Accounting for Boundary Effects in Nearest Neighbor Searching*

Sunil Arya† David M. Mount‡ Onuttom Narayan §

Abstract

Given n data points in d-dimensional space, nearest neighbor searching involves determining the nearest of these data points to a given query point. Most average-case analyses of nearest neighbor searching algorithms are made under the simplifying assumption that d is fixed and that n is so large relative to d that *boundary effects* can be ignored. This means that for any query point the statistical distribution of the data points surrounding it is independent of the location of the query point. However, in many applications of nearest neighbor searching (such as data compression by vector quantization) this assumption is not met, since the number of data points n grows roughly as 2^d. Largely for this reason, the actual performances of many nearest neighbor algorithms tend to be much better than their theoretical analyses would suggest. We present evidence of why this is the case. We provide an accurate analysis of the number of cells visited in nearest neighbor searching by the bucketing and k-d tree algorithms. We assume m^d points uniformly distributed in dimension d, where m is a fixed integer ≥ 2. Further, we assume that distances are measured in the L_∞ metric. Our analysis is tight in the limit as d approaches infinity. Empirical evidence is presented showing that the analysis applies even in low dimensions.

*A preliminary version of this paper appeared in the *Proc. of the 11th Annual ACM Symp. on Computational Geometry*, 1995, pp. 336–344.

†Department of Computer Science, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. Part of this research was conducted while the author was visiting the Max-Planck-Institut für Informatik, Saarbrücken, Germany. The author was supported by the ESPRIT Basic Research Actions Program, under contract No. 7141 (project ALCOM II).

‡Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, Maryland. The support of the National Science Foundation under grant CCR–9310705 is gratefully acknowledged.

§Physics Department, Harvard University, Cambridge, MA 02138 and AT&T Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974. Supported in part by the Society of Fellows at Harvard University.

1 Introduction

Finding nearest neighbors is a fundamental problem in computational geometry with applications to many areas such as pattern recognition, data compression, and statistics. The *nearest neighbor* problem is: given a set of n points P in d-dimensional space, and given a query point $q \in R^d$, find the point of P that minimizes the distance to q.

The problem of preprocessing a set of n points P so that nearest neighbor queries can be answered efficiently has been extensively studied. Nearest neighbor searching can be performed quite efficiently in relatively low dimensions. However, as the dimension d increases, in the worst case, either the space or time complexities increase dramatically. There are many important applications of this problem in moderate dimensions (e.g. in the range from 10 to 20), where best worst-case algorithms are of little practical value. For dimensions in this range, the most practical approaches for nearest neighbor searching are based on simple spatial subdivisions, having poor worst-case performance, but relatively good average-case performance. (Recently it has been suggested that another way to achieve good performance in practice is to compute approximate rather than exact nearest neighbors [2, 4, 7].)

A very simple algorithm that works well for uniformly distributed data is the *bucketing algorithm* (sometimes called *Elias's algorithm* [15]). Rivest [13] presented an analysis of the performance of this algorithm for points uniformly distributed on the vertices of a d-dimensional hypercube, and later Cleary analyzed the case of general uniformly distributed point sets [8]. A practical and more flexible approach for nearest neighbor searching in high dimensions is based on the k-d tree, as introduced by Bentley [5] and applied to the nearest neighbor problem by Friedman, Bentley and Finkel [9].

The analyses of these algorithms presented in the literature [8, 9] show that their running times contain a constant factor, which grows on the order of 2^d, assuming the L_∞ metric. The analysis in [8] assumes that the points are uniformly distributed while the analysis in [9] assumes that the points are randomly chosen from some smooth underlying distribution. The dimension d is regarded as a fixed constant, and the analysis is asymptotic as n tends to infinity. This greatly simplifies the analysis because for any query point (assuming it is chosen from the same distribution as the data points), the statistical distribution of the data points surrounding it can be assumed to be essentially independent of the location of the query point.

However, there are many important applications where the number of data points n and dimension d are related. One such application is *vector quantization*, a technique used in the compression of speech and images [11]. Samples taken from a signal are blocked into vectors of length d (typically after applying some smoothing transforms). Based on a training set of vectors, a set of codevectors is first precomputed. The technique then encodes each new

vector by the index of its nearest neighbor among the codevectors. The rate r of a vector quantizer is the number of bits used to encode a sample, and it is related to n, the number of codevectors, by $n = 2^{rd}$. For the common case of $r = 1$, it follows that $n = 2^d$.

For applications in which d and n are related, the theoretical analyses may significantly overestimate the running time of the algorithm. In fact, our interest in this problem stemmed from our observations of the empirical running times for the k-d tree and related algorithms (even for more general point distributions). These algorithms regularly run faster than their predicted performance in higher dimensions. Intuitively, the reason is that when a query point lies close to the periphery of the point set, a significant amount of the data structure that would otherwise need to be searched for the nearest neighbor may be pruned away. Because exponential constant factors in dimension are one of the main obstacles to extending nearest neighbor searching to much higher dimensions (where many more important applications reside), it is of important practical interest to accurately understand the nature of these factors.

In this paper we provide a theoretical explanation of the phenomenon of these *boundary effects* in nearest neighbor searching. We analyze the bucketing algorithm for the uniform distribution, taking into account the effects of the boundary. Our results also apply to the k-d tree algorithm [8]. Because of the complexity of the analysis, we assume that points are uniformly distributed in a d-dimensional unit hypercube, and that distances are measured using the L_∞ metric. Our main result is that given 2^d points in d dimensions, as d tends to infinity, the expected number of cells visited by the bucketing algorithm grows as $(0.90\ldots)(1.56594\ldots)^d$. Empirical evidence indicates that this is remarkably close, even for small d. This is significantly smaller than the growth rate of 2^d predicted by previous analyses which ignore boundary effects. (For example, a difference of roughly a factor 7.8 for dimension 8, and a factor of 55 in dimension 16.) We also generalize these results to the case of m^d points in d dimensions, where m is an integer ≥ 2.

The remainder of the paper is organized as follows. In the next section we present some background on the k-d tree and bucketing algorithms. We present two different analyses on the number of cells visited by the bucketing algorithm. The model is described in Section 3 and analyzed in Section 4 and 5. Both the analyses are done for the limiting case of large dimensions. Section 4 provides a simple analysis which yields a fairly good upper bound. This analysis, however, relies on a technical assumption which we call the *monotonicity conjecture*. In Section 5 we present a sophisticated analysis which yields a tight bound. This does not rely on the monotonicity conjecture. We also present a "numerical proof" of the monotonicity conjecture in Section 5, supporting the validity of the simpler analysis of Section 4. In Section 6 we discuss extensions and generalizations of this refined analysis. Finally, in Section 7 we provide the results of our empirical analysis.

2 Preliminaries

One of the the most practical and simple approaches for nearest neighbor searching in high dimensions is based on the *k-d* tree. Bentley [5] introduced the *k-d* tree as a generalization of the binary search tree in high dimensions. Each internal node of the *k-d* tree is associated with a hyperrectangle and a hyperplane orthogonal to one of the coordinate axis, which splits the hyperrectangle into two parts. These two parts are then associated with the two child nodes. The process of partitioning space continues until the number of data points in the hyperrectangle falls below some given threshold. Given a suitable splitting rule, the *k-d* tree induces a partitioning of space into cells whose sizes adapt to the local density of the data points; the partitioning is finer where the density is higher.

Friedman, Bentley and Finkel [9] gave an algorithm to find the nearest neighbor using optimized *k-d* trees. The internal nodes of the *optimized k-d* tree split the set of data points lying in the corresponding hyperrectangle into two equal parts, along the dimension in which the data points have maximum spread. The algorithm works by first descending the tree to find the data points lying in the cell that contains the query point. Then it examines surrounding cells if they overlap the ball B centered at the query point and having radius equal to the distance between the query point and the closest data point visited so far. Efficient implementations of the *k-d* tree algorithm have been given by Sproull [14] and Arya and Mount [3].

Friedman et al. [9] showed that their algorithm takes $O(\log n)$ expected time, under certain simplifying assumptions on the distribution of data and query points. They also showed that the expected number of points examined attains its minimum value when each cell contains one point, and is bounded by

$$E[N] \leq \{[G(d)]^{1/d} + 1\}^d. \tag{1}$$

Here d is the dimension and $G(d)$ is the ratio of the volume of a d-dimensional hypercube to the volume of the largest enclosed ball in the metric used for distance measurement. For the L_∞ metric, $G(d)$ is 1, and the expected number of points examined is bounded by 2^d.

A much simpler algorithm that works well for uniformly distributed data is the *bucketing algorithm*, (sometimes called *Elias's algorithm* [15]). Space is divided into identical cells and for each cell, the data points inside it are stored in a list (see Fig. 1). The cells are examined in order of increasing distance from the query point and for each cell the distance is computed between the data points inside it and the query point. The search terminates when the distance from the query point to the cell exceeds the distance to the closest point visited. The algorithm was analyzed for data points uniformly distributed on the vertices of the d-dimensional hypercube by Rivest [13], and later for general uniformly distributed point sets by Bentley, Weide and Yao [6] and Cleary [8]. Cleary's analysis showed that the

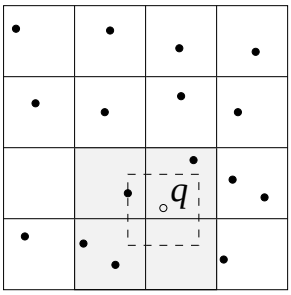

Figure 1: Bucketing Algorithm.

number of points examined was independent of n, and that the fewest number of points were examined when there was one data point per cell. His analysis also applies to the k-d tree algorithm and furnishes more accurate bounds than those obtained by Friedman et al [9].

3 Model

Consider a set of m^d data points distributed uniformly in a d-dimensional hypercube H with edge length m. We will assume that m is an integer greater than 1. The hypercube H is split into m^d cells of equal size, which gives a density of one data point per cell. Each cell stores a pointer to the data points inside it (if any). Fig. 1 shows the subdivision in two dimensions for $m = 4$. Consider that the query point is also chosen from the uniform distribution. The bucketing algorithm visits the cells in increasing order of distance from the query point, examines the point(s) associated with each cell, and updates the closest point seen so far. The algorithm terminates when the next cell to be visited is farther than the closest point visited until then. The complexity of the algorithm is measured by the number of cells visited averaged over all possible locations of the query point.

4 Crude Analysis

In this section, we compute an upper bound on the expected number of cells visited by the algorithm. Our analysis is done for the large d limit, treating m as fixed. We will assume that all distances are measured in the L_∞ metric. Given a query point $\mathbf{x} = (x_1, \ldots, x_d)$, define the *nearest neighbor ball* to be the L_∞ ball centered at $\mathbf{x}$ and having radius equal to the distance between $\mathbf{x}$ and the data point closest to it. It is clear from the description of the bucketing algorithm that a cell is visited if and only if it overlaps the nearest neighbor ball. The quantity of interest therefore is the expected number of cells that overlap the nearest neighbor ball, where the expectation is computed over all possible locations of the

query point.

Denote an L_∞-ball of radius r centered at the query point $\mathbf{x}$ by $B(\mathbf{x}, r)$. This is a hypercube of side length $2r$ centered at the query point. It is easy to show that the expected volume of intersection of the nearest neighbor ball with hypercube H approaches unity in the large d limit. This is true no matter where the query point is located. This suggests the following model for the nearest neighbor ball, which we employ to simplify the analysis. For any query point $\mathbf{x}$, we call a ball centered at $\mathbf{x}$ and having unit volume of intersection with hypercube H, a *unit intersection ball*. We will compute an upper bound on the expected number of cells overlapped by the unit intersection ball, and regard this as an approximation to the true upper bound. (Rigorous justification for this assumption is provided in Section 6, where we will see that the expected number of cells overlapped by the unit intersection ball differs from the expected number of cells overlapped by the nearest neighbor ball by at most a constant multiplicative factor, which does not depend on the dimension.)

We define several random variables which we need for our analysis. These random variables are all functions of the location $\mathbf{x}$ of the query point. Let the random variable u denote the radius of the unit intersection ball, and let N denote the number of cells overlapped by the unit intersection ball. More precisely, $u(\mathbf{x})$ is the value of r such that $Vol(B(\mathbf{x}, r) \cap H) = 1$, and $N(\mathbf{x})$ is the number of cells overlapped by the unit intersection ball, $B(\mathbf{x}, u(\mathbf{x}))$.

For any given r, we define the random variable $V_r(\mathbf{x})$ to be volume of intersection of the ball of radius r centered at $\mathbf{x}$ with hypercube H. Thus, $V_r(\mathbf{x}) = Vol(B(\mathbf{x}, r) \cap H)$. For any given r, we have d associated random variables, $I_{i,r}, 1 \leq i \leq d$, which we call the *intercepts*. The i-th intercept, $I_{i,r}(\mathbf{x})$ is the length of the side of $B(\mathbf{x}, r) \cap H$ along the i-th dimension. It is easy to see that the volume of intersection, $V_r(\mathbf{x})$, is the product of d intercepts,

$$V_r(\mathbf{x}) = \prod_{i=1}^{d} I_{i,r}(\mathbf{x}). \tag{2}$$

Further, for any given r, the d intercepts, $I_i(r), 1 \leq i \leq d$, are independent, identically distributed random variables.

Finally, for given r, we define the random variable $Z_r(\mathbf{x})$ to be the number of cells overlapped by the ball of radius r centered at $\mathbf{x}$. For fixed r, we have d associated random variables, $Z_{i,r}, 1 \leq i \leq d$, where $Z_{i,r}(\mathbf{x})$ is the number of unit intervals overlapped along the i-th dimension. Thus,

$$Z_r(\mathbf{x}) = \prod_{i=1}^{d} Z_{i,r}(\mathbf{x}). \tag{3}$$

Note that for given r, the d random variables, $Z_{i,r}, 1 \leq i \leq d$ are independent, identically distributed random variables.

Let log denote the natural logarithm function. We define the critical radius c to be the value of r such that $E[\log(I_{i,r})] = 0$. In taking the expectation, i and r are regarded as fixed, and the expectation is taken over all possible locations $\mathbf{x}$ of the query point. Such a value of r must exist in the range $0.5 \leq r \leq 1$, since for any $\mathbf{x}$, $I_{i,r}(\mathbf{x}) \geq 1$ for $r \geq 1$ and $I_{i,r}(\mathbf{x}) \leq 1$ for $r \leq 0.5$. It is important to note that the critical radius c depends only on the edge length m defined earlier (see Section 3). In particular, it is independent of the dimension. This is because for fixed r, $I_{i,r}$ for all $1 \leq i \leq d$ are independent and identically distributed, so that $E[\log(I_{i,r})] = 0$ implies $E[\log(I_{j,r})] = 0$ for any pair i, j.

The following lemma establishes that the radius of the unit intersection ball, u, almost always lies close to c, for high dimension.

Lemma 1 *For any constant $\epsilon > 0$,*

$$\lim_{d \to \infty} P[c - \epsilon < u < c + \epsilon] = 1. \tag{4}$$

Proof: We only establish that $\lim_{d\to\infty} P[u < c + \epsilon] = 1$. The proof of the other part is similar and is omitted.

Consider the d independent and identically distributed random variables, $I_{i,c+\epsilon}, 1 \leq i \leq d$. From the definition of the critical radius c and the fact that $E[\log(I_{i,r})]$ is clearly a monotonically increasing function of r, it follows that $E[\log(I_{i,c+\epsilon})] > 0$.

We will apply the Weak Law of Large Numbers to the variables, $I_{i,c+\epsilon}, 1 \leq i \leq d$. The Weak Law of Large Numbers states that given a sequence of independent, identically distributed variables X_i with finite expectation, the probability that the average $\sum_{1 \leq i \leq d} X_i/d$ differs from the expectation $E[X_i]$ by less than an arbitrarily small positive δ approaches 1, as d approaches infinity [10, page 243]. Applying this to the variables $I_{i,c+\epsilon}, 1 \leq i \leq d$, using sufficiently small δ, and noting that $E[\log(I_{i,c+\epsilon})] > 0$, we have

$$\lim_{d \to \infty} P\left[\frac{1}{d} \sum_{i=1}^{d} \log(I_{i,c+\epsilon}) > 0\right] = 1. \tag{5}$$

Rewriting this expression and taking antilogs, we get

$$\lim_{d \to \infty} P\left[\prod_{i=1}^{d} I_{i,c+\epsilon} > 1\right] = 1. \tag{6}$$

Since $V_r = \prod_{i=1}^{d} I_{i,r}$, we can write this as

$$\lim_{d \to \infty} P\left[V_{c+\epsilon} > 1\right] = 1. \tag{7}$$

Since $u(\mathbf{x}) < c + \epsilon$ if and only if $V_{c+\epsilon}(\mathbf{x}) > 1$, we obtain the desired result. □

The following lemma gives the expected value of Z_r, the number of cells overlapped by the L_∞-ball $B(\mathbf{x}, r)$.

Lemma 2 *For* $0.5 \leq r \leq 1$,

$$E[Z_r] = \left[1 + 2r\left(1 - \frac{1}{m}\right)\right]^d. \tag{8}$$

Proof: Recall that the random variable $Z_{i,r}$, for $1 \leq i \leq d$, denotes the number of unit intervals overlapped along the i-th dimension, by the corresponding side of $B(\mathbf{x}, r)$. Then Z_r is given by $\prod_{i=1}^{d} Z_{i,r}$. For any given r, the d random variables $Z_{i,r}, 1 \leq i \leq d$ are independent variables. Therefore,

$$E[Z_r] \quad = \quad \prod_{i=1}^{d} E[Z_{i,r}]. \tag{9}$$

Fox fixed r, the variables $Z_{i,r}, 1 \leq i \leq d$ have a common distibution. It is easy to see that for $0.5 \leq r \leq 1$, $Z_{i,r}$ takes on the values 1, 2, and 3 with the following probabilities.

$$\begin{array}{ll} Z_{i,r} & \text{Probability} \\ 1 & \frac{2-2r}{m} \\ 2 & \frac{(m-2)(2-2r)+2r}{m} \\ 3 & \frac{(m-2)(2r-1)}{m} \end{array} \tag{10}$$

From this, we get the following expression for the expected value of $Z_{i,r}$,

$$E[Z_{i,r}] = \left[1 + 2r\left(1 - \frac{1}{m}\right)\right]. \tag{11}$$

Substituting into Eq. (9), we get the desired result. □

We now come to the main theorem of this section, which establishes an upper bound on the expected number of cells overlapped by the unit intersection ball. (Since Lemma 1 says that in high dimensions the radius of the unit intersection ball, u, is almost always close to c, one may be tempted to conclude that the expected number of cells overlapped by the unit intersection ball, $E[N]$, approaches the expected number of cells overlapped by a ball of radius c, $E[Z_c]$. However, we will see in Section 5 that this is in fact not true.) Our proof of the theorem relies on the following conjecture.

Conjecture 1 (Monotonicity conjecture) *For all* d*, the expected number of cells overlapped by the unit intersection ball subject to the constraint that the radius of the unit intersection ball is* r*,* $E[N \mid u = r]$*, is a monotonically decreasing function of* r*.*

The intuition is that the radius of the unit intersection ball is usually smaller when the query point is away from the boundaries of hypercube H; this case leads to a large number of cells being overlapped by the unit intersection ball. While we we have confirmed this observation in our experiments, we do not have a rigorous proof for it.

We present some evidence in favor of the conjecture in Fig. 2 and 3, which show the plots obtained experimentally of $[E[N \mid u = r]]^{1/d}$ versus r for $d = 8$ and $d = 16$, respectively (here $m = 2$). To compute this plot, we partitioned the radii in the range 0.5 to 0.75 into a set of intervals. We generated 10^7 points from the uniform distribution, and for each point, grew a ball having unit volume of intersection with hypercube H. We determined the interval in which the radius of the ball lies and the number of cells overlapped by it. For each interval, we computed the average number of cells overlapped over all balls whose radius lies in that interval, and plotted these against the interval midpoint. (The plots terminate at $r = 0.75$ since the probability of finding points where the radius of the nearest neighbor ball is higher than this is very low and our sampling technique does not find enough of them.) We will see further justification for the monotonicity conjecture in the next section, where we give a numerical proof for it in the large d limit, which does not rely on simulation.

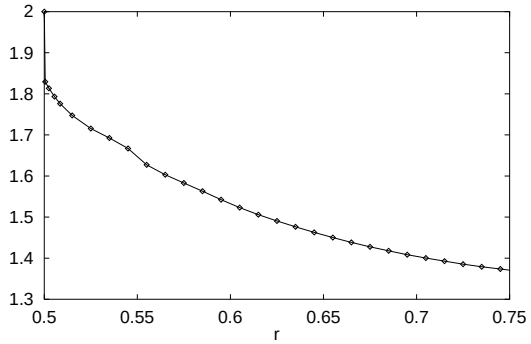

Figure 2: Monotonicity conjecture (for $d = 8$): $[E[N \mid u = r]]^{1/d}$ vs. r.

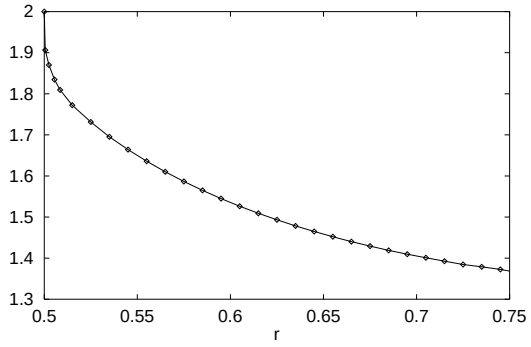

Figure 3: Monotonicity conjecture (for $d = 16$): $[E[N \mid u = r]]^{1/d}$ vs. r.

Theorem 1 *Assuming Conjecture 1, for any constant $\epsilon > 0$, there exists d_0 such that for*

all $d > d_0$,

$$E[N] \leq \left[1 + 2(c+\epsilon)\left(1 - \frac{1}{m}\right)\right]^d. \tag{12}$$

Proof:

Let ϵ' be any positive constant less than ϵ. We compute $E[N]$ over two possible cases, $u < c + \epsilon'$ and $u \geq c + \epsilon'$.

$$\begin{aligned} E[N] = & \\ & P[u < c+\epsilon'] \cdot E[N \mid u < c+\epsilon'] + \\ & P[u \geq c+\epsilon'] \cdot E[N \mid u \geq c+\epsilon']. \end{aligned} \tag{13}$$

Conjecture 1 implies that

$$E[N \mid u \geq c+\epsilon'] \leq E[N \mid u < c+\epsilon']. \tag{14}$$

Substituting into Eq. (13) we get an upper bound for $E[N]$,

$$E[N] \leq E[N \mid u < c+\epsilon']. \tag{15}$$

Further,

$$E[N \mid u < c+\epsilon'] \leq E[Z_{c+\epsilon'} \mid u < c+\epsilon']. \tag{16}$$

This follows from the fact that for any $\mathbf{x}$, the number of cells overlapped by a ball of radius r is a non-decreasing function of the radius. Combining this with Eq. (15), we have

$$\begin{aligned} E[N] &\leq E[Z_{c+\epsilon'} \mid u < c+\epsilon'] \\ &\leq \frac{E[Z_{c+\epsilon'}]}{P[u < c+\epsilon']}. \end{aligned} \tag{17}$$

Substituting for $E[Z_{c+\epsilon'}]$ from Lemma 2, we get

$$E[N] \leq \frac{\left[1 + 2(c+\epsilon')\left(1 - \frac{1}{m}\right)\right]^d}{P[u < c+\epsilon']}. \tag{18}$$

From Lemma 1, it follows that for sufficiently large d, $P[u < c+\epsilon']$ is arbitrarily close to 1. Since $\epsilon' < \epsilon$, so that $[1+2(c+\epsilon)(1-1/m)]^d / [1+2(c+\epsilon')(1-1/m)]^d$ diverges exponentially with d, this implies that for sufficiently large d,

$$E[N] \leq \left[1 + 2(c+\epsilon)\left(1 - \frac{1}{m}\right)\right]^d, \tag{19}$$

which is the desired result.

□

m	$E[N] \leq$
2	$(1.601 + \epsilon)^d$
3	$(1.748 + \epsilon)^d$
4	$(1.815 + \epsilon)^d$
5	$(1.854 + \epsilon)^d$
10	$(1.929 + \epsilon)^d$
100	$(1.993 + \epsilon)^d$

Table 1: Upper bound on expected number of cells overlapped by the unit intersection ball.

To compute the critical radius c, we first write the cumulative density for $I_{i,r}$. It is easy to see that this is given by

$$P[I_{i,r} \leq t] = \begin{cases} 0 & \text{if } t < r \\ \frac{2(t-r)}{m} & \text{if } r \leq t < 2r \\ 1 & \text{if } t = 2r. \end{cases} \tag{20}$$

From this, after some calculation, we get the following expression for $E[\log(I_{i,r})]$,

$$E[\log(I_{i,r})] = \frac{2r}{m}(\log(2) - 1) + \log(2r). \tag{21}$$

For any m, we can equate this expression to 0, and solve it numerically to obtain the critical radius c. We can then use Theorem 1 to compute an upper bound on $E[N]$, the expected number of cells visited. The results are shown in Table 1 for several different values of m. For $m = 2$, the table shows an upper bound of $(1.601\ldots + \epsilon)^d$, for any small $\epsilon > 0$. Since the average density is one point per cell, this bound also applies to the number of points visited by the algorithm. The fact that the number of points visited is much fewer than the asymptotic bound of 2^d confirms the importance of boundary effects when the number of points is on the order of 2^d.

5 Refined Analysis

In this section and the next, we achieve a tight bound for the expected number of cells visited by the bucketing algorithm. Our analysis does not rely on the monotonicity conjecture. For simplicity we do the analysis for the case of $m = 2$. Further, as in the previous section, we will assume that the ball of interest centered at the query point has unit intersection volume with hypercube H. In the next section we generalize the result to arbitrary m, and consider the modifications necessitated by the fact that the nearest neighbor ball is not quite the unit intersection ball.

Let $\mathbf{x} = (x_1, \ldots, x_d)$ denote a point in the hypercube H. Consider an L_∞ ball centered at $\mathbf{x}$. Let $\hat{V}$ denote the logarithm of its volume of intersection with hypercube H, and let $Z(\mathbf{x}; \hat{V})$ be the number of cells that this ball overlaps. The quantity of interest $\overline{N}$ is then $Z(\mathbf{x}; 0)$, averaged over $\mathbf{x}$ lying anywhere inside hypercube H (note that this is the same as the unconstrained expectation value $E[N]$ defined in the last section). Thus

$$\overline{N} = \int d\hat{V} \delta(\hat{V}) \int_0^1 dx_1 \ldots \int_0^1 dx_d Z(\mathbf{x}; \hat{V}) \tag{22}$$

where the δ-function enforces the condition that we are only interested in balls which have unit intersection volume with hypercube H. (Recall that the Dirac δ-function has the properties (a) $\delta(x) = 0$ if $x \neq 0$ and (b) $\int_{-\infty}^{\infty} g(x)\delta(x)dx = g(0)$ for all integrable $g : \mathbb{R} \rightarrow \mathbb{R}$ [1, pages 481–484].) Here the range of each x_i integral is taken to be 0 to 1, since the 0 to 2 interval is symmetric around 1. Note that in this form we do not need any normalization factors, because the volume of the region over which we are integrating is unity.

To solve the above integral, we need some elementary concepts from the theory of complex functions [1, pages 352–434]. We will use the following notation. The real and imaginary parts of a complex number z are denoted by $\mathrm{Re}(z)$ and $\mathrm{Im}(z)$, respectively. Let $z = r\exp[i\theta]$ be the polar form of a complex number z. In this representation, r is called the modulus or magnitude of z, and is denoted $|z|$ or $\mathrm{Abs}(z)$. The angle θ is labeled the argument or phase of z, and is denoted $\mathrm{Arg}(z)$.

Eq. (22) can be expressed in terms of the radius r of the ball as

$$\overline{N} = \int_{\frac{1}{2}}^1 dr \int_0^1 dx_1 \ldots \int_0^1 dx_d Z(\mathbf{x}; r) \frac{\partial \hat{V}(\mathbf{x}; r)}{\partial r} \delta(\hat{V}(\mathbf{x}; r)). \tag{23}$$

The integral over r ranges from $\frac{1}{2}$ to 1, since these are the minimum and maximum values of r needed for any $\mathbf{x}$ in order to achieve a ball with unit intersection volume.

The quantity $\hat{V}$, which is the logarithm of the volume of intersection, is the sum of the logarithms of the intercepts on all the d sides, which are given by

$$\hat{I}_i(x_i; r) \quad = \quad \begin{cases} \log(x_i + r) & \text{for } x_i \leq r \\ \log(2r) & \text{for } x_i \geq r. \end{cases} \tag{24}$$

Therefore the derivative, $\partial\hat{V}/\partial r$ is the sum of d terms, each of which lies between $1/r$ and $1/2r$, *i.e.* between $\frac{1}{2}$ and 2. Therefore, to within a constant factor,

$$\overline{N} = \Theta\left(d \int_{\frac{1}{2}}^1 dr \int_0^1 dx_1 \ldots \int_0^1 dx_d Z(\mathbf{x}; r) \delta\left(\sum_{1 \leq i \leq d} \hat{I}_i(x_i; r) \right) \right). \tag{25}$$

For any specific choice of r and $\mathbf{x}$, the number Z of cells overlapped is the product of d factors $Z_i(x_i; r)$, which are given by

$$Z_i(x_i; r) \quad = \quad \begin{cases} 1 & \text{for } x_i \leq 1 - r \\ 2 & \text{for } x_i > 1 - r. \end{cases} \tag{26}$$

Thus Eq. (25) can be written as

$$\overline{N} = \Theta\left(d\int_{\frac{1}{2}}^{1} dr \int_{-\infty}^{\infty} \frac{d\omega}{2\pi}\left\{\int_{0}^{1} dx_i Z_i(x_i;r)\exp[-j\omega\hat{I}_i(x_i;r)]\right\}^d\right). \tag{27}$$

Here we have used the representation of the δ-function, $2\pi\delta(y) = \int d\omega \exp[-j\omega y]$, where $j^2 = -1$ [1, pages 481–484]. For fixed ω and r, the integrals over all the x_i's then decouple from each other.

The term in the {}-brackets is equal to

$$\begin{aligned} F(\omega;r) = & \\ & \int_{x_i=r}^{1} dx_i 2\exp[-j\omega\log(2r)] + \\ & \int_{x_i=1-r}^{x_i=r} dx_i 2\exp[-j\omega\log(x_i+r)] + \\ & \int_{x_i=0}^{x_i=1-r} dx_i \exp[-j\omega\log(x_i+r)], \end{aligned} \tag{28}$$

which can be evaluated to be

$$\begin{aligned} F(\omega;r) = & \\ & \left\{2(1-r) + \tfrac{4r}{1-j\omega}\right\}\exp[-j\omega\log(2r)] - \\ & \tfrac{1}{1-j\omega} - \tfrac{r}{1-j\omega}\exp[-j\omega\log(r)]. \end{aligned} \tag{29}$$

To solve for $\overline{N}$ from Eq. (27), we first evaluate the ω integral. Although ω is a real variable in Eqs. (27) and (29), it is convenient to analytically continue the definition of $F(\omega;r)$ to the complex plane for ω. We then deform the contour of integration for ω from the real axis to some other contour. We shall first evaluate the integral over this deformed contour, and then prove that it is equal to the integral over the original contour (the real axis). In order to choose a convenient contour of integration, we use the following theorem, called the method of *steepest descent* [1, pages 428–434],[12, pages 287–291]. Since we need a slightly more general version of this method than is discussed in these references, we offer a proof.

Theorem 2 *If $f(z)$ and $g(z)$ are analytic functions of the complex variable z, and*

(a) $\partial_z f(z) = 0$ at $z = z_0$,

(b) $\partial_z^2 f(z) \neq 0$ at $z = z_0$,

(c) the contour C is finite, passes through z_0 and satisfies the condition $\lim_{z\to z_0}(\mathrm{Arg}[f(z)] - \mathrm{Arg}[f(z_0)])/|z-z_0|^3$ exists, and

(d) along the contour C there is a global maximum in $\mathrm{Abs}[f(z)]$ at $z = z_0$,

then in the asymptotic $d \to \infty$ limit,

$$\int_C dz g(z)[f(z)]^d \sim g(z_0)\exp[j\alpha][f(z_0)]^d \sqrt{\frac{2\pi}{d}}\left|\frac{\partial^2 f(z)}{\partial z^2}\Big|_{z=z_0}\frac{1}{f(z_0)}\right|^{-1/2} \tag{30}$$

where α is $\mathrm{Arg}(z-z_0)$ in the limit that z approaches z_0 along the contour of integration.

We use the $\sim$ symbol to mean that the ratio of the two sides of the equation tends to 1 as $d \to \infty$. The basic idea behind this theorem is that if $f(z)$ has a global maximum along the contour C at z_0, then $[f(z)]^d$ falls off extremely rapidly as z moves away from z_0, so that the integral is dominated by a small neighborhood of z_0. If the phase of $f(z)$ changes slowly in the neighborhood of z_0, then $[f(z)]^d$ has an approximately constant phase in the small neighborhood of z_0, and is governed by the Gaussian falloff close to the maximum.

Proof:

We prove this theorem for $g(z) = 1$; the proof can be easily generalized. Parametrize the contour C by $s(z)$, which is the distance of the point z from z_0 measured along C. Let s_0 be the total length of the contour C. Divide the contour C into two parts: C_1 consists of the part of C satisfying the condition $|s| < s_1 = \mu[(\log d)/d]^{1/2}$, with μ some constant, and C_2 is the rest of C.

We first do the integral over the curve C_1. For large d, C_1 is very small, so that we can Taylor expand $f(z)$ around z_0 and write

$$\int_{C_1} dz[f(z)]^d = [f(z_0)]^d \int_{C_1} ds \frac{dz}{ds}\exp\left[-d\left(\frac{1}{2}Ks^2 + O(s^3)\right)\right] \tag{31}$$

where by condition c), K is real, and is equal to $|\partial_z^2 f(z_0)/f(z_0)|$. Since $|s| < \mu[(\log d)/d]^{1/2}$, the $dO(s^3)$ term in the exponent tends to zero over the whole range of integration as $d \to \infty$, and can be neglected. Similarly, dz/ds can be replaced by $dz/ds|_{s=0} = \exp[j\alpha]$ as $d \to \infty$. The resulting Gaussian integral can be performed by changing variables to $\overline{s} = s\sqrt{d}$, and recognizing that $\int d\overline{s}\exp[-K\overline{s}^2/2]$ over $|\overline{s}| < \mu[\log d]^{1/2}$ tends to $(2\pi/K)^{1/2}$ as $d \to \infty$, which yields the right hand side of Eq. (30).

Now since $\mathrm{Abs}[f(z)]$ has a smooth global maximum over C at z_0, it follows that for sufficiently small C_1 (i.e. sufficiently large d), $\mathrm{Abs}[f(z)]$ at $s = s_1$ is greater than $\mathrm{Abs}[f(z)]$ anywhere in C_2. Expanding $f(z)$ as in Eq. (31), we see that $\int_{C_2} dz[f(z)]^d < s_0\exp[-Kds_1^2/2][f(z_0)]^d$. Substituting $s_1 = \mu[(\log d)/d]^{1/2}$, we see that the integral over C_2 is negligible compared to that over C_1 as $d \to \infty$ if μ is chosen to be greater than $1/\sqrt{K}$. □

In order to use this theorem, we need to find a contour C in the ω plane that satisfies the conditions of the theorem. We first seek a point $\omega(r)$ that satisfies conditions (a) and (b) required of the point z_0 in Theorem 2:

Lemma 3 *For any $\frac{1}{2} < r < 1$, there exists a saddle point $\omega(r)$ on the imaginary axis, where $\partial_\omega F(\omega; r) = 0$. At this point $\partial^2_\omega F(\omega; r) \neq 0$. In the neighborhood of this point, with $\omega = \omega(r) + u$ (with real u), the real and imaginary parts of $F(\omega; r)$ can be expanded as $\mathrm{Re}[F(\omega; r)] = \mathrm{Re}[F(\omega(r); r)] - Au^2$ plus higher order terms in u (with A positive), and $\mathrm{Im}[F(\omega; r)] = O(u^3)$.*

Proof: By inspection of Eq. (29), we see that $\mathrm{Re}[F(\omega; r)]$ is an even function of $\mathrm{Re}(\omega)$, and $\mathrm{Im}[F(\omega; r)]$ is an odd function of $\mathrm{Re}(\omega)$. Therefore the derivative of the real part of $F(\omega; r)$ with respect to $\mathrm{Re}(\omega)$ vanishes everywhere on the imaginary ω axis. By numerical construction, we have verified that, for ω on the imaginary axis, $\mathrm{Re}[F(\omega; r)]$ has a minimum at some $\omega(r)$ for all $\frac{1}{2} < r < 1$, so that $\partial_\omega \mathrm{Re}[F(\omega; r)]$ is zero at $\omega(r)$. We have also verified numerically that the second derivative of $\mathrm{Re}[F(\omega; r)]$ along the imaginary ω axis is non-zero at $\omega(r)$.

By the properties of analytic functions, if $\omega = \omega(r) + u + jv$ (with real u and v), then $\partial_u \mathrm{Re}[F(\omega; r)] = \partial_v \mathrm{Im}[F(\omega; r)]$ and $\partial_v \mathrm{Re}[F(\omega; r)] = -\partial_u \mathrm{Im}[F(\omega; r)]$. These imply that $\partial_\omega \mathrm{Re}[F(\omega; r)] = 0$ is equivalent to $\partial_\omega F(\omega; r) = 0$. At the saddle point $\omega(r)$, expanding in a Taylor series around $\omega(r)$ up to third order in u and v, and using the fact that the real and imaginary parts of $F(\omega; r)$ are even and odd functions of u respectively, we have $\mathrm{Re}[F(\omega; r)] = \mathrm{Re}[F(\omega(r); r)] + A(v^2 - u^2) + B(3u^2 v - v^3)$, and $\mathrm{Im}[F(\omega; r)] = -2Auv + B(3uv^2 - u^3)$ plus higher order terms in u and v. Since $\mathrm{Re}[F(\omega; r)]$ has a minimum on the imaginary axis at $\omega(r)$, with a non-zero second derivative, it follows that $A > 0$. Restricting ourselves to the case $v = 0$, we see that $\mathrm{Re}[F(\omega; r)] = \mathrm{Re}[F(\omega(r); r)] - Au^2$ and $\mathrm{Im}[F(\omega; r)] = -Bu^3$ up to cubic order in u. □

Let L_1 be the straight line in the ω plane parallel to the real axis and running through $\omega(r)$, and let $L_1(u_0)$ be the part of L_1 for which $|\mathrm{Re}(\omega)| < u_0$. Then, we have seen that $L_1(u_0)$ satisfies conditions (a), (b) and (c) required of the contour C in Theorem 2. We now verify condition (d):

Lemma 4

$$\mathrm{Abs}[F(\omega; r)] < \mathrm{Abs}[F(j\mathrm{Im}(\omega); r)] \qquad \text{for } \mathrm{Re}(\omega) \neq 0. \tag{32}$$

Proof: This result is obvious from the definition $F(\omega; r) = \int dx Z(x; r) \exp[-j\omega I(x; r)]$. Since $I(x; r)$ is real, and $Z(x; r)$ is real and positive, $F(\omega; r) \leq \int dx Z(x; r) \mathrm{Abs}(\exp[-j\omega I(x; r)])$, which is equal to $\int dx Z(x; r) \exp[\mathrm{Im}(\omega) I(x; r)] = F(j\mathrm{Im}(\omega); r) = \mathrm{Abs}[F(j\mathrm{Im}(\omega); r)]$. By considering the integral as a summation over x of real positive terms $Z(x; r) \exp[\mathrm{Im}(\omega) I(x; r)]$, each weighted by a complex weight $\exp[-j\mathrm{Re}(\omega) I(x; r)]$, we see that the magnitude of $F(\omega; r)$ is maximum when all the terms add up in phase, which occurs only when $\mathrm{Re}(\omega) = 0$. Thus the inequality is an equality only if $\mathrm{Re}(\omega) = 0$. □

Thus the contour $L_1(u_0)$ satisfies all the conditions of Theorem 2. Applying this theorem yields

$$\int_{L_1(u_0)} d\omega [F(\omega;r)]^d \sim \sqrt{\frac{2\pi}{d}} [F(\omega(r);r)]^d \left\{ -\frac{\partial^2 \mathrm{Re}[F(\omega;r)]}{\partial \mathrm{Re}(\omega)^2}\Big|_{\omega=\omega(r)} \frac{1}{F(\omega(r);r)} \right\}^{-1/2} \tag{33}$$

for any u_0.

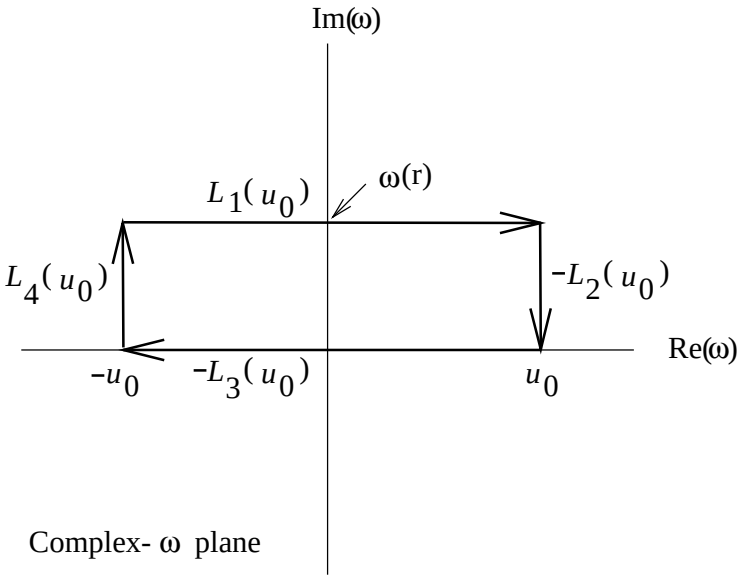

Figure 4: Contours used in the evaluation of the ω integral.

In order to complete the evaluation of the ω integral, we show that the integral over the real axis is also equal to the right hand side of Eq. (33) in the large d limit. Let L_3 denote the real axis for ω, and $L_3(u_0)$ be the part of the real axis for which $|\omega| < u_0$. Also, let $L_2(u_0)$ and $L_4(u_0)$ be the vertical straight line segments at $\mathrm{Re}(\omega) = u_0$ and $-u_0$ respectively that connect $L_1(u_0)$ and $L_3(u_0)$, so that $L_1(u_0) - L_2(u_0) - L_3(u_0) + L_4(u_0)$ is a closed contour. (See Fig. 4. The negative signs arise from the fact that, in the closed contour, L_2 and L_3 are oriented backwards.) By inspection of Eq. (29), we see that the function $F(\omega;r)$ has no singularities in the complex ω-plane. The only potential singularity is a pole at $1 - j\omega = 0$, but this is cancelled by a factor of $1 - j\omega$ in the numerator, as can be seen by substituting $\omega = -j + \delta$ in Eq. (29). Since by Cauchy's theorem the integral of an analytic function on a closed contour in the complex plane that does not enclose any singularities is zero, it follows that the integral of $[F(\omega;r)]^d$ over $L_3(u_0)$ is equal to the integral over $L_1(u_0) + L_4(u_0) - L_2(u_0)$. By inspection of Eq. (29), we see that if u_0 is large, then (provided that $\mathrm{Abs}[\omega(r)]$ is finite) the integral over $L_2(u_0)$ and $L_4(u_0)$ are dominated by the term $2(1-r)\exp[-j\omega\log(2r)]$ in $F(\omega;r)$. This is in the sense that the integrals over $L_2(u_0)$ and $L_4(u_0)$ are bounded by $[a(u_0)]^d$ times the integral of $\{2(1-r)\exp[-j\omega\log(2r)]\}^d$,

where $a(u_0)$ tends to unity as $u_0 \to \infty$. The maximum modulus of $2(1-r)\exp[-j\omega\log(2r)]$ over $L_2(u_0)$ and $L_4(u_0)$ is $2(1-r)$ if $j\omega(r) > 0$ and $2(1-r)\exp[-j\omega(r)\log(2r)]$ if $j\omega(r) < 0$. In the second case, from Eq. (28) we see that $2(1-r)\exp[-j\omega(r)\log(2r)]$ is just the first term in $F(\omega(r); r)$. As can be seen by inspection (and as discussed in the proof of Lemma 4), all the terms in the right hand side of Eq. (28) are real and positive for $\omega = \omega(r)$, so that $F(\omega(r); r) > 2a(u_0)(1-r)\exp[-j\omega(r)\log(2r)]$ for sufficiently large u_0. (Here the criterion for u_0 to be considered large is *independent* of d.) For sufficiently large u_0, the right hand side of Eq. (33) thus dominates the integral of $[F(\omega; r)]^d$ over $L_3(u_0)$ and $L_4(u_0)$ in the limit of large d (neither the $1/\sqrt{d}$ factor in Eq. (33) nor the factor of $|\omega(r)|$ from the length of $L_3(u_0)$ and $L_4(u_0)$ being large enough to compete with the exponential growth). If on the other hand $j\omega(r) > 0$, the integrand is bounded over $L_2(u_0)$ and $L_4(u_0)$ by $[2(1-r)a(u_0)]^d$. Again, if $F(\omega(r); r) > 2(1-r)$ (as is shown to be the case in Fig. 5), for sufficiently large u_0 this is negligible compared to Eq. (33) as $d \to \infty$.

Thus for large (but d-independent) u_0, the large d limit of the integral of $[F(\omega; r)]^d$ over $L_3(u_0)$ is given by Eq. (33). What remains is to evaluate the integral over $L_3 - L_3(u_0)$, and show that this is negligible for large u_0 in the $d \to \infty$ limit. Expanding $[F(\omega; r)]^d$ in a binomial series using Eq. (29) and arranging terms in increasing powers of $1/(1-j\omega)$ we have

$$[F(\omega; r)]^d = \sum_{p=0}^{d} \binom{d}{p} \cdot \{2(1-r)\exp[-j\omega\log(2r)]\}^{(d-p)} \cdot \{4r\exp[-j\omega\log(2r)] - 1 - r\exp[-j\omega\log(r)]\}^p \cdot \left(\frac{1}{1-j\omega}\right)^p. \quad (34)$$

For all terms except the first two, we can bound the integrand by its modulus. Recognizing that the integral of $1/(1+\omega^2)^{n/2}$ from u_0 to infinity is less than $u_0^{(1-n)}$ for $n > 1$, we can bound the sum of all these terms by $(5r+1)[2(1-r)]^{(d-1)}\{\exp[td] - 1 - td\}/t$, where $t = (5r+1)/[2(1-r)u_0]$. This is a monotonically decreasing function of u_0. Since $F(\omega(r); r) > 2(1-r)e^t$ for sufficiently large u_0, it follows that we can neglect the sum of all the terms except the first two as $d \to \infty$.

The integral of the first two terms has to be dealt with separately: the integral of $[2(1-r)]^d \exp[-jd\omega\log(2r)]$ over $|\omega| > u_0$ is bounded by $2\pi[2(1-r)]^d/\{d\log(2r)\}$ (unless $r = 1/2$, which is discussed separately later), which can be neglected. The second term is $[2(1-r)]^{(d-1)}$ multiplied by

$$\frac{d}{1-j\omega}\Big\{4r\exp[-j\omega d\log(2r)] - \exp[-j\omega(d-1)\log(2r)] - r\exp[-j\omega\log(r) - j\omega(d-1)\log(2r)]\Big\}. \quad (35)$$

By adding a similar contribution from $[F(-\omega; r)]^d$, we get expressions of the form $\Theta([2(1-r)]^{(d-1)}\sin[A(r)\omega]\omega/(1+\omega^2))$, which can be dealt with like the first term, and

$\Theta([2(1-r)]^{(d-1)}\cos[A(r)\omega]/(1+\omega^2))$, which can be dealt with like all the other terms, and both are negligible as $d \to \infty$.

Thus, to within numerical factors, Eq. (25) can be expressed as

$$\overline{N} = \Theta\left(\sqrt{d}\int_{\frac{1}{2}}^{1} dr[F(\omega(r);r)]^d\right) \tag{36}$$

where $F(\omega(r);r)$ can be evaluated numerically for any r by searching for $\omega(r)$ along the imaginary axis, and finding the value of F there.[1] Fig. 5 shows a plot of this quantity made using Mathematica.

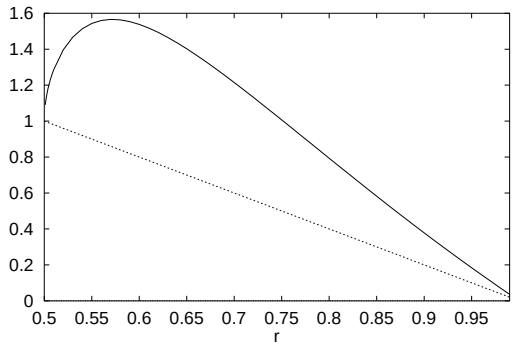

Figure 5: $F(\omega(r);r)$ vs. r. The straight line corresponding to $2(1-r)$ is everywhere below it.

The integral in Eq. (36), having a d in the exponent, can again be done by the method of steepest descent. This time the procedure is easier, since $F(\omega(r);r)$ is real, so that the analytic continuation to the complex plane carried out for the ω integral is not necessary: $\mathrm{Im}(r) = 0$ is manifestly a 'contour' along which $\mathrm{Arg}[F]$ is constant, and, as can be seen from Fig. 5, F has a global maximum with non-zero second derivative within the domain of r between $\frac{1}{2}$ and 1 so that the conditions of Theorem 2 are satisfied. For large d, the integral is again dominated by the point where $F(\omega(r);r)$ attains a maximum as a function of r over the domain $\frac{1}{2}$ to 1. This can be evaluated numerically. We thus obtain, to within

[1]Note that our evaluation of the integral over ω by the method of steepest descent is only valid for $\frac{1}{2} < r < 1$. For $r = \frac{1}{2}$, the integral is *not well defined*. This is because for all points $\mathbf{x}$ satisfying $0.5 \leq x_i \leq 1.5$, which is a finite fraction of the volume of the hypercube H, the radius of the nearest neighbor ball that is centered around the point is $\frac{1}{2}$. Thus the probability density that the radius of the nearest neighbor ball is r has a δ-function at $r = \frac{1}{2}$. However, since the weight of this δ-function is $\frac{1}{2}^d$, and the number of cells intersected for any point in this central region is 2^d, the contribution this singularity makes to $\overline{N}$ is $(\frac{1}{2})^d \cdot 2^d = 1$, which can be neglected compared to the eventual answer, obtained in Eq. (37). Although the method of steepest descent does not work for $r = 1$ too, we know that there is no δ-function in the probability density that the radius of the nearest neighbor ball is r at $r = 1$, so that the contribution to $\overline{N}$ from r *exactly* equal to 1 is zero.

a constant factor,

$$\overline{N} = \Theta\left([F(\omega(r_M); r_M)]^d\right) = \Theta\left((1.56594\ldots)^d\right) \tag{37}$$

where r_M is the point where $F(\omega(r); r)$ has a maximum, and is found numerically to be $r_M = 0.5715 \pm 0.0005$.

In order to find the multiplicative prefactor in Eq. (37), we have to include

(a) the factor of $\sqrt{2\pi}|f''(z_0)/f(z_0)|^{-1/2}$ in Eq. (30) that each of our two applications of the steepest descent method induces, and

(b) the derivative $\partial_r \hat{V}$ that was dropped in passing from Eq. (23) to Eq. (25).

The first of these simply leads to a nontrivial function $g(z)$ in the second application of Theorem 2. The second factor is a little more involved. However, the derivative $\partial_r \hat{V}$ is the sum of d terms $\partial_r \hat{I}_l$, where l ranges from 1 to d. By carrying out the x_l integral separately from all the $x_{i\neq l}$ integrals in each term, we obtain a modified version of Eq. (27):

$$\overline{N} \sim d \int_{\frac{1}{2}}^{1} dr \int_{-\infty}^{\infty} \frac{d\omega}{2\pi} [F(\omega; r)]^{d-1} \tilde{F}(\omega; r) \tag{38}$$

where $\tilde{F}$ comes from the x_l integration. This expression satisfies the conditions of Theorem 2 (with a nontrivial $g(z)$), and can be evaluated as above. By this method, we have calculated the multiplicative prefactor of Eq. (37) to be approximately 1.02.

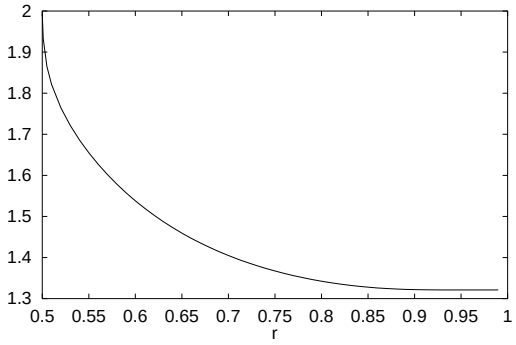

Figure 6: Evidence for the Monotonicity conjecture: $\lim_{d\to\infty}[E[N \mid u = r]]^{1/d}$ vs. r.

Remark: The procedure that we have carried out above can be carried out (more easily) to evaluate the probability density $\rho(r)$ that the radius of the unit intersection ball is r; the only differences are those resulting from the fact that the factor of Z_i in Eq. (27) is missing. Since the expression in Eq. (36) can be written as $\overline{N} = \int dr \rho(r) \overline{N}(r)$, and the corresponding expression one would get in evaluating the probability density is $1 = \int dr \rho(r)$, one can obtain the $\overline{N}(r)$, which is the expected number of cells overlapped by the unit intersection ball for

points where the unit intersection ball has radius r. Using Mathematica as in Fig. 5, the d-th root of this quantity was computed for various values of r and is plotted in Fig. 6 (here $m = 2$). If one believes that the plot in Fig. 6 is a faithful representation of the function, then one may regard this as a sort of "numerical proof" of the monotonicity conjecture in the large d limit for $m = 2$.

6 Extensions and Generalizations

In this section, we first outline how the results of the previous section would be generalized to $m > 2$, and present results from a numerical evaluation similar to that carried out in the previous section. We then consider the effect of the fact that the nearest neighbor ball around a query point will only have unit intersection volume on the *average*, and show that fluctuations around this average can only change the constant prefactor of our earlier expression for $\overline{N}$.

For general m, the x-integrals in Eq. (22) and thereafter are over the interval 0 to m, with a prefactor of $1/m$ for each x-integral. (Note that we used the symmetry under $x_i \to 2 - x_i$ for the case of $m = 2$ to reduce the range of the x-integrals to $[0, 1]$.) Eq. (24) is modified, to

$$\hat{I}_i(x_i; r) = \begin{cases} \log(x_i + r) & \text{for } x_i \leq r \\ \log(2r) & \text{for } m - r \geq x_i \geq r \\ \log(m + r - x_i) & \text{for } x_i > m - r. \end{cases} \tag{39}$$

This, however, does not affect the transition to Eq. (25). Eq. (26) is also changed, since $Z_i(x_i; r) = 3$ for $k + r > x_i > k + 1 - r$ for k an integer, $1 \leq k \leq m - 2$. Similarly one can determine when $Z_i(x_i; r)$ takes the values 1 and 2. (Recall that r lies between $1/2$ and 1, so that it is never possible for Z_i to be greater than 3.) After some calculation, we obtain the following general form for Eq. (29),

$$\begin{aligned} F(\omega; r) = & \\ & (2r + 1) \exp[-j\omega \log(2r)] + \\ & \tfrac{2}{m} \left(\left\{ (1 - 4r) + \tfrac{4r}{1 - j\omega} \right\} \exp[-j\omega \log(2r)] - \right. \\ & \left. \tfrac{1}{1 - j\omega} - \tfrac{r}{1 - j\omega} \exp[-j\omega \log(r)] \right). \end{aligned} \tag{40}$$

The numerical analysis after Eq. (29) can be carried out from here for any value of m. The results are given in Table 2, which shows the expected number of cells overlapped by the unit intersection ball, for different values of m.

We now consider possible modifications to our result, Eq. (37), due to the fact that the nearest neighbor ball around any query point has unit intersection volume with the

m	$E[N]$
2	$\Theta(1.566^d)$
3	$\Theta(1.715^d)$
4	$\Theta(1.788^d)$
5	$\Theta(1.831^d)$
10	$\Theta(1.916^d)$
100	$\Theta(1.992^d)$

Table 2: Expected number of cells overlapped by the unit intersection ball.

hypercube H only *on average*. For simplicity, we return to the case of $m = 2$, although the results can be extended to any m. We show that Eq. (37) is valid, to within constant factors, even without this assumption. We only require that the probability density that the nearest neighbor ball has intersection volume λ be *independent* of the location of $\mathbf{x}$. This weaker assumption is satisfied in the limit for large dimension, for points uniformly distributed in the unit hypercube. In this case, it is easily verified that the probability that the nearest neighbor ball has intersection volume λ, has the Poisson density, $\exp[-\lambda]$.

Consider an ensemble of hypercubes with points distributed randomly and uniformly in their interior, with unit mean density. We assume that, by averaging over query points, $\overline{N}$ is the same for every member of the ensemble, even though for a *particular* query point $\mathbf{x}$ it is possible for $N(\mathbf{x})$ to have significant fluctuations from one member to another in the ensemble. For the ensemble average of $\overline{N}$, the expression in Eq. (37) for $\overline{N}$ can be replaced by

$$\overline{N} \sim \int d\lambda \exp[-\lambda] \overline{N}(\lambda) \tag{41}$$

where $\overline{N}(\lambda)$ is the expected number of cells when the intersection volume is λ (the quantity evaluated in Eq. (37) is $\overline{N}(\lambda = 1)$).

We now use the fact that $\overline{N}(\lambda)$ increases monotonically with λ, so that $\overline{N}(\lambda < 1) < \overline{N}(1)$, and $\overline{N}(\lambda)$ is equal to 2^d as $\lambda \to \infty$. This implies that $\int_{d \log 2}^{\infty} d\lambda \exp[-\lambda] \overline{N}(\lambda) < 1$ and $\int_0^{1/d} d\lambda \exp[-\lambda] \overline{N}(\lambda) < \overline{N}(1)/d$, and we thus obtain

$$\int_{1/d}^{d \log 2} d\lambda \exp[-\lambda] \overline{N}(\lambda) < \overline{N} < \int_{1/d}^{d \log 2} d\lambda \exp[-\lambda] \overline{N}(\lambda) + 1 + \overline{N}(1)/d. \tag{42}$$

Both the extra terms in the upper bound for $\overline{N}$ are negligible compared to the lower bound for large d, so that $\overline{N}$ is asymptotically equal to the lower bound obtained for it in Eq. (42). We now evaluate this lower bound.

In evaluating $\overline{N}(\lambda)$, we have the following changes compared to $\overline{N}(1)$:

(a) there is an extra factor of $\exp[j\omega(\log\lambda)]$ in $[F(\omega;r)]^d$, arising from the shift in the δ-function,

(b) the integral over r now ranges from $\lambda^{1/d}/2$ to $\lambda^{1/d}$, and

(c) in going from Eq. (23) to (25), the derivative $\partial_r\hat{V}$ lies between $\frac{1}{2}\lambda^{-1/d}$ and $2\lambda^{-1/d}$.

Since at both the upper and the lower limit of the λ-integral in Eq. (42), $\lim_{d\to\infty}\log(\lambda)/d = 0$, r_M lies within the range of integration of r for all λ. Also, by expanding $\int d\omega[F(\omega;r)]^d$ around $\omega(r)$, it is easy to see that the change in $\omega(r)$ caused by the extra factor in (a) is $O((\log\lambda)/d)$. The result of this shift in $\omega(r)$ is a factor of $\exp[O((\log\lambda)^2/d)]$ in $[F(\omega(r);r)]^d$, which tends to unity for all λ in the range of integration as $d\to\infty$. Thus $\overline{N}(\lambda)$, which is dominated by the region around r_M, is given by $\overline{N}(1)\cdot\lambda^c$, where $c = -\mathrm{Im}[\omega(r_M)]$. Integrating over λ with an $\exp[-\lambda]$ weight factor, this gives the same result as in Eq. (37), but with a different multiplicative factor. The multiplicative factor can be obtained as in Section 5, and is approximately equal to 0.90. We summarize our main result:

Theorem 3 *Given a set of 2^d points uniformly distributed in a d-dimensional hypercube, and a query point also from the uniform distribution, the expected number of cells visited by the bucketing algorithm, $\overline{N}_d$, satisfies*

$$\lim_{d\to\infty}\frac{\overline{N}_d}{(0.90\ldots)\cdot(1.56594\ldots)^d} = 1.$$

7 Experimental Results

In this section, we present experimental results which show that even for low dimensions (≤ 16), our analysis yields good bounds. Fig. 7 shows the average number of cells examined by the bucketing algorithm and the optimized k-d tree algorithm, when the dimension d ranges from 1 to 16, and the number of data points is 2^d. In each case the number of cells examined is averaged over 10 different experiments, where each experiment is performed with a different set of data points and averaged over 2,500 query points. Both data and query points are chosen from the uniform distribution. The figure shows that Theorem 3 provides an excellent fit to the number of points visited by the two algorithms. The fit is slightly worse for the k-d tree algorithm than for the bucketing algorithm. This could be because the k-d tree partition is not perfectly regular. It could also be because the k-d tree algorithm does not necessarily visit the cells in increasing order of distance, rather in an order determined by the structure of the tree. These factors could lead to a small overhead.

It is also clear from Fig. 7 that the cell based algorithms examine significantly fewer data points than exhaustive search, when the number of data points is on the order of 2^d. In view of the greater overhead of the cell algorithms, we were interested in determining how

much of this advantage translates into an actual gain in the running time. We conducted these experiments in dimension 16, for the same data and query sets as used to plot Fig. 7. For this study we used the efficient implementation of k-d trees as described by Arya and Mount[3], and stored 4 data points in each leaf cell which helped to reduce the overhead. We found that the k-d tree algorithm ran over 37 times faster than brute force search on a Sun SPARC 10 system. We also conducted similar experiments for the Euclidean distance metric. Although we do not have theoretical results for this metric, our experiments indicate that boundary effects are important and lead to some speedup but not as much as for the L_∞ metric. In dimension 16, we observed that the k-d tree algorithm ran about 4 times faster than exhaustive search.

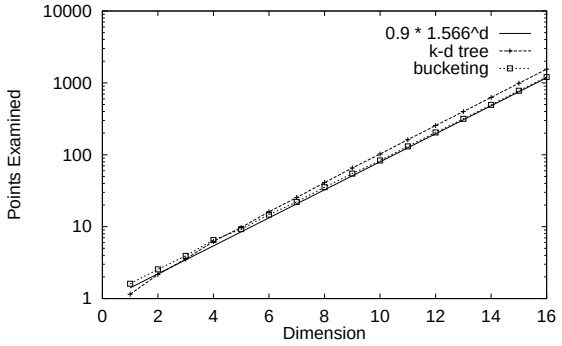

Figure 7: Average number of points examined by the bucketing and k-d tree algorithms; the total number of data points in dimension d is 2^d.

8 Conclusions

Previous analysis have neglected the effects of the boundary in order to simplify the analysis. However, there is plenty of evidence of the importance of boundary effects in realistic instances in high dimensions. We have presented analysis that takes these effects into account, thus providing a significantly more accurate analysis.

The main limitation of our analysis is that it applies only to the L_∞ metric and the uniform distribution. It would be nice to extend these results to other distance norms and other distributions. Another interesting open problem is to prove the monotonicity conjecture.

References

[1] G. B. Arfken. *Mathematical methods for physicists.* Academic Press, 3rd edition, 1985.

[2] S. Arya and D. M. Mount. Approximate nearest neighbor queries in fixed dimensions. In *Proceedings of the 4th ACM-SIAM Symposium on Discrete Algorithms*, pages 271–280, 1993.

[3] S. Arya and D. M. Mount. Algorithms for fast vector quantization. In J. A. Storer and M. Cohn, editors, *Proc. of DCC '93: Data Compression Conference*, pages 381–390. IEEE Press, 1993.

[4] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman and A. Wu. An optimal algorithm for approximate nearest neighbor searching. In *Proceedings of the 5th ACM-SIAM Symposium on Discrete Algorithms*, pages 573–582, 1994.

[5] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, September 1975.

[6] J. L. Bentley, B. W. Weide, and A. C. Yao. Optimal expected-time algorithms for closest point problems. *ACM Transactions on Mathematical Software*, 6(4):563–580, 1980.

[7] K. L. Clarkson. An algorithm for approximate closest-point queries. In *Proceedings of the 10th Annual ACM Symposium on Computational Geometry*, pages 160–164, 1994.

[8] J. G. Cleary. Analysis of an algorithm for finding nearest neighbors in Euclidean space. *ACM Transactions on Mathematical Software*, 5(2):183–192, June 1979.

[9] J. H. Friedman, J. L. Bentley, and R.A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, September 1977.

[10] W. Feller. *An introduction to probability theory and its applications*, volume 1. Wiley Eastern, 3rd edition, 1968.

[11] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic, 1991.

[12] G. Goertzel and N. Tralli. *Some mathematical methods of physics.* McGraw-Hill, 1960.

[13] R. L. Rivest. On the optimality of Elias's algorithm for performing best-match searches. In *Information Processing*, pages 678–681. North Holland Publishing Company, 1974.

[14] R. L. Sproull. Refinements to nearest-neighbor searching in k-dimensional trees. *Algorithmica*, 6, 1991.

[15] T. Welch. Bounds on the information retrieval efficiency of static file structures. Technical Report 88, MIT, June 1971.