$\mathbf{Pr}_v[w_l \in \mathbb{B}_{R_l}(S_l q + v_l) \text{ for all } l = 1, \ldots, K | \mathcal{E}^1] = \prod_{l=1}^{K} \frac{|\mathbb{B}_{r_{p,l}}(S_l q) \cap \mathbb{B}_{R_l}(w_l)|}{|\mathbb{B}_{r_{p,l}}(S_l q)|} \geq e^{-K\frac{1}{2}(1-\alpha)^2 d_{near}^2 n(1+\epsilon)}$$

for some $\epsilon = o(1)$. Since $\mathbf{Pr}[\mathcal{E}^1] \geq 1 - o(1)$, this gives the claimed result. ■

We now give the formal argument for **Step 1b.**

Lemma 23 *Let $\alpha \in [0,1]$ be a constant. Let $c > 1$ denote the desired approximation ratio, and suppose that $K = n^{\Theta(1)}$, $Kd_{near}^2 = n^{\Omega(1)} = \omega(1), d_{near}n = n^{\Omega(1)}$. Let $p, q \in \mathbb{R}^d$ be a pair of near points, i.e. $||p-q||_2 \leq d_{near}$. Let $z = (1-\alpha)p + \alpha q$.*

Let $w_l \in \mathcal{W}_l$ denote the element that $S_l z$ hashes to, i.e. BASICHASH$(S_l z, R_l, n, \mathcal{W}_l)$. *Suppose that $B \geq (C \log N)^{2K} e^{(1+o(1))(1-\alpha)^2}$ and $S, \{\mathcal{U}_l\}$ are such that*

$$\mathbf{Pr}_{v^j}[w_l \in \mathbb{B}_{R_l}(S_l q + v_l^j) \text{ for all } l = 1, \ldots, K] \geq e^{-K\frac{1}{2}(1-\alpha)^2 d_{near}^2 n(1+o(1))}.$$

Then with probability $1 - o(1)$ over the choice of perturbations $v^j, j = 1, \ldots, B$ there exists a set $J \subseteq [B]$ of size at least $(2C \log N)^{2K}/2$ such that

$$w_l \in \mathbb{B}_{R_l}(S_l q + v_l^j) \text{ for all } l = 1, \ldots, K$$

and all $j \in J$. An analogous statement holds for perturbations of data points.

Proof: For each $j = 1, \ldots, B$ let $Y_j = 1$ if $w_l \in \mathbb{B}_{R_l}(S_l q + v_l^j)$ for all $l = 1, \ldots, K$ and 0 otherwise. Since the choices of v^j are independent for different j, we get that $\sum_{j=1}^B Y_j$ is a sum of independent Bernoulli 0/1 rv's with $\mathbf{E}[Y_j] \geq e^{-K\frac{1}{2}(1-\alpha)^2 d_{near}^2 n(1+\epsilon)}$ for some $\epsilon = o(1)$. Thus, $\mathbf{E}[\sum_{j=1}^B Y_j] \geq (2C \log N)^{2K}$, and by standard concentration inequalities

$$\mathbf{Pr}\left[\sum_{j=1}^B Y_j < (2C \log N)^{2K}/2\right] = o(1)$$

as required, where the probability is over the choice of perturbations v^j. ■

Step 1c. is provided by

Lemma 24 *Let $\alpha \in [0,1]$ be a constant. Let $c > 1$ denote the desired approximation ratio, and suppose that $K = n^{\Theta(1)}$, $Kd_{near}^2 = n^{\Omega(1)}, d_{near}^2 n = n^{\Omega(1)}, n = \omega(1), n = o(\log N)$. Let $p, q \in \mathbb{R}^d$ be a pair of near points, i.e. $||p-q||_2 \leq d_{near}$. Let $z = (1-\alpha)p + \alpha q$. Suppose that there exists a set $J \subseteq [B]$ of size at least $(2C \log N)^{2K}/2$ such that*

$$w_l \in \mathbb{B}_{R_l}(S_l q + v_l^j) \text{ for all } l = 1, \ldots, K$$

and all $j \in J$. Then with probability $1 - o(1)$ over the choice of center to hash to in line 3 of BASICHASH *there exists $j \in J$ such that $\mathbf{h}(Sq + v^j) = \mathbf{h}(Sz)$.*

Proof: Recall that if the event $\mathcal{E}^*(p,q)$ occurrs (see (2) and Claim 11), then for every perturbed point $S_l q + v_l^j$ one has

$$\left|\mathbb{B}_{R_l}(S_l q + v_l^j) \cap \mathcal{W}_l\right| \leq 2C \log N.$$

Thus, each perturbation $j \in J$ chooses w_l independently with probability at least $1/(2C \log N)$ for each $l = 1, \ldots, K$. By independence of these choices for different v_l^j's, at least one perturbation v^j, $j \in J$ is hashed to $\mathbf{h}(Sz)$ with probability at least

$$1 - (1 - (2C \log N)^{-K})^{(2C \log N)^{2K}} \geq 1 - e^{-\Omega((2C \log N)^K)} = 1 - o(1).$$

Since by Claim 11 one has $\mathbf{Pr}[\mathcal{E}^*(p,q)] \geq 1 - 1/N$, the result follows. ■

We can now get

Proof of Lemma 19: Follows by putting together Lemma 22, Lemma 23 and Lemma 24. ■

C Proof of Lemma 14

In this section we give a proof of Lemma 14. We use the notation $\mathbb{B}(0)$ for the unit ball in ℓ_2 norm. To estimate the intersection we will use the following results of [BGMN05].

Theorem 25 *[BGMN05] Let $X_i \sim \frac{1}{\sqrt{\pi}} e^{-x_i^2}$, and let $Y \sim e^{-y}$. Then*

$$\left(\frac{X_1}{(|X_1|^2 + \ldots + |X_n|^2 + Y)^{1/2}}, \ldots, \frac{X_n}{(|X_1|^2 + \ldots + |X_n|^2 + Y)^{1/2}} \right)$$

is uniformly distributed in $\mathbb{B}(0)$.

Theorem 26 ([RR91, SZ90]; see also [BGMN05], Theorem 2) *Let $X_i \sim \frac{1}{\sqrt{\pi}} e^{-|x_i|^2}$. Then the random vector*

$$\left(\frac{X_1}{(|X_1|^2 + \ldots + |X_n|^2)^{1/2}}, \ldots, \frac{X_n}{(|X_1|^2 + \ldots + |X_n|^2)^{1/2}} \right)$$

is independent of $(|X_1|^2 + \ldots + |X_n|^2)^{1/2}$.

In this section we derive an expression for a uniformly random point in $R \cdot \mathbb{B}(0)$, where $R^2 \sim \Gamma(n/2+1)$. By Theorem 25 sampling a uniformly random point from $R \cdot \mathbb{B}$ can be done as follows. Sample $X_1, \ldots, X_n \sim \frac{1}{\sqrt{\pi}} e^{-x^2}$, $Y \sim e^{-y}$ and $R^2 \sim \Gamma(n/2+1, 1)$ independently. Then

$$R \cdot \left(\frac{X_1}{(X_1^2 + \ldots + X_n^2 + Y)^{1/2}}, \ldots, \frac{X_n}{(X_1^2 + \ldots + X_n^2 + Y)^{1/2}} \right). \tag{21}$$

is a uniformly random point in $R \cdot \mathbb{B}_\epsilon(0)$. We now rewrite (21) as

$$\begin{aligned} & R \cdot \left(\frac{X_i}{(X_1^2 + \ldots + X_n^2 + Y)^{1/2}} \right)_{i=1}^n \\ &= \left(\frac{X_i}{(X_1^2 + \ldots + X_n^2)^{1/2}} \right)_{i=1}^n \cdot \frac{(X_1^2 + \ldots + X_n^2)^{1/2}}{(X_1^2 + \ldots + X_n^2 + Y)^{1/2}} \cdot R \\ &= \left(\frac{X_i}{(X_1^2 + \ldots + X_n^2)^{1/2}} \right)_{i=1}^n \cdot \frac{1}{(1 + Y/(X_1^2 + \ldots + X_n^2))^{1/2}} \cdot R \\ &= V \cdot \frac{1}{(1+Q)^{1/2}} \cdot R, \end{aligned} \tag{22}$$

where

$$V = \left(\frac{X_i}{(X_1^2 + \ldots + X_n^2)^{1/2}} \right)_{i=1}^n \in \mathbb{R}^n,$$

and

$$Q = Y/(X_1^2 + \ldots + X_n^2).$$

By Theorem 26 V is independent of $X_1^2 + \ldots + X_n^2$. In particular, since R is sampled independently of (V, Q), this implies that V, Q, R are independent. Let μ denote the distribution of Q. We now prove

Lemma 27 *Let $X_i \sim \frac{1}{\sqrt{\pi}} e^{-x^2}$, $i = 1, \ldots, n$. Let Y be exponential with mean 1. Let $R = (X_1^2 + \ldots + X_n^2 + Y)^{1/2}$. Then*

$$(X_1, \ldots, X_n)$$

is uniformly distributed in the ball $R \cdot \mathbb{B}(0)$.

Proof: We have

$$\begin{aligned} (X_i)_{i=1}^n &= \left(\frac{X_i}{(X_1^2 + \ldots + X_n^2)^{1/2}} \right)_{i=1}^n \cdot \left(\frac{X_1^2 + \ldots + X_n^2}{X_1^2 + \ldots + X_n^2 + Y} \right)^{1/2} \cdot (X_1^2 + \ldots + X_n^2 + Y)^{1/2} \\ &= \left(\frac{X_i}{X_1^2 + \ldots + X_n^2} \right)_{i=1}^n \cdot \left(\frac{1}{1 + Y/(X_1^2 + \ldots + X_n^2)} \right)^{1/2} \cdot (X_1^2 + \ldots + X_n^2 + Y)^{1/2} \\ &= V \cdot \left(\frac{1}{1+Q'} \right)^{1/2} \cdot R. \end{aligned} \tag{23}$$

Note that $Q' \sim \mu$ if we do not condition on R. Furthermore, $R^2 \sim \Gamma(n/2+1,1)$ by Claim 7 and the additivity property of the Γ distribution, so R has the correct distribution as well. Hence, it is sufficient to show that V, Q', R are independent. First, V is independent of $X_1^2 + \ldots + X_n^2$ by Theorem 26, and independent of Y by definition. Thus, since Q' is a function of $X_1^2 + \ldots + X_n^2$ and Y, V is independent of (Q', R). It remains to show that Q' is independent of R.

Let $Z^2 = X_1^2 + \ldots + X_n^2$. Note that $Z^2 \sim \Gamma(n/2, 1)$ and $Y \sim \Gamma(1,1)$. We now compute the distribution of Y/Z^2 conditional on $Y + Z^2 = R^2$:

$$\begin{aligned}
\mathbf{Pr}[Y/Z^2 \geq \alpha | Y + Z^2 = R^2] &= \frac{\int_{R^2\alpha/(1+\alpha)}^{R^2} e^{-y}(R^2-y)^{n/2-1}e^{-(R^2-y)}dy}{\int_0^{R^2} e^{-y}(R^2-y)^{n/2-1}e^{-(R^2-y)}dy} \\
&= \frac{\int_{R^2\alpha/(1+\alpha)}^{R^2} (R^2-y)^{n/2-1}dy}{\int_0^{R^2}(R^2-y)^{n/2-1}dy} \\
&= \frac{(R^2 - R^2\alpha/(1+\alpha))^{n/2}}{(R^2)^{n/2}} = (1+\alpha)^{-n/2},
\end{aligned} \tag{24}$$

which is independent of R. Thus, V, Q', R are independent, which completes the proof. ■

Proof of Lemma 14: Follows by Lemma 27 and Claim 7. ■