Randomly-oriented k-d Trees Adapt to Intrinsic Dimension

Santosh S. Vempala*

School of Computer Science
Georgia Tech
vempala@gatech.edu

Abstract

The classic k-d tree data structure continues to be widely used in spite of its vulnerability to the so-called curse of dimensionality. Here we provide a rigorous explanation: for randomly rotated data, a k-d tree adapts to the intrinsic dimension of the data and is not affected by the ambient dimension, thus keeping the data structure efficient for objects such as low-dimensional manifolds and sparse data. The main insight of the analysis can be used as an algorithmic pre-processing step to realize the same benefit: rotate the data randomly; then build a k-d tree. Our work can be seen as a refinement of *Random Projection trees* [7], which also adapt to intrinsic dimension but incur higher traversal costs as the resulting cells are polyhedra and not cuboids. Using k-d trees after a random rotation results in cells that are cuboids, thus preserving the traversal efficiency of standard k-d trees.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, E.1 Data Structures, G.3 Probability and Statistics

Keywords and phrases Data structures, Nearest Neighbors, Intrinsic Dimension, k-d Tree

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2012.48

1 Introduction

The k-d tree, introduced by Bentley [4], is a classic data structure for nearest neighbor search. Roughly speaking, a k-d tree is constructed by recursively partitioning space using axis-parallel cuts, with each cut placed at the median of the point set along some axis. It is widely used in machine learning, computer vision, bioinformatics, astronomy and other fields.

For the ubiquitous nearest neighbor problem, k-d trees are the method of choice. Although other more sophisticated algorithms have been proposed in the nearly forty years since the invention of k-d trees, they remain the default approach to nearest neighbor problems in a variety of settings. They are space efficient, being only linear in size with respect to the number of points, and they are easy to construct and query.

Although they are so popular, k-d trees do have a major weakness. As the dimension n becomes large, in the worst case, nearest neighbor queries can take time close to linear in the total number of points (a manifestation of the "curse of dimensionality"). Thus our current state of knowledge is that k-d trees are an efficient heuristic approach in low dimension (without precise knowledge of running times) and their performance can degrade significantly as the dimension increases.

To overcome the the challenge of high dimensionality, researchers have designed other data structures for nearest neighbor search. These include tree-based structures such as

* Supported in part by NSF award AF-0915903

32nd Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012).
Editors: D. D'Souza, J. Radhakrishnan, and K. Telikepalli; pp. 48–57

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

approximate Voronoi diagrams [12], cover trees [5], PCA trees [19] and navigating nets [17]. Another algorithmic solution is based on random projection [18], inspired by the Johnson-Lindenstrauss Lemma [15]. This idea was subsequently developed by applying a series of random projections together as *locality-sensitive* hash functions [14, 13, 1], leading to the strongest known upper bounds for *approximate* nearest neighbor search. Roughly speaking, these bounds allow for fixed polynomial (slightly superlinear) storage space and sublinear query time, in return for constant-factor approximations to the nearest neighbor distance.

A different line of research has focused on exploiting some form of low-dimensional structure in the data, e.g., cover trees [5] and random projection trees [7, 10]. The idea here is that the ambient dimension is not the critical factor in the complexity, but rather some much smaller quantity that corresponds to the intrinsic dimension of the data set. An important notion of intrinsic dimension is the *Assouad* (or *doubling*) dimension [11], a slight variant of a concept defined by Assouad [3]. Let $B(x, r)$ be the closed ball of radius r centered at x.

▶ **Definition 1.** The *doubling dimension* of a set $S \subseteq \mathbb{R}^n$ is the smallest integer d s.t. for every $x \in \mathbb{R}^n$ and $r > 0$, $B(x, r) \cap S$ can be covered by at most 2^d balls of radius $r/2$.

An interval in $\mathbb{R}$ has doubling dimension 1. Any subset of $\mathbb{R}^k$ has doubling dimension $O(k)$; so subsets restricted to k-dimensional subspaces in $\mathbb{R}^n$ have doubling dimension $O(k)$. The union over $1 \leq i \leq n$ of the intervals $[-e_i, e_i]$ has doubling dimension $\log n$ (where $e_1, \ldots, e_n$ are the canonical unit vectors). This sparse example can be generalized — if every point in $S \subset \mathbb{R}^n$ has at most d nonzero coordinates, then the doubling dimension of S is $O(d \log n)$ [6].

Dasgupta and Freund's random projection trees (RP trees) are built as follows: pick a random direction at every partitioning step, independently for each cell, and split the current cell at a random point within a small interval of the median of the current data points. These trees have the property that the diameter of cells in the data structure decreases quickly with the number of splits – it takes roughly $O(d \log d)$ splits to halve the diameter where d is the intrinsic dimension. Dasgupta and Freund termed this behavior "adapting to Assouad dimension". Subsequently, the RP tree has found applications in other settings including tree-based vector quantization [8] and regression [16]. The random directions used to build an RP tree are not orthogonal to each other and at each level of the tree, there are many different cuts used, leading to a data structure whose cells are general polyhedra rather than cuboids as in standard k-d trees. One advantage of the standard k-d tree that is lost here is that traversing the tree only needs comparison on a single coordinate at a time, while for RP trees this goes up by at least a factor of n (since one has to compare along a random direction). Moreover, computing the distance of a point to a cell, which is now a general polyhedron, is substantially more complicated. Indeed, their original paper does not give a nearest neighbor algorithm.

On the other hand, k-d trees do not have a nice dependence on doubling dimension while RP trees do. This is seen in the example of a points distributed along n orthogonal lines, one parallel to each axis. In this example, halving the diameter requires n splits, i.e., depth n, even though the doubling dimension is only $\log(2n)$.

In this paper, we propose a conceptually simple and algorithmically efficient variant of k-d trees that adapts to intrinsic dimension. In fact, our algorithm is essentially a pre-processing step for a k-d tree. The preprocessing consists of a *random rotation* of the ambient space, i.e., instead of the standard basis for constructing a tree, we use a random basis. The overhead in the running time could be negligible as the database would be rotated in advance, and a query point only has to be mapped once to the chosen basis before the search is carried out.

Our main theorem asserts that such a transformation leads to a strong guarantee for k-d trees, namely that they adapt to the intrinsic dimension, in the same way that RP trees do. The Randomized k-d tree algorithm is described precisely in the next section.

▶ **Theorem 2** (Main). *Let $S \subset \mathbb{R}^n$ be a finite set with m points and doubling dimension d. Assuming $d \log d \leq c_0 n$, for the Randomized k-d tree, with probability at least $1 - me^{-c_1 n}$, for any cell C of the tree and every cell C' that is at least $c_2 d \log d$ levels below C, we have*

$$\text{diam}(C' \cap S) \leq \frac{1}{2}\text{diam}(C \cap S)$$

where c_0, c_1, c_2 are absolute constants.

In other words, when the doubling dimension is low, it takes only a small number of rounds of splits to halve the diameter of cells. A stronger guarantee would be to give an actual bound on the query time based on doubling dimension. However, we are not aware of such a connection between cell diameter and query times.

This theorem also provides an intriguing perspective on the standard use of k-d trees showing that simply taking a random rotation of the data could yield a configuration of the dataset more amenable to nearest neighbor search via k-d trees. Alternatively, one can also view this result as an explanation of the success of k-d trees, namely if one assumes that the basis in which measurements are made is essentially random, then these trees adapt to the intrinsic dimension, i.e., they work well on average, if one views the data as coming from a randomly chosen basis. Both these perspectives are supportive of the idea that k-d trees are actually an excellent choice whenever the intrinsic dimension is significantly lower than the ambient dimension.

We note that this is a technically simple paper, based heavily on techniques from the literature. The main obstacle we overcome in the analysis is that the splits used in our method are not independent, unlike RP trees where each splitting direction is chosen independently of all others. As far as we know, the simple idea of a random rotation in advance does provide the first reasonable explanation of the performance of k-d trees with increasing dimension on real data sets (as they might have low Assouad dimension). Moreover, the insight of the analysis can be made algorithmic: rotate the data randomly; then build a k-d tree. It remains to be seen whether this pre-processing step is useful in practice.

2 Algorithm

As mentioned in the introduction, our algorithm is the following: we pick a random orthogonal basis for space and then build a k-d tree using this basis. The only change we make is that we make is that instead of splitting at the median when we partition a cell, we split at a random point in an interval around the median (this modification was also used by Dasgupta and Freund [7]). It is conceivable that perturbation near the median can be avoided by adding some random points to the data before building the tree; we do not explore this here.

3 Analysis

3.1 Outline

Our goal is to show that any subset S of bounded diameter will be partitioned into cells of at most half the diameter within $O(d \log d)$ levels of partitioning applied to the subset. To prove this, we first cover S with balls of significantly smaller diameter, then show that

Randomized k-d Tree.
1. Pick a random basis $V = \{v_1, \ldots v_n\}$ of $\mathbb{R}^n$.
2. Run KD-Tree$(S, V, 1)$.

KD-Tree(S, V, i)**.**
- If $|S| = 1$, return S.
  1. Let 2Δ be the diameter of S.
  2. Let m be the median of S along v_i and δ be uniform random in $\left[-\frac{6\Delta}{\sqrt{n}}, \frac{6\Delta}{\sqrt{n}}\right]$.
  3. $S^- = \{x \in S : \langle x, v_i \rangle \leq m + \delta\}$; $S^+ = S \setminus S^-$.
  4. $T^- =$ KD-Tree$(S^-, V, i \mod n + 1)$; $T^+ =$ KD-Tree$(S^+, V, i \mod n + 1)$.
  5. Return $[T^-, T^+]$.

Figure 1 Randomized k-d Tree Algorithm

with good probability, our partitioning procedure separates any pair of balls that are far enough part into different cells within $O(d \log d)$ levels of partitioning. The cells obtained at the end of this process will have the claimed diameter bound. Dasgupta and Freund use random independent splits for each cell, and a union bound for the failure probability. In our case, the splits come from a single basis and the same split direction is applied to all cells at one level, so we have to analyze the resulting conditioning and dependencies. RP trees pick a completely random direction to make the next split, our trees pick the next vector in the random basis. To argue that the latter achieves similar performance, we observe that a random basis can be chosen by picking a random unit vector, then a random unit vector orthogonal to it, and so on, each time picking a random unit vector orthogonal to the span of the vectors chosen so far. Our analysis idea is to consider the projection orthogonal to all basis vectors used for cuts so far and argue that this projection does not collapse points or shrink balls too much as long as not too many vectors have been chosen. Once we condition on this, a random vector in the remaining subspace is almost as good as a random vector in the full space.

It is, however, necessary that we incur some dependence on the number of points, since we are picking only a fixed basis, i.e., the total randomness is bounded. We could set up a large enough point set such that for any chosen basis, eventually we reach a cell that takes much more than $O(d \log d)$ cuts to halve in diameter. We get around this issue by assuming that the total number of points is at most exponential in the ambient dimension, i.e., at most 2^{cn} for some constant c.

3.2 Preliminaries

Our main tool is the Johnson-Lindenstrauss Lemma [15]. For a subspace V of $\mathbb{R}^n$, let $\pi_V(.)$ denote orthogonal projection to the subspace V. We will use the following version from [2, 20] (see also [9, 13]).

▶ **Lemma 3.** *Fix a unit vector $u \in \mathbb{R}^n$, let V be a random k dimensional subspace where $k < n$, and $\epsilon > 0$ then:*

$$\Pr\left(\|\pi_V(u)\|^2 > (1+\epsilon)\frac{k}{n}\right) \leq e^{-\frac{k}{4}(\epsilon^2-\epsilon^3)}$$

$$\Pr\left(\|\pi_V(u)\|^2 < (1-\epsilon)\frac{k}{n}\right) \leq e^{-\frac{k}{4}(\epsilon^2-\epsilon^3)}.$$

As a direct implication, for any finite set of points S in $\mathbb{R}^n$, with probability at least

$$1 - 2\binom{|S|}{2} e^{-\frac{k}{4}(\epsilon^2-\epsilon^3)},$$

we have

$$\forall u, v \in S, (1-\epsilon)\frac{k}{n}\|u-v\|^2 < \|\pi_V(u-v)\|^2 < (1+\epsilon)\frac{k}{n}\|u-v\|^2.$$

We also use the following standard bounds for $k = 1$.

▶ **Lemma 4.** *Let $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^n$ be a random unit vector. For any $\beta > 0$,*

$$\begin{aligned}
\mathsf{Pr}\left(\|\pi_v(u)\| > \frac{\beta}{\sqrt{n}}\|u\|\right) &\leq \frac{2}{\beta}e^{-\frac{\beta^2}{2}} \\
\mathsf{Pr}\left(\|\pi_v(u)\| \leq \frac{\beta}{\sqrt{n}}\|u\|\right) &\leq \alpha\sqrt{\frac{2}{\pi}}.
\end{aligned}$$

3.3 Projection properties

The next lemma is a structural property that uses the doubling dimension, and is similar to what was shown in [7] for RP trees.

▶ **Lemma 5.** *Let $S \subset B(x, r)$ be a set of doubling dimension d. Let V be an arbitrary k-dimensional subspace of $\mathbb{R}^n$, v be a random unit vector orthogonal to V, $1 > \delta > 0$ and*

$$r' = \frac{3r}{\sqrt{n-k}}\sqrt{2\left(d + \log(2/\delta)\right)}$$

Then $\pi_v(S) \subseteq [\pi_v(x) - r', \pi_v(x) + r']$ with probability at least $1 - \delta$.

Proof. We consider a projection orthogonal to the given subspace V first, then a projection to a random vector in this subspace $W = V^\perp$. Since W has sufficiently large dimension, this will be nearly as good as projecting to a random vector in the full space.

Let C_1 be a minimum cover of S consisting of balls of radius $r/2$. From the definition of doubling dimension, C_1 has at most 2^d elements. Similarly C_2 will be a cover of $C_1 \cap S$ with balls of radius $r/4$; at level i, C_i will be cover of $C_{i-1} \cap S$ using at most 2^d balls of radius $r/2^i$ for each element of C_{i-1}.

Fix a ball $B(c, r/2^i)$ at level i and consider one of the balls, $B(c', r/2^{i+1})$, which covers it. Let $W = V^\perp$. We have for c and c', the center of these balls that

$$\|c - c'\| \leq \frac{r}{2^i}.$$

Next we compute the following:

$$\begin{aligned}
\mathsf{Pr}\left(\|\pi_v(c-c')\| \geq \beta\frac{r}{2^i}\sqrt{\frac{i+1}{n-k}}\right) &\leq \mathsf{Pr}\left(\|\pi_v(c-c')\| \geq \beta\frac{\|\pi_W(c-c')\|}{\sqrt{n-k}}\sqrt{i+1}\right) \\
&= \mathsf{Pr}\left(\|\pi_v(\pi_W(c-c'))\| \geq \beta\frac{\|\pi_W(c-c')\|}{\sqrt{n-k}}\sqrt{i+1}\right) \\
&\leq \frac{2}{\beta\sqrt{i+1}}e^{-\frac{\beta^2}{2}(i+1)} \\
&\leq \frac{\delta}{\beta}\left(\frac{\delta}{2}\right)^i e^{-d(i+1)}
\end{aligned}$$

where $\beta = \sqrt{2(d + \log(2/\delta))}$. Now we take a union bound over all balls used in covers at all levels. For level i, there are $2^{id}2^d = 2^{(i+1)d}$ pairs we need to consider. Thus, via a standard chaining argument a la Dudley,

$$\begin{aligned}\Pr\left(\exists c, c' \text{st } \|\pi_v(c-c')\| \geq \beta\frac{r}{2^i}\sqrt{\frac{i+1}{n-k}}\right) &\leq \sum_{i=0}^{\infty} 2^{(i+1)d}\frac{\delta}{\beta}\left(\frac{\delta}{2}\right)^i e^{-d(i+1)} \\ &\leq \frac{\delta}{\beta}\frac{1}{1-\delta/2} \\ &\leq \delta.\end{aligned}$$

So with probability at least $1-\delta$, every point $y \in S$ satisfies

$$\|\pi_v(y) - \pi_v(x)\| \leq \frac{\beta r}{\sqrt{n-k}}\sum_{i=0}^{\infty}\frac{\sqrt{i+1}}{2^i} \leq \frac{3r}{\sqrt{n-k}}\sqrt{2\left(d+\log(2/\delta)\right)}.$$

◀

The next lemma is from [7].

▶ **Lemma 6.** *For $S \subset B(x,\Delta)$, $\delta \in (0, 2/e^2]$ and a random unit vector $v \in \mathbb{R}^n$, with probability at least $1-\delta$,*

$$\|\text{median}(\pi_v(S)) - \pi_v(x)\| \leq \Delta\sqrt{\frac{2\log(2/\delta)}{n}}$$

▶ **Lemma 7.** *Let $S \subseteq B(x,\Delta)$ and $z \in B(x,\Delta)$. Let V be a k-dimensional subspace of $\mathbb{R}^n$ with $k < n/9$ and v be a random unit vector orthogonal to V. Then, with probability at least* 0.95,

$$\|\text{median}(\pi_v(S)) - \pi_v(z)\| \leq \frac{6\Delta}{\sqrt{n}}$$

Proof. By the triangle inequality, it suffices to show that $\|\pi_v(z-x)\| \leq 3\Delta/\sqrt{n}$ and that $\|\text{median}(\pi_v(S)) - \pi_v(x)\| \leq 3\Delta/\sqrt{n}$. The first bound uses Lemma 4 setting $\beta = \sqrt{8}$:

$$\Pr\left(\|\pi_v(z-x)\| \geq \beta\frac{\|z-x\|}{\sqrt{n-k}}\right) \leq \frac{2}{\beta}e^{-\frac{\beta^2}{2}} \leq \frac{1}{\sqrt{2}e^4}.$$

Next, since $k < n/9$,

$$\beta\frac{\|z-x\|}{\sqrt{n-k}} \leq 3\frac{\|z-x\|}{\sqrt{n}}.$$

Therefore,

$$\Pr\left(\|\pi_v(z-x)\| \geq 3\frac{\|z-x\|}{\sqrt{n}}\right) \leq \frac{1}{\sqrt{2}e^4}.$$

The second inequality is derived using Lemma 6 with $\delta = 2/e^4$.

$$\begin{aligned}\|\text{median}(\pi_v(S)) - \pi_v(x)\| &\leq \frac{\Delta}{\sqrt{n-k}}\sqrt{2\log\left(\frac{2}{\delta}\right)} \\ &\leq \frac{3\Delta}{\sqrt{n}}.\end{aligned}$$

Putting these inequalities together completes the proof, with a total failure probability of at most $(1/\sqrt{2}e^4) + (2/e^4) < 1/20$. ◀

3.4 Proof of Main Theorem

We are now ready to prove the main theorem. Let S be a set of points contained in a cell C of the tree with $\Delta = \operatorname{diam}(C \cap S)$, i,e., $S \subseteq B(x, \Delta)$.

Since S has doubling dimension d, we can cover it using $100d^{d/2}$ balls of radius $r = \Delta/100\sqrt{d}$.

Let $k < c_0 n$ and $\{v_1, \ldots, v_n\}$ be a set of random orthonormal vectors with $W = \operatorname{span}(v_1, \ldots, v_k)^{\perp}$. By by Lemma 3 and the remark following it, we have that for all centers u and v of our ball cover (including the center x) at all of at most m nodes of the tree,

$$\begin{aligned}
&\mathsf{Pr}\left(\forall u, v : \frac{9}{10}\frac{\|u-v\|^2(n-k)}{n} < \|\pi_W(u-v)\|^2 < \frac{11}{10}\frac{\|u-v\|^2(n-k)}{n}\right) \\
&\geq 1 - 10^4 d^d m e^{-\frac{n-k}{4}\left(\frac{1}{100}-\frac{1}{1000}\right)} \\
&\geq 1 - 10^4 m e^{c_0 n} e^{-\frac{n}{500}} \\
&\geq 1 - 10^4 m e^{-\frac{n}{1000}}
\end{aligned}$$

with $c_0 \leq 1/1000$. We will assume that this distortion bound holds for the rest of the proof.

Now consider two balls in this cover, $B = B(z, r)$ and $B' = B(z', r)$ where $z, z' \in B_(x, \Delta)$ and are more than $\Delta/2 - r$ apart. For each split there are three possibilities: either the partition separates B and B' (which we call a "good split"), or it intersects both B and B' (a "bad split") or the partition only intersects one or none of the two balls ("neutral split"). In the case of a "bad split", we have to now separate the four parts of B and B', and in the case of a "neutral split", we still only have to separate two objects. We will bound these probabilities for single steps in Lemmas 8 and 9 (whose proofs we defer to the end of this section).

▶ **Lemma 8** (Good splits). *Let $S \subset B(x, \Delta) \subset \mathbb{R}^n$ have doubling dimension d. Fix two balls $B(z, r)$ and $B(z', r)$ and a subspace V of dimension $k \leq n/9$ where:*

1. $z, z' \in B(x, \Delta)$.
2. $\|z - z'\| \geq \Delta/2 - r$.
3. $r \leq \Delta/(100\sqrt{d})$.
4. *The squared distances between x, z and z' are distorted by at most $1/10$ in $V^{\perp}$.*

Let v be a random unit vector orthogonal to V, and s be a point uniformly at random in the interval

$$[\operatorname{median}(\pi_v(S)) - 6\Delta/\sqrt{n}, \operatorname{median}(\pi_v(S)) + 6\Delta/\sqrt{n}].$$

Then with probability at least $1/200$, $\pi_v(B)$ and $\pi_v(B')$ lie on different sides of s.

▶ **Lemma 9** (Bad splits). *Under the above hypotheses of Lemma 8, then probability at most $1/300$, s is contained in the supports of $\pi_v(B)$ and $\pi_v(B')$.*

We prove these lemmas at the end of this section. To complete the mian proof, following [7], let p_i, be the probability that B and B' share a cell after i levels, i.e., they are not completely separated. Clearly, $p_1 \leq 199/200$ using Lemma 8. Moreover,

$$\begin{aligned}
p_i &\leq \mathsf{Pr}(\text{good split}) \times 0 + 2p_{i-1}\mathsf{Pr}(\text{bad split}) + \mathsf{Pr}(\text{neutral split})\, p_{i-1} \\
&\leq \frac{1}{200}\cdot 0 + \frac{2}{300}p_{i-1} + \left(1 - \frac{1}{200} - \frac{1}{300}\right)p_{i-1} \\
&\leq \frac{599}{600}p_{i-1}.
\end{aligned}$$

Thus we have p_k as being exponentially small in k. Denote $\alpha = 599/600$. If we take:

$$k = \frac{1}{\log(\alpha)}\left(d\log d + 5\log 10\right)$$

rounds of partitioning, then each pair of balls is in the same cell with probability at most $1/(10^5 d^d)$. Hence, by taking a union bound over all pairs of balls, no pair is in the same partition with probability at least $9/10$.

To extend the analysis to the entire tree, we simply note that with m points in S, there are at most m covers (one for each internal node of the tree) of $10^4 d^{d/2}$ balls where we have to preserve the distances between the centers.

We conclude this section with the proofs of the claims regarding good and bad splits.

Proof of Lemma 8. In Lemma 5, if we take $\delta = 2/e^9$ and $r \leq \Delta/100\sqrt{d}$, then we find that $\pi_v(B)$ lies in an interval of radius $\frac{3r}{\sqrt{n-k}}\sqrt{2(d+\log(2/\delta))}$:

$$\begin{aligned}\frac{3r}{\sqrt{n-k}}\sqrt{2(d+\log(2/\delta))} &\leq \frac{3\Delta}{100}\sqrt{\frac{2(d+9)}{d(n-k)}} \\ &\leq \frac{\Delta}{16\sqrt{n-k}}\end{aligned}$$

Next we show that with good probability,

$$\|\pi_v(z-z')\| \geq \Delta/(4\sqrt{n-k}).$$

To see this, note that projecting $z-z'$ to v is equivalent to projecting $\pi_W(z-z')$ to a random unit vector in W. We can apply Lemma 4 with $\beta = \sqrt{\frac{10}{9}}/4$

$$\begin{aligned}\mathsf{Pr}\left(\|\pi_v\pi_W(z-z')\| \leq \beta\frac{\|\pi_W(z-z')\|}{\sqrt{n-k}}\right) &\leq \beta\sqrt{\frac{2}{\pi}} \\ \mathsf{Pr}\left(\|\pi_v\pi_W(z-z')\| \leq \beta\sqrt{9/10}\frac{\|z-z'\|}{\sqrt{n-k}}\right) &\leq \beta\sqrt{\frac{2}{\pi}} \\ \mathsf{Pr}\left(\|\pi_v\pi_W(z-z')\| \leq \frac{\Delta}{4\sqrt{n-k}}\right) &\leq \sqrt{\frac{20}{\pi}}\frac{1}{12} < 0.42.\end{aligned}$$

Thus, with probability at least 0.68, there is a gap of at least

$$\frac{\Delta}{4\sqrt{n-k}} - 2\frac{\Delta}{16\sqrt{n-k}} \geq \frac{\Delta}{8\sqrt{n}}$$

between $\pi_v(B)$ and $\pi_v(B')$. On the other hand, by Lemma 7 applied to the centers of these balls, with probability at least 0.9, they are both within $6\Delta/\sqrt{n}$ of the median of the projection. Thus, with probability greater than $1/2$, we have both events: a large gap between the balls and both balls intersecting the interval around the median. The probability that random partition hits this gap, conditioned on these events is:

$$\frac{\Delta/8\sqrt{n}}{12\Delta/\sqrt{n}} \geq \frac{1}{96}$$

This gives us a final success probability of at least $1/200$. ◀

Proof of Lemma 9. As in the previous proof, we will assume that $\pi_v(B)$ and $\pi_v(B')$ concentrate in intervals of radius $\Delta/16\sqrt{n}$ around $\pi_v(z)$ and $\pi_v(z')$ respectively (this happens with probability $1-2\delta$). Now the probability that s intersects $\pi_v(B)$ is $1/96$, since $\pi_v(B)$ is contained in a ball of diameter $\Delta/8\sqrt{n}$ and the partition occurs uniformly in an interval of $12\Delta/\sqrt{n}$. Using Lemma 4, the probability that $\pi_v(B)$ and $\pi_v(B')$ intersect is bounded as follows:

$$\begin{aligned}
& \mathsf{Pr}\left(\|\pi_v(\pi_W(z-z'))\| \le \frac{\Delta}{8\sqrt{n}}\right) \\
\le\ & \mathsf{Pr}\left(\|\pi_v(\pi_W(z-z'))\| \le \left(\frac{\Delta}{8\sqrt{n}}\frac{\sqrt{n-k}}{\|\pi_W(z-z')\|}\right)\frac{\|\pi_W(z-z)\|}{\sqrt{n-k}}\right) \\
\le\ & \sqrt{\frac{2}{\pi}}\frac{\Delta}{8\sqrt{n}}\frac{\sqrt{n-k}}{\|\pi_W(z-z')\|} \\
\le\ & \sqrt{\frac{2}{\pi}}\frac{1}{8}\sqrt{\frac{8}{9}}\frac{\Delta}{\sqrt{9/10}\|z-z'\|} \\
\le\ & \frac{1}{18}\sqrt{\frac{10}{\pi}}\frac{\Delta}{(\Delta/2)-r} \\
\le\ & 0.2.
\end{aligned}$$

The probability of a bad split is upper bounded by

$$2\delta + \mathsf{Pr}\left(\pi_v(B)\cap\pi_v(B')\neq\emptyset\right)\mathsf{Pr}\left(s\in\pi_v(B)\right) < \frac{4}{e^9} + \frac{0.2}{96} < \frac{1}{300}.$$

◀

References

1 Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for near neighbor problem in high dimensions. In *STOC*, 2006.

2 Rosa Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Mach. Learn.*, 63:161–182, May 2006.

3 P. Assouad. Plongements lipschitziens dans r^n. *Bull. Soc. Math. France*, 111:429–448, 1983.

4 Jon Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

5 Alina Beygelzimer, Sham Kakade, and John Langford. Cover tree for nearest neighbor. In *ICML*, 2006.

6 Sanjoy Dasgupta. Hierarchical clustering with performance guarantees. In Hermann Locarek-Junge and Claus Weihs, editors, *Proceedings of the 11th IFCS Biennial Conference and 33rd Annual Conference of the Gesellschaft für Klassifikation e.V., Studies in Classification, Data Analysis, and Knowledge Organization*. Springer-Verlag, 2010.

7 Sanjoy Dasgupta and Yoav Freund. Random projection trees and low dimensional manifolds. In *STOC*, 2008.

8 Sanjoy Dasgupta and Yoav Freund. Random projection trees for vector quantization. *IEEE Trans. Inf. Theor.*, 55:3229–3242, July 2009.

9 Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures and Algorithms*, 22(1):60–65, 2003.

10 Aman Dhesi and Purushottam Kar. Random projection trees revisited. In *NIPS*, pages 496–504, 2010.

11 Anupam Gupta, Robert Krauthgamer, and James R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *FOCS*, pages 534–543, 2003.

12 Sariel Har-Peled. A replacement for voronoi diagrams of near linear size. In *FOCS*, 2001.

13 Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, 1998.

14 Piotr Indyk, Rajeev Motwani, Prabhakar Raghavan, and Santosh Vempala. Locality-preserving hashing in multidimensional spaces. In *STOC*, 1997.

15 William Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

16 Samory Kpotufe. Escaping the curse of dimensionality with a tree-based regressor. In *Conference on Computational Learning Theory*, 2009.

17 Robert Krauthgamer and James Lee. Navigating nets: simple algorithms for proximity search. In *SODA*, 2004.

18 Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM J. Comput.*, 30(2):457–474, 2000.

19 Robert Sproull. Refinements to nearest-neighbor searching in k-dimensional trees. *Algorithmica*, 6:579–589, 1991.

20 Santosh S. Vempala. *The Random Projection Method*. The American Mathematical Society, 2004.