# A Refined Analysis of LSH for Well-dispersed Data Points*

Wenlong Mou[†] Liwei Wang[‡]

## Abstract

Near neighbor problems are fundamental in algorithms for high-dimensional Euclidean spaces. While classical approaches suffer from the curse of dimensionality, locality sensitive hashing (LSH) can effectively solve $\alpha$-approximate $r$-near neighbor problem, and has been proven to be optimal in the worst case. However, for real-world data sets, LSH can naturally benefit from well-dispersed data and low doubling dimension, leading to significantly improved performance.

In this paper, we address this issue and propose a refined analyses for running time of approximating near neighbors queries via LSH. We characterize dispersion of data using $N_\beta$, the number of $\beta r$-near pairs among the data points. Combined with optimal data-oblivious LSH scheme, we get a $O\left(\left(1+\frac{4\sqrt{2}\alpha}{\beta}\right)^{\frac{d}{2\alpha^2}}(n+N_\beta)^{\frac{1}{2\alpha^2}}\right)$ bound for expected query time. For many natural scenarios where points are well-dispersed or lying in a low-doubling-dimension space, our result leads to sharper performance than existing worst-case analysis. This paper not only presents the *first* rigorous proof on how LSHs make use of the structure of data points, but also provides important insights into parameter setting in the practice of LSH beyond worst case. Besides, the techniques in our analysis involve a generalized version of sphere packing problem, which might be of some independent interest.

## 1 Introduction

Near neighbor search is a fundamental problems in metric spaces, and is playing an increasingly important role in databases [17], machine learning [16] and computer vision [7]. With the large-scale data set, we usually need a data structure with sub-linear query time, which only visits a small portion of candidates. There are many classical results on near neighbor search in Euclidean spaces with fixed dimensions. Voronoi diagrams partition the space based on nearest neighbors, but the computation of which is formidable for high-dimensional spaces. Tree structures such as k-d trees [8] and VP trees [19] are effective for low dimensional spaces. However, those data structures usually suffer heavily from the curse of dimensionality, and could not be put into practical use for the emerging large-scale data sets. There are also many works on generalizing those tree structures to high dimensional cases via low-distortion dimensionality reduction [1, 12] and doubling dimensions [10].

Though it seems hard to derive deterministic algorithms to the exact problems, randomization and approximation allow effective solutions. The approximate randomized formulation was proposed in [11, 2] as follows:

**Randomized $\alpha$-approximate $r$-near neighbor** Given $S = \{x_1, x_2, \dots, x_n\} \subset \mathbb{R}^d$, $r \in \mathbb{R}^+$, $\alpha > 1, \delta \in (0,1)$, construct a data structure such that given any query point $x_0$, if there is some $x_i \in S$ s.t. $\|x_0 - x_i\|_2 \le r$, then we can report a point $x_j \in S$ s.t. $\|x_0 - x_j\|_2 \le \alpha r$ with probability at least $1-\delta$.

One of the most successful solutions to this problem was locality sensitive hashing(LSH), which constructs a distribution on hash functions, and makes near neighbors more likely to be hashed into the same bucket. In each query, we only need to visit the buckets where query point have ever been put. So we only need to visit a small proportion of data points in total, and that will guarantee essentially sub-linear query time. Basically, the more sharply collision probability decreases with distance, the better performance will be obtained. Let $p_1$ be the collision probability for points at distance $r$, and $p_2$ be that for points at distance $\alpha r$. The key performance measure for LSH is $\rho = \frac{\log p_1}{\log p_2}$, since $\alpha$-approximate $r$-near neighbor will be solved within query $O(n^\rho)$, with proper parametric setting. The LSHs for Euclidean spaces have been intensively studied, and a series of hashing schemes were proposed with improving $\rho$ parameter [9, 3, 4, 5].

Despite their success, however, the classical analyses for locality sensitive hashing methods are not always optimal: they simply treat all distant points as the same. One can see from a simple geometric intuition,

*Supported by grant xxx

[†] Key Laboratory of Machine Perception, School of EECS, Peking University. Email: mouwenlong@pku.edu.cn

[‡] Key Laboratory of Machine Perception, School of EECS, Peking University. Email: wanglw@cis.pku.edu.cn

that when the number of near pairs is controlled, then the data points have to be dispersed. In this paper, we quantify this idea and present a refined analysis for LSH. Concretely, we introduce a new parameter as $N_\beta = \big|\{(x_i, x_j) : \|x_i - x_j\|_2 \le \beta r\}\big|$ and show how the performance of LSH can be improved if $N_\beta$ increases slowly with $\beta$. In our analyses, we require a slightly stronger uniform LSH condition, namely, the collision probability bound to hold uniformly for any $\alpha > 1$. It is easy to see that this condition is satisfied by all existing data-independent LSHs. To characterize the points that are more than $\alpha r$ far from $x_0$, we propose a generalized version of the sphere packing problem and give an upper bound. We proved that for any $\beta > 0$, every uniform LSH with $\rho = \rho(\alpha)$ for $\mathbb{R}^d$ guarantees a $O\left(\left(1 + O(1)\cdot\frac{\alpha}{\beta}\right)^{\frac{d\rho(\alpha)}{2}}(n + N_\beta)^{\frac{\rho(\alpha)}{2}}\right)$ query time, by setting the parameters properly. We also show that our dimensionality-dependent analysis for LSH can be easily generalized to the case of doubling metrics, at a loss of constant factor. Compared with classical analyses of LSH, our bounds achieved essentially better performance in two cases: (i) when the dimensionality or the doubling dimension is low (ii) when $N_\beta$ remains at $O(n)$ for some large constant $\beta$. In addition to a good explanation on how LSH works better than worst-case analyses on real-world data, this bound can provide more insights for the choice of parameters for LSH. To the best of our knowledge, this is the first theoretical explanation on how LSHs exploit the intrinsic characteristics of data points.

## 2 LSH for near neighbor problems

In this section, we will give a brief introduction to the formulation of LSH and query algorithms. A precise analysis on the expected query time based on uniform LSH will be presented. We will also summarize the existing works on LSHs for Euclidean spaces.

LSH is formulated as a distribution over hashing functions, for which the probability of collision increases as two points get closer. In the original formulation of LSH, parameter $\alpha$ and $r$ are fixed, and only the collision probability for $\|x - y\|_2 \le r$ and $\|x - y\|_2 \ge \alpha r$ is considered. A key property in the evaluation of LSH is $\rho = \frac{\log p_1}{\log p_2}$, which depicts how sharply the collision probability decreases with distance. As discussed in [3, 4], we usually need to make some tradeoff between computational cost of $h(\cdot)$ itself and accuracy of LSH. So the actual $\rho$-parameter for LSH is often written as $\tilde{\rho}(\alpha) + o(1)$, where the residual term $o(1)$ term diminishes as $n \to \infty$, though the ratio between log probability $\tilde{\rho}$ for ideal hashing class is independent of $n$. We will also use this notation.

**Locality sensitive hashing** : A distribution $\mathcal{H}$ is called $\langle r, \alpha r, p_1, p_2\rangle$-sensitive if for $\forall x, y \in \mathbb{R}^d$:

- $\forall p, q \in \mathbb{R}^d, \|p - q\|_2 \le r$ we have $Pr_{h\sim\mathcal{H}}\{h(p) = h(q)\} \ge p_1$
- $\forall p, q \in \mathbb{R}^d, \|p - q\|_2 \ge \alpha r$ we have $Pr_{h\sim\mathcal{H}}\{h(p) = h(q)\} \le p_2$

We slightly strengthen the requirements for LSH in our analysis, where the collision probability gap should be guaranteed uniformly for $\forall\alpha > 1$. Fortunately, all the known data-independent LSH do not depend upon parameter $\alpha$ and are suitable for this formulation. In the rest of this paper, we will denote uniform locality sensitive hashings using the term LSH, except when it is specified as data-dependent.

**Uniform locality sensitive hashing** A distribution $\mathcal{H}$ is called uniformly $\langle r, \rho = \rho(s)\rangle$-sensitive if for $\forall x, y \in \mathbb{R}^d$ satisfies:
$p(s) \triangleq Pr_{h\sim\mathcal{H}}\big\{h(x) = h(y)\big|\|x - y\|_2 = s \cdot r\big\}$ is a monotonic decreasing function of $s \in (1, +\infty)$ and $\rho(s) = \frac{\log p(1)}{\log p(s)}$.

The general algorithm framework for LSH was proposed in [11, 2], as described in Algorithm 1.

**Algorithm 1** Framework for near neighbor approximation

```
Input: $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$,   $\alpha > 1$,   $r \in R^+$
Parameters: $K, L$
Preprocessing:
Sample $L \cdot K$ functions $h_{11}, h_{12}, \ldots, h_{1k}, h_{21}, \ldots, h_{2k}, \ldots, h_{LK} \sim i.i.d.\mathcal{H}$
and let $g_i = (h_{i1}, h_{i2}, \ldots, h_{iK})$,   $\forall i = 1, 2, \ldots. L$.
Construct $L$ hash tables with respect to $\{g_i\}_{i=1}^{L}$.
Query:
Given query point $x_0$:
For $i = 1, 2, \ldots, L$:
    • Compute $g_i(x_0)$ and locate the hashing bucket.
    • Traverse the elements in bucket and compute the actual distances, if any $\alpha r$ near neighbor is found, report it and stop.
```

The parameter $\langle K, L\rangle$ will be chosen carefully to optimize the performance while guaranteeing success probability. If there is a neighbor $x^*$ within distance $r$, to guarantee $\alpha r$-near approximation, we just need to visit $x^*$ with $(1 - \delta)$ probability:

$$\underset{g\sim\mathcal{H}^{KL}}{Pr}\{\forall g_i, g_i(x_0) \neq g_i(x^*)\} \le \left(1 - p(1)^K\right)^L \le \delta \tag{2.1}$$

So it is sufficient to set the number of rounds run for each query as $L = p(1)^{-K} \log \frac{1}{\delta}$.

Consider the expected number of points visited in each round of hashing computation. If a point at distance not more than $\alpha r$ was ever visited, the algorithm would stop and report this point. So at most one point at distance within $\alpha r$ was visited in each round, and other visited points are located farther than $\alpha r$. Let the set of points visited in the $i$-th round be $S_i$.

$$\begin{aligned} &\underset{h\sim\mathcal{H}}{E}\left[|S_i|\right] \\ \leq & 1 + \sum_{\|x_i - x_0\|_2 \geq \alpha r} \underset{h\sim\mathcal{H}}{Pr}\left\{h(x_i) = h(x_0)\right\}^K \\ = & 1 + \sum_{\|x_i - x_0\|_2 \geq \alpha r} \exp\left\{\frac{K \cdot \log p(1)}{\rho\left(\frac{\|x_i - x_0\|_2}{r} + o(1)\right)}\right\} \end{aligned} \tag{2.2}$$

In the classical analysis for locality sensitive hashing, we usually relax the inequality by plugging in $\|x_i - x_0\|_2 \geq \alpha r$, and get:

$$\underset{h\sim\mathcal{H}}{E}\left[|S_i|\right] \leq 1 + n \cdot p(1)^{\frac{K}{\rho(\alpha)+o(1)}} \tag{2.3}$$

However, in many cases, this relaxation cannot be tight simultaneously for points in $S$, and some geometric constraints will force some of the points farther away from $x_0$. This phenomenon is the main focus of this paper, and will be analyzed in detail in the next sections. Here we just use this relaxation and go through the main results by classical analysis of LSH.

By putting all the above together, we can bound the expectation of running time for near neighbor approximation with: (Let $\tau$ be the time required for computing LSH)

$$T(n) = O\left(p(1)^{-K}\left(1 + n \cdot p(1)^{\frac{K}{\rho(\alpha)+o(1)}}\right)\tau \log \frac{1}{\delta}\right) \tag{2.4}$$

By choosing the optimal parameter $K$, we get an upper bound for the query running time: $T_{query}(n) = O\left(n^{\rho(\alpha)+o(1)}\tau \log \frac{1}{\delta}\right)$ and the corresponding preprocessing time can be upper bounded with $T_{preprocess}(n) = O\left(n^{1+\rho(\alpha)+o(1)}\tau \log \frac{1}{\delta}\right)$.

Locality sensitive hashing and near neighbor problem have been intensively studied in existing literature. In [9] a class of locality sensitive hashing was firstly proposed. They obtained a $\rho(\alpha) = \frac{1}{\alpha}$ performance by projecting the data points to a calibrated real line. Later, a significant improvement was done in [3]. They first perform random projection and reduce to a low-dimensional space, then the hashing buckets were constructed using random grids of balls. By setting appropriate parameters, they achieved the performance $\rho(\alpha) = \frac{1}{\alpha^2} + o(1)$, asymptotically. As shown in [15], this bound is essentially optimal in the worst case.

Recently, there is also a series of works on data-dependent locality sensitive hashing. Andoni et al., [4] first introduced a class of data dependent hashing class, and improved the $\rho$ parameter to $\frac{7}{8\alpha^2} + \frac{O(1)}{\alpha^3}$). In [5] it was further improved to $\frac{1}{2\alpha^2 - 1} + o(1)$, which is proven to be optimal in [6]. Unfortunately, since their construction of hashing schemes depend on parameter $\alpha$, they could not be generalized to the uniform LSH case, and those bounds are not suitable for our refined analyses.

## 3 Generalized Sphere Packing Problem

In the classical analysis of locality sensitive hashing, we relax the estimation for query time by assuming all the data points visited but not accepted in a query are just $\alpha r$ far away from the query point. But this is not usually true in reality: if most of the data points approximately on a sphere with radius $\alpha r$ centered at $x_0$, then the sphere will be "crowded" and there will be many pairs of near neighbors within the data set. In reality, however, most data points in $S$ are far from each other. For parameter $\beta > 1$, a fixed constant that is not very large, we will have $N_\beta = \left|\{(x_i, x_j) : \|x_i - x_j\|_2 \leq \beta r\}\right| \ll n^2$. So in our refined analysis, we seek to bound the running time of algorithm in terms of not only $n = |S|$ but also $N_\beta$, where $\beta$ is a parameter used for minimize the bound depending on the structure of data. In the following analysis, we make use of a generalized version of famous sphere packing problem: we want to characterize the phenomenon that a set of points must be well-dispersed if there are only a few near pairs.

An intuitive view to this problem is to consider the "worst case", where the two data points either coincide, or be closely packed in $\mathbb{R}^d$ with distance at least $\beta r$. We construct a graph where two vertices are linked if their corresponding points are $\beta r$ near neighbors. Roughly speaking, we want to show that by adjusting the configuration of points into the "worst-case", we will shrink the space those points take without increasing the number of near pairs. The following lemma gives us a quantitative description of this "worst-case" intuition.

Lemma 3.1. *For an undirected graph $G = \langle V, E\rangle$, with $|V| = n$. There is a subset of vertices $T \subset V$ and a mapping $\phi : V \to T$, such that*

- $\forall u \in T, \phi(u) = u$ *and* $\forall v \in V - T, (v, \phi(v)) \in E$
- $\forall u, v \in T, \quad (u, v) \notin E$
- *For $u \in T$, let $n_u = \left|\{v \in V : \phi(v) = u\}\right|$, then we have:*

$$\sum_{u \in T} \binom{n_u}{2} \leq |E| \tag{3.5}$$

*Proof.* Without loss of generality, we assume the vertices $v_1, v_2, \ldots, v_n$ are sorted in increasing order by degree, i.e., $d(v_1) \leq d(v_2) \leq \ldots \leq d(v_n)$. We can construct $T$ and $\phi$ in the following way:

```
**Algorithm 2** Construction of $T$ and $\phi$
**Initialization**: $T = \varnothing$
**Processing**: For $i = 1, 2, \ldots, n$
  • If $\exists u \in T$, s.t. $(u, v_i) \in E$, then we let
      (3.6)   $\phi(v_i) = \underset{u \in T, (u, v_i) \in E}{argmin} \; d(u)$
  • If such $u$ does not exist, we let $T = T \cup \{v_i\}$ and $\phi(v_i) = v_i$
```

According to the construction above, we can guarantee the first two requirements in the lemma. Furthermore, since the degrees are sorted in increasing order, we have $d(\phi(u)) \geq d(u), \forall u \in V$. Thus

$$\begin{aligned} |E| =& \frac{1}{2} \sum_{v \in V} d(v) = \frac{1}{2} \sum_{u \in T} \left( \sum_{\phi(v) = u} d(v) \right) \\ \geq& \frac{1}{2} \sum_{u \in T} d(u) \left( d(u) + 1 \right) \\ \geq& \sum_{u \in T} \binom{n_u}{2} \end{aligned} \tag{3.7}$$

(The algorithm described in the lemma is designed in assistance to the theoretical analysis, and does not need to be actually run.)

With the lemma in hand, we are ready to handle the $N_\beta$ near pairs in the space, based on which we can lower bound the distance to farther points.

Theorem 3.1. *For points $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$, $N_\beta = \left| \{(x_i, x_j) : \|x_i - x_j\|_2 \leq \beta r\} \right|$. Then $\forall x_0 \in \mathbb{R}^d$, we have:*

$$\underset{1 \leq i \leq n}{max} \|x_i - x_0\|_2 \geq \frac{1}{2} \left( \left( \frac{n^2}{2N_\beta + n} \right)^{\frac{1}{d}} - 1 \right) \beta r \tag{3.8}$$

*Proof.* We construct $G = \langle V, E \rangle$ with $V = \{v_1, v_2, \ldots, v_n\}$ for any pair of vertices $v_i, v_j \in V$, we set $(v_i, v_j) \in E$ if and if only $\|x_i - x_j\|_2 \leq \beta r$. According to Lemma 3.1, we have set $T = \{v_{i_1}, v_{i_2}, \ldots v_{i_t}\}$ and mapping $\phi$ for this graph. Let the points in $\mathbb{R}^d$ corresponding to $T$ be $Q = \{x_{i_1}, x_{i_2}, \ldots x_{i_t}\}$. By definition we know every two points in $Q$ have distance at least $\beta r$.

According to Cauchy-Schwartz inequality, we have:

$$\begin{aligned} n^2 = \left( \sum_{u \in T} n_u \right)^2 \leq& \left( \sum_{u \in T} n_u^2 \right) \left( \sum_{u \in T} 1 \right) \\ =& \left| T \right| \cdot \left( n + 2 \sum_{u \in T} \binom{n_u}{2} \right) \\ \leq& \left| T \right| (n + 2 \left| E \right|) \end{aligned} \tag{3.9}$$

So we have $\left| Q \right| = \left| T \right| \geq \frac{n^2}{n + 2N_\beta}$.

Let $r^* = max_{1 \leq i \leq n} \|x_i - x_0\|_2$, consider a ball $B_0 = B(x_0, r^* + \frac{\beta r}{2})$, centered at $x_0$ with radius $r^* + \frac{\beta r}{2}$. And for each $\forall x_i, 1 \leq i \leq n$, let $d$-dimensional ball $B_i = B(x_i, \frac{\beta r}{2})$. Since $\|x_i - x_0\|_2 \leq r^*$, we have $\bigcup_{i=1}^n B_i \subset B_0$. On the other hand, since the points in $Q$ are at least $\beta r$ distant from each other, we have $B_i \cap B_j = \varnothing, \forall x_i, x_j \in Q, i \neq j$. Thus we have:

$$\begin{aligned} & \frac{\pi^{\frac{d}{2}}}{\Gamma\left(1 + \frac{d}{2}\right)} \left( r^* + \frac{\beta r}{2} \right)^d \\ =& Vol \left( B \left( x_0, r^* + \frac{\beta r}{2} \right) \right) \\ \geq& Vol \left( \bigcup_{x' \in Q} B(x', \frac{\beta r}{2}) \right) \\ =& \sum_{x' \in Q} Vol \left( B(x', \frac{\beta r}{2}) \right) \\ =& \left| Q \right| \cdot \frac{\pi^{\frac{d}{2}}}{\Gamma\left(1 + \frac{d}{2}\right)} \left( \frac{\beta r}{2} \right)^d \end{aligned} \tag{3.10}$$

By plugging in the lower bound for $\left| Q \right|$ we get:

$$r^* \geq \left( \left( \frac{n^2}{n + 2N_\beta} \right)^{\frac{1}{d}} - 1 \right) \frac{\beta r}{2} \tag{3.11}$$

This bound is informative only for relatively low dimensionality. Otherwise, for example, if $d = \omega(logn)$, we will have $\lim_{n \to +\infty} \left( \frac{n^2}{n + 2N_\beta} \right)^{\frac{1}{d}} = 1$, and the bound converges to zero. Actually, the geometric structure of $\mathbb{R}^d$ can tell us little information when $d = \omega(\log n)$ and no other constraints are posed. Indeed, the classical analysis for LSH is tight for this case. On the other hand, there are still much we can do in high dimensions: by standard Johnson-Lindenstrauss argument we can restrict dimensionality at the order of $O(\log n)$ while preserving locality; we can also replace the dimensionality of space with doubling dimension, as discussed in Section 5.

## 4 Dimension-dependent Refined Analysis for LSH

In section 3 we have already proposed a tighter estimation for the distance from query point to the farthest point. This result can be applied in the analysis of locality sensitive hashing, and yield an essentially sharper bound for it.

Given points $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$, we can build hash tables using Algorithm 1, whose parameters $K, L$ will be set to optimize the bound later on. When a query point arrives, as the analysis in Section 2, we have:

$$\underset{h\sim\mathcal{H}}{E}[|S_i|] \leq 1 + \sum_{\|x_i - x_0\|_2 \geq \alpha r} \exp\left\{ \frac{K \cdot \log p(1)}{\rho\left(\frac{\|x_i - x_0\|_2}{r}\right)} \right\} \tag{4.12}$$

Based on the sphere packing results, we are ready to upper bound the value of big summation above.

Lemma 4.1. *Given $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$, fixed parameters $\alpha \geq 1, \beta > 0, \eta > 0, p \in (0,1)$, $\rho(\cdot)$ be a monotonic increasing function on $(1, +\infty)$, and let $N_\beta = \left|\{(i,j) : \|x_i - x_j\|_2 \leq \beta r\}\right|$, then we have:*

$$\begin{aligned} &\sum_{\|x_i - x_0\|_2 \geq \alpha r} \exp\left\{ \frac{\log p}{\rho\left(\frac{\|x_i - x_0\|_2}{r}\right)} \right\} \\ \leq & p^{\frac{1}{\rho(\alpha)}} \left(1 + \frac{2(\alpha+\eta)}{\beta}\right)^{\frac{d}{2}} \sqrt{2N_\beta + n} + p^{\frac{1}{\rho(\alpha+\eta)}} n \end{aligned} \tag{4.13}$$

In estimating the summation, classical analyses roughly divide the points according to their distances from query point, at threshold $\alpha r$. In Lemma 4.1, we divide further at threshold $(\alpha + \eta)r$ to get more accurate bounds. The detailed proof for this lemma is deferred to Appendix. We will see from the later analysis that this bound could not be asymptotically improved by more precise dividing, since the points outside $(\alpha + \eta)r$ distance only make tiny contribution to the sum.

By applying the standard strategies for the parameter setting in LSH, here follows our dimension-dependent query time bound, the detailed proof is deferred to Appendix.

Theorem 4.1. *(Main theorem for real dimension)*

*For $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$, if we use a uniform locality sensitive hashing $\mathcal{H}$ with $\rho = \rho(\alpha) + o(1)$ and $\tau$ computation cost to solve $\alpha$-approximate $r$-near neighbor problem with probability $1 - \delta$, then $\forall \beta > 0$, with proper parameter selection, the expected query time is upper bounded by:*

$$O\left( \left(1 + \frac{2\mu}{\beta}\right)^{\frac{d}{2}\rho(\alpha)} (N_\beta + n)^{\frac{1}{2}\rho(\alpha) + o(1)} \tau \log \frac{1}{\delta} \right) \tag{4.14}$$

*where $\mu$ satisfies $\rho(\mu) = \frac{1}{2}\rho(\alpha)$ for monotonic decreasing function $\rho(\cdot)$*

We apply this theorem to the two famous locality sensitive hashing schemes. Since their constructions do not involve parameter $\alpha$ to be known in advance, they actually satisfy the uniform locality sensitive property proposed in Section 2.

Proposition 4.1. *For the line projection LSH in [11], we have $\rho(\alpha) = \frac{1}{\alpha}$, the expected running time is bounded with:*

$$O\left( d \left(1 + \frac{4\alpha}{\beta}\right)^{\frac{d}{2\alpha}} (N_\beta + n)^{\frac{1}{2\alpha}} \log \frac{1}{\delta} \right), \forall \beta > 0 \tag{4.15}$$

Proposition 4.2. *For the random grid of ball LSH in [3], we have $\rho(\alpha) = \frac{1}{\alpha^2} + o(1)$, the expected running time is bounded with:*

$$O\left( d \left(1 + \frac{2\sqrt{2}\alpha}{\beta}\right)^{\frac{d}{2\alpha^2}} (N_\beta + n)^{\frac{1}{2\alpha^2} + o(1)} \log \frac{1}{\delta} \right), \forall \beta > 0 \tag{4.16}$$

This bound on expectation holds uniformly for $\forall \beta > 0$, and the optimal $\beta$ can be selected to minimize the bound. Actually, if the points are well-dispersed, i.e., $N_\beta$ increases slowly with $\beta$, this bound leads to significantly lower running time for LSH. The classical analysis can be seen as a special case of this dimensionality-dependent analysis, for we have:

$$\lim_{\beta \to +\infty} \left(1 + \frac{2\mu}{\beta}\right)^{\frac{d}{2}\rho(\alpha)} (N_\beta + n)^{\frac{1}{2}\rho(\alpha) + o(1)} = n^{\rho(\alpha) + o(1)} \tag{4.17}$$

Despite its sharpness, the major drawback of this bound is that it depends exponentially on the data dimensionality, if the parameter $\alpha$ and $\beta$ are constants. We will explain in the next section on how to overcome it, by further exploiting intrinsic structure of the points.

## 5 Doubling Metric Counterparts

Since the analyses in previous sections are based upon the point sets' rate of expansion, instead of their real dimensionality, it would naturally generalize to the case of dimensionality intrinsic in the data and provide a better bound. In this section, we extend our analysis to doubling dimension case. On the one hand, doubling dimension can appropriately capture the phenomena that high-dimensional data are usually lying approximately on a low-dimensional manifold; on the other hand, bounds based on doubling dimension are also applicable to more general metric spaces.

The notion of doubling dimension has been studied in a wide range of literature. There are several different definitions for doubling dimension[18, 14, 13]. They are equivalent except for an absolute constant factor. Here we adopt the definition in [14], for it not only adapts more naturally to our problem, but also generalizes to metrics other than Euclidean distances.

**Doubling dimension** The doubling dimension of a metric $X$ is the minimal $d_0$ such that $\forall Y \subseteq X$, there exists a series of subsets $\{Y_i\}_{i=1}^{2^{d_0}}$, such that $Y \subseteq \bigcup_{i=1}^{2^{d_0}} Y_i$ and

$$(5.18)\qquad \underset{y,y'\in Y_i}{max} \|y-y'\|_X \leq \frac{1}{2} \underset{y,y'\in Y}{max} \|y-y'\|_X \qquad \forall i = 1, 2, \dots 2^{d_0}$$

The following lemma from [14] characterizes the packing properties for metrics with doubling dimensions, and are widely applied in various problems:

Lemma 5.1. *For a metric $X$ with doubling dimension $d_0$, then for any finite subset $Y \subseteq X$ we have*

$$(5.19)\qquad \lceil \log \frac{max_{y,y'\in Y} \|y-y'\|_2}{min_{y,y'\in Y} \|y-y'\|_2} \rceil \geq \frac{1}{d_0} \log |Y|$$

This helps us to establish the doubling-dimension version of Theorem 3.1.

Lemma 5.2. *For points $x_1, x_2, \dots, x_n \in \mathbb{R}^d$ with doubling dimension $d_0$, for $\beta > 0$, let $N_\beta = \big|\{(x_i, x_j) : \|x_i - x_j\|_2 \leq \beta r\}\big|$. Then $\forall x_0 \in \mathbb{R}^d$, we have:*

(5.20)

$$\underset{1\leq i\leq n}{max} \|x_i - x_0\|_2 \geq \frac{1}{4}\left[\left(\frac{n^2}{2N_\beta + 2n}\right)^{\frac{1}{d_0+1}} - 1\right]\beta r$$

By plugging this result into the proof of Theorem 4.1, we get:

Theorem 5.1. *(Query time bound for doubling dimension)*

*For $x_1, x_2, \dots, x_n \in \mathbb{R}^d$ with doubling dimension $d_0$, if we use a uniform locality sensitive hashing $\mathcal{H}$ with $\rho = \rho(\alpha) + o(1)$ and $\tau$ computation cost to solve $\alpha$-approximate $r$-near neighbor problem with probability $1-\delta$, then $\forall \beta > 0$, with proper parameter selection, the expected query time is upper bounded by:*

(5.21)

$$O\left(\left(1 + \frac{4\mu}{\beta}\right)^{\frac{d_0+1}{2}\rho(\alpha)} (N_\beta + n)^{\frac{1}{2}\rho(\alpha)+o(1)} \tau \log\frac{1}{\delta}\right)$$

*where $\mu$ satisfies $\rho(\mu) = \frac{1}{2}\rho(\alpha)$ for monotonic decreasing $\rho(\cdot)$*

Similar results as Proposition 4.1 and Proposition 4.2 also holds for doubling dimension. Furthermore, Theorem 5.1 also implies improved bounds for well-dispersed points in general metric spaces. For example, by applying Theorem 5.1 to LSH based on $p$-stable distribution [9], we can get the following proposition for general $\ell_p$(the $p$-stable distribution guarantees for $p \in (0, 2]$, and we need $p \geq 1$ to guarantee the triangle inequality):

Proposition 5.1. $\forall p \in [1, 2]$, *consider the $p$-stable distribution LSH in [9] which solves approximate near neighbor in $\ell_p$ metric. Assuming the doubling dimension of $\langle X, \|\cdot\|_p\rangle$ to be $d_0$, we have $\rho(\alpha) = \frac{1}{\alpha} + o(1)$, $\forall \beta > 0$, with proper parameter selection, the expected query time is bounded with:*

$$(5.22)\qquad O\left(d\left(1 + \frac{8\alpha}{\beta}\right)^{\frac{d_0+1}{2\alpha}} (N_\beta + n)^{\frac{1}{2\alpha}+o(1)} \log\frac{1}{\delta}\right)$$

## 6 When do our bounds work?

In this section, we will discuss the parameters in our bounds and show that, in many natural scenarios, where data points are well-dispersed or doubling dimension is low, are bound can significantly improve over classical worst-case bounds for LSH.

In our analyses, the parameter $\beta$ should be chosen carefully in order to minimize our bounds. We are particularly interested in the case where $N_\beta$ is approximately at the same order as $n$. Specifically, for a small constant $\epsilon > 0$, let $C_\epsilon(n) = \sup\{\beta : N_\beta < n^{1+\epsilon}\}$. Well-dispersed data will result in larger value of $C_\epsilon$.

In the low-dimensional case, namely, $d = o(\log n)$, we use the bound in Theorem 4.1; otherwise we will turn to doubling dimension bound in Theorem 5.1. For a finite metric space, we have $d_0 \leq \log n$ by definition. Let $d_0 = \xi \log n$ be the doubling dimension of data set, with $\xi \in (0, 1]$. Combined with optimal data-independent LSH, we get the following expected query time bound:

(6.23)

$$\frac{\log E[T_{query}]}{\log n} \leq \frac{1}{2\alpha^2}\left(1 + \epsilon + \xi \log\left(1 + \frac{4\sqrt{2}\alpha}{C_\epsilon(n)}\right)\right) + o(1)$$

Our bounds are informative in the following scenarios:

- Data points are significantly well-dispersed, i.e., $\lim\limits_{n\to\infty} C_\epsilon(n) = \infty$ for some small $\epsilon$. For this case we have $E[T_{query}] \leq n^{\frac{1+\epsilon}{2\alpha^2}+o(1)}$, since $\xi$ is bounded by one. An example for this condition is a set of sparse vectors in high dimensions, i.e. $d = poly(n)$, where the dimension gets higher when $n$ gets larger, and the data points become more dispersed.

- Spaces with significantly low (doubling) dimensions, i.e., $\lim_{n\to\infty} \xi = 0$. For this case our bound becomes $E[T_{query}] \leq n^{\frac{1}{2\alpha^2}+o(1)}$. Though the bound is still polynomial in $n$, it implies that LSH can automatically adapt to low-dimensional space or doubling metrics.

- For general case, we assume $C$ and $\xi$ are both $\Theta(1)$. The quality of our bound depends on whether $1 + \frac{4\sqrt{2}\alpha}{C} < 2^{\frac{1}{\xi}}$ holds. Larger $C$ and smaller $\xi$ will be helpful.

For the above three cases, our analysis is uniformly sharper than all existing data-oblivious LSH results. It is also sharper than the best data-dependent LSH in the first two cases. Interestingly, our results guarantee sub-linear query time even when $\alpha = 1$, while all existing worst-case bounds for LSH query time are informative only for $\alpha > 1$. This result implies that LSH can take advantage of well-dispersed data, even if we want to perform exact $r$-near neighbor search.

Furthermore, our analysis shows that the parameter choice designed for worst-case performance can be suboptimal in practice, when the data points satisfy dispersion or doubling dimension structure. Specifically, since the collision probabilities are usually overestimated, bucket sizes can be smaller than needed, namely, $K$ is often set too large in the worst-case-optimal setting. To make use of our bounds in practice, we may incorporate empirical estimates for $N_\beta$ and prior knowledge about doubling dimension, and plug into the choice of $K$ in Theorem 4.1.

## 7 Conclusion and Open Questions

In this paper, we present a refined analysis for query time of approximate near neighbor via LSH, given well-dispersed data points. Though the previous analyses are tight for worst case, they could not explain how LSHs work better in real data sets where some structures are assumed. We address this issue by introducing $N_\beta$, the number of $\beta r$-near pairs, to describe the dispersion of data points. Using a generalized version of sphere packing argument, we present an $O\left(\left(1+\frac{2\mu}{\beta}\right)^{\frac{d}{2}\rho(\alpha)} (N_\beta + n)^{\frac{1}{2}\rho(\alpha)+o(1)}\right)$ upper bound for expected query time of LSH, based on proper choice of parameters. We also generalize this result to the case of doubling dimension, and obtained an $O\left(\left(1+\frac{4\mu}{\beta}\right)^{\frac{d_0+1}{2}\rho(\alpha)} (N_\beta + n)^{\frac{1}{2}\rho(\alpha)+o(1)}\right)$ bound. Compared with other existing results, these bounds make essential improvements when the data points are dispersed well or have low doubling dimensions.

There are still many problems on explaining the performance of LSHs that are left open:

- In our analyses, the relaxation of inequalities are actually loose, in terms of the constant factor before $\frac{\alpha}{\beta}$. We intend to give a tighter bound in the future, so that the dependence on dispersion parameters can be further relaxed.

- The data-dependent schemes[5] could not be applied to our analyses directly, since they are not uniform. A more precise analyses on the geometric structures of high-dimensional data sets will shed lights on data-dependent LSHs.

- We will also seek quantities other than $N_\beta$ that describes characteristics and structures of data points, which can explain the performance of near neighbor algorithms.

## References

[1] Evangelos Anagnostopoulos, Ioannis Z. Emiris, and Ioannis Psarros. Low-quality dimension reduction and high-dimensional approximate nearest neighbor. In *31st International Symposium on Computational Geometry, SoCG 2015, June 22-25, 2015, Eindhoven, The Netherlands*, pages 436–450, 2015.

[2] Alexandr Andoni. *Nearest Neighbor Search: the Old, the New, and the Impossible.* PhD thesis, MIT, 2009.

[3] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 459–468, 2006.

[4] Alexandr Andoni, Piotr Indyk, Huy L. Nguyen, and Ilya Razenshteyn. Beyond locality-sensitive hashing. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1018–1028, 2014.

[5] Alexandr Andoni and Ilya Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 793–801, 2015.

[6] Alexandr Andoni and Ilya P. Razenshteyn. Tight lower bounds for data-dependent locality-sensitive hashing. *CoRR*, abs/1507.04299, 2015.

[7] Nir Ben-Zrihem and Lihi Zelnik-Manor. Approximate nearest neighbor fields in video. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 5233–5242, 2015.

[8] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.

[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, Brooklyn, New York, USA, June 8-11, 2004*, pages 253–262, 2004.

[10] Lee-Ad Gottlieb and Robert Krauthgamer. Proximity algorithms for nearly-doubling spaces. In Maria Serna, Ronen Shaltiel, Klaus Jansen, and Jos Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 6302 of *Lecture Notes in Computer Science*, pages 192–204. Springer Berlin Heidelberg, 2010.

[11] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 604–613, 1998.

[12] Piotr Indyk and Assaf Naor. Nearest-neighbor-preserving embeddings. *ACM Transactions on Algorithms*, 3(3), 2007.

[13] David R. Karger and Matthias Ruhl. Finding nearest neighbors in growth-restricted metrics. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 741–750, 2002.

[14] Robert Krauthgamer and James R. Lee. Navigating nets: simple algorithms for proximity search. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11-14, 2004*, pages 798–807, 2004.

[15] Ryan O'Donnell, Yi Wu, and Yuan Zhou. Optimal lower bounds for locality-sensitive hashing (except when q is tiny). *TOCT*, 6(1):5, 2014.

[16] Anshumali Shrivastava and Ping Li. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2321–2329, 2014.

[17] Yifang Sun, Wei Wang, Jianbin Qin, Ying Zhang, and Xuemin Lin. SRS: solving c-approximate nearest neighbor queries in high dimensional euclidean space with a tiny index. *PVLDB*, 8(1):1–12, 2014.

[18] Kunal Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 281–290, 2004.

[19] Peter N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 25-27 January 1993, Austin, Texas.*, pages 311–321, 1993.

## Appendix: Deferred Proofs

*Proof.* (Proof of Lemma 4.1) Let $x_{p(1)}, x_{p(2)}, \dots, x_{p(n)}$ be a permutation of $x_1, \dots, x_n$ s.t. $\left\{\|x_{p(i)} - x_0\|_2\right\}_{i=1}^{n}$ is sorted in ascending order. For $\forall k \in \{1, 2, \dots, n\}$ and $\beta > 0$, apparently we have

$$\left|\left\{(i,j) : \|x_{p(i)} - x_{p(j)}\|_2 \le \beta r, 1 \le i < j \le k\right\}\right| \le N_\beta \tag{7.24}$$

Plugging it into Theorem 3.1, we have:

$$\begin{aligned}\|x_{p(k)} - x_0\|_2 &= \underset{1\le i\le k}{max} \|x_{p(i)} - x_0\|_2 \\ &\ge \left(\left(\frac{k^2}{k+2N_\beta}\right)^{\frac{1}{d}} - 1\right)\frac{\beta r}{2} \\ &\ge \left(\left(\frac{k^2}{n+2N_\beta}\right)^{\frac{1}{d}} - 1\right)\frac{\beta r}{2}\end{aligned} \tag{7.25}$$

Let $k_0 = max\left\{k : \|x_{p(k)} - x_0\|_2 \le (\alpha+\eta)r\right\}$, according to the inequality above, we have:

$$\begin{aligned}&\left(\left(\frac{k_0^2}{n+2N_\beta}\right)^{\frac{1}{d}} - 1\right)\frac{\beta r}{2} \le (\alpha+\eta)r \\ \Rightarrow &k_0 \le \left(1 + \frac{2(\alpha+\eta)}{\beta}\right)^{\frac{d}{2}}\sqrt{n+2N_\beta}\end{aligned} \tag{7.26}$$

Then we can easily lower bound $\|x_{p(i)} - x_0\|_2$ with $\alpha r$ for $i \le k_0$ and bound with $(\alpha+\eta)r$ for $i > k_0$, and get the following result:

$$\begin{aligned}&\sum_{\|x_i - x_0\|_2 \ge \alpha r} \exp\left\{\frac{\log p}{\rho\left(\frac{\|x_i - x_0\|_2}{r}\right)}\right\} \\ \le &p^{\frac{1}{\rho(\alpha)}}\left(1 + \frac{2(\alpha+\eta)}{\beta}\right)^{\frac{d}{2}}\sqrt{2N_\beta + n} + p^{\frac{1}{\rho(\alpha+\eta)}}\end{aligned} \tag{7.27}$$

*Proof.* (Proof of Theorem 4.1) We use the algorithmic framework described in Algorithm 1, the parameters $K, L$ will be later set to optimize the bound.

Let $p(s) = Pr_{h\sim\mathcal{H}}\left\{h(x) = h(y) \big| \|x-y\|_2 = sr\right\}$ be the collision probability induced by $\mathcal{H}$. As discussed in Section 1, it suffices to set $L = \frac{1}{p(1)^K}\log\frac{1}{\delta}$ to guarantee the collision probability. Then the expected query time is bounded with:

$$\frac{T(n)}{\tau\log\frac{1}{\delta}} \le \frac{1}{p(1)^K}\left(1 + \sum_{\|x_i - x_0\|_2 \ge \alpha r} \exp\left\{\frac{K \cdot \log p(1)}{\rho\left(\frac{\|x_i - x_0\|_2}{r}\right)}\right\}\right) \tag{7.28}$$

By applying Lemma 4.1 with $p = p(1)^K$, and let $M = \left(1 + \frac{2(\alpha+\eta)}{\beta}\right)^{\frac{d}{2}} \sqrt{2N_\beta + n}$, we get:

$$\begin{aligned}\frac{T(n)}{\tau \log\frac{1}{\delta}} \le p(1)^{-K} &+ p(1)^{K(\frac{1}{\rho(\alpha)+o(1)}-1)}M \\ &+ p(1)^{K(\frac{1}{\rho(\alpha+\eta)+o(1)}-1)}n\end{aligned} \tag{7.29}$$

We set the parameters as follows:

$$K = -\frac{\rho(\alpha)\log M}{\log p(1)}, \quad \eta = \mu - \alpha \tag{7.30}$$

From our choice of parameters, we have $p(1)^K = \frac{1}{M^{\rho(\alpha)}}$, and

$$M \cdot p(1)^{K\left(\frac{1}{\rho(\alpha)+o(1)}-1\right)} = M^{\rho(\alpha)+o(1)} \tag{7.31}$$

$$\begin{aligned}&n \cdot p(1)^{K\left(\frac{1}{\rho(\alpha+\eta)+o(1)}-1\right)} \\ =&nM^{\rho(\alpha)-\frac{\rho(\alpha)}{\rho(\mu)}+o(1)} \le M^{\rho(\alpha)+o(1)}\end{aligned} \tag{7.32}$$

Thus the query time is upper bounded by $O(M^{\rho(\alpha)+o(1)}\tau\log\frac{1}{\delta})$.

*Proof.* (Proof of Lemma 5.1)
By definition we can construct a series of subsets $\{Y_i\}_{i=1}^{2^{d_0}}$ satisfying the doubling metric condition, such that $Y \subseteq \bigcup_{i=1}^{2^{d_0}} Y_i$. Apparently we have:

$$\left|Y\right| \le \sum_{i=1}^{2^{d_0}} \left|Y_i\right| \tag{7.33}$$

Each $Y_i$ will be a subset with at most half of the radius. We can then recursively divide each $Y_i$ until there are at most 2 points contained in each set, and by simple calculation we get this result.

*Proof.* (Proof of Lemma 5.2)
For $\beta > \frac{max\|x_i - x_j\|_2}{r}$, the inequality is trivial since RHS becomes negative.

For $\beta \in \left(0, \frac{max\|x_i - x_j\|_2}{r}\right)$, let $X = \{x_0, x_1, \ldots, x_n\}$. By adding just one point, the doubling dimension will not exceed $d_0 + 1$. Since adding $x_0$ to the point set will increase the number of near pairs up to $n$, we have $\left|\{(i,j) : 0 \le i < j \le n : \|x_i - x_j\|_2 \le \beta r\}\right| \le N_\beta + n$. According to Lemma 3.1, we obtain a set of points $T$ with $\left|T\right| \ge \frac{n^2}{2n+2N_\beta}$ and $\forall x, x' \in T, \|x - x'\|_2 \ge \beta r$. The doubling dimension of $T$ does not exceed $d_0 + 1$. By applying Lemma 5.1 to $T$, we get:

$$\begin{aligned}&\frac{1}{d_0+1}\log\left|T\right| \\ \le&\lceil\log\frac{max_{y,y'\in T}\|y - y'\|_2}{\beta r}\rceil \\ \le&1 + \log\frac{max_{y,y'\in T}\|y - y'\|_2}{\beta r}\end{aligned} \tag{7.34}$$

Thus we have:

$$\max_{y,y'\in T}\|y - y'\|_2 \ge \frac{\beta r}{2}\left|T\right|^{\frac{1}{d_0+1}} \ge \left(\frac{n^2}{2n + 2N_\beta}\right)^{\frac{1}{d_0+1}}\frac{\beta r}{2} \tag{7.35}$$

On the other hand, according to the triangle inequality, we have

$$2\max_{1\le i\le n}\|x_i - x_0\|_2 \ge \max_{y,y'\in T}\|y - y'\|_2 \tag{7.36}$$

By putting them together, we get:

$$\begin{aligned}&\max_{1\le i\le n}\|x_i - x_0\|_2 \\ \ge&\frac{1}{4}\left[\left(\frac{n^2}{2N_\beta + 2n}\right)^{\frac{1}{d_0+1}}\right]\beta r \\ \ge&\frac{1}{4}\left[\left(\frac{n^2}{2N_\beta + 2n}\right)^{\frac{1}{d_0+1}} - 1\right]\beta r\end{aligned} \tag{7.37}$$