When Is “Nearest Neighbor” Meaningful?

Kevin Beyer **Jonathan Goldstein** **Raghu Ramakrishnan** **Uri Shaft**

CS Dept., University of Wisconsin-Madison
1210 W. Dayton St., Madison, WI 53706
`email: beyer,jgoldst,raghu,uri@cs.wisc.edu`
Fax: (608)262-9777

1 Introduction

In recent years, there has been much interest in finding efficient solutions to the “nearest neighbor” (NN) problem, defined as follows: *Given a collection of data points and a query point in an m-dimensional metric space, find the data point that is closest to the query point.* Particular interest has centered on solving this problem in high dimensional spaces, which arise from techniques that approximate (see [15]) complex data—such as images (e.g. [7, 17, 18, 12, 18, 14, 16, 9, 3]), sequences (e.g. [2, 1]), video (e.g. [7]), and shapes (e.g. [7, 19, 16, 13])—with long “feature” vectors. Similarity queries are performed by taking a given complex object, approximating it with a high dimensional vector to obtain the query point, and determining the data point closest to it in the underlying feature space.

In this paper, we study the nearest neighbor problem and make the following contributions:

- We show that under certain conditions (in terms of data and query distributions, or workload), as dimensionality increases, the distance to the nearest neighbor approaches the distance to the farthest neighbor. In other words, virtually every data point is as good as any other, and slight perturbations to the query point would result in another data point being chosen as the nearest neighbor. Our result characterizes the problem itself, rather than specific algorithms that address the problem. This observation places some fundamental limits upon current approaches to multimedia similarity search based upon high-dimensional feature vector representations. In addition, our observations apply equally to the k-nearest neigbor variant of the problem.

- To provide a practical perspective, we present empirical results based on a wide range of synthetic distributions showing that the distinction between nearest and farthest neighbors blurs with as few as 15 dimensions.

- We identify special workloads for which the concept of nearest neighbor continues to be meaningful in high dimensionality. We present a reformulation of the k-nearest neighbors problem that allows a user to specify a query in terms of the meaningfulness of the desired answers. Since, in our reformulation, the size of the answer set varies, we provide a mechanism for estimating the size of the answer set.

- We observe that the literature on nearest neighbor processing techniques fails to compare their techniques to linear scans. Furthermore, we can infer from their data that a linear scan always outperforms their techniques in high dimensionality. This is unsurprising as the workloads used to evaluate these techniques are in the class of “badly behaving” workloads identified by our results. Our results underscore the point that evaluation of a technique for nearest-neighbor search should be based on meaningful workloads.

In summary, our results suggest that a fundamental re-thinking of nearest neighbor approaches and high-dimensional indexing algorithms is called for; we supplement our theoretical results with experimental data and a careful discussion.

2 On the Significance of “Nearest Neighbor”

The NN problem involves determining the point in a dataset that is nearest to a given query point (see Figure 1). It is frequently used in Geographical Information Systems (GIS), where points are associated with some geographical location (e.g., cities). A typical NN query is: “What city is closest to my current location?”

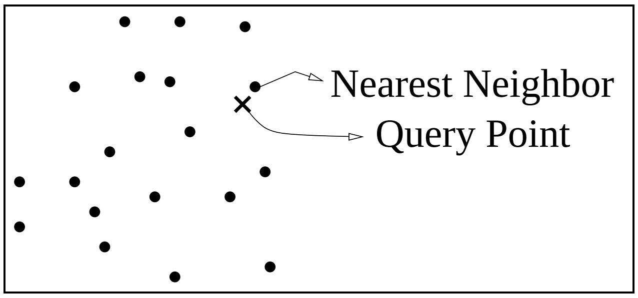

Figure 1: Query point and its nearest neighbor.

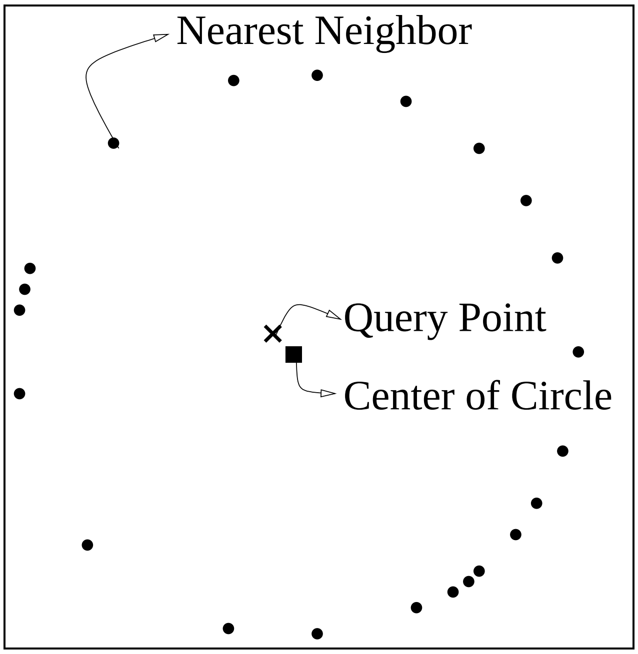

Figure 2: Another query point and its nearest neighbor.

While it is natural to ask for the nearest neighbor, there is not always a meaningful answer. For instance, consider the scenario depicted in Figure 2. Even though there is a well-defined nearest neighbor, the difference in distance between the nearest neighbor and any other point in the dataset is very small. Since the difference in distance is so small, the utility of the answer in solving concrete problems (e.g. minimizing travel cost) is very low.

While the scenario depicted in Figure 2 is very contrived for a geographical database, we show that it is the norm for a broad class of data distributions in high dimensionality. To establish this, we will examine the number of points that fall into a query sphere enlarged by some factor ε (see Figure 3). If few points fall into this enlarged sphere, it means that the data point nearest to the query point is separated from the rest of the data in a meaningful way. On the other hand, if many (let alone most!) data points fall into this enlarged sphere, differentiating the "nearest neighbor" from these other data points is meaningless if ε is small. We show that in many situations, for **any** fixed $\varepsilon > 0$, as dimensionality rises, the number of points that fall in the enlarged query region becomes the size of the dataset.

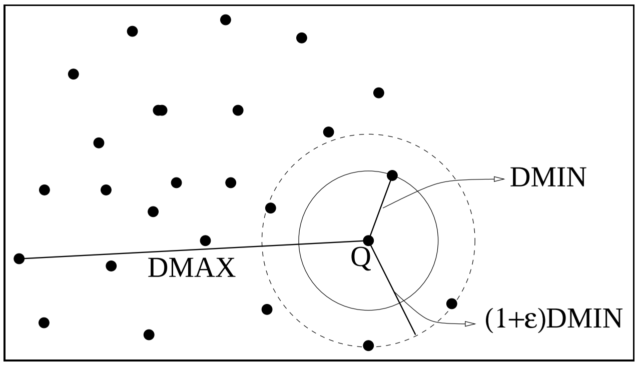

Figure 3: Illustration of query region and enlarged region.

3 NN in High-Dimensional Spaces

We use the following notation in the rest of the paper:

A vector: $\vec{x}$

Probability of an event E: $P\left[\mathrm{E}\right]$.

Expectation of a random variable X: $\mathcal{E}\left[X\right]$.

Variance of a random variable X: $\mathrm{var}\left(X\right)$.

IID: Identical and independent in distribution. (This phrase is used with reference to the values assigned to a collection of random variables.)

$\vec{X} \sim F$: A random variable $\vec{X}$ that takes on values following the distribution F.

Superscripts: We note that superscripts such as A^b always refer to exponents (i.e., A^b should be read as A raised to the power b). Superscripts are never used as an alternative to subscripts in this paper.

Given a dataset and a query point, we want to analyze how much the distance of the nearest neighbor differs from the distance of other data points. We do this by evaluating the number of points that lie within the hypersphere centered about the query point with radius ε larger than the distance between the query point and the NN, as illustrated in Figure 3.

We now introduce several terms used in stating our result formally.

Definition 1

n: *The number of points in the dataset.*

m: *The number of dimensions (i.e., a point is a vector with arity m).*

$F_{\mathbf{data}}$: *The m-dimensional distribution of data points.*

$F_{\mathbf{query}}$: *The m-dimensional distribution of query points.*

$\vec{Q} \sim F_{\mathbf{query}}$: *The query point, chosen according to the distribution F_{query}.*

$\vec{X}_1, \ldots, \vec{X}_n \sim F_{\mathbf{data}}$: *The n data points in the dataset.*

p: *The p value for the chosen L_p metric (e.g. $p = 2$ for L_2, the Euclidean distance metric).*

$d(\vec{Q}, \vec{X})$: *The function d gives the L_p distance between the points $\vec{Q}$ and $\vec{X}$.*

$\mathbf{DMIN}_m$: *The distance from the query point to its nearest neighbor.*

$\mathbf{DMAX}_m$: *The distance from the query point to the farthest data point.*

ε: *The radius increase about the query used to evaluate the distinction of the nearest neighbor relative to other points in the dataset.*

$J_{m,\varepsilon}$: *The number of data points ($1 \leq J_{m,\varepsilon} \leq n$) in the hypersphere with radius $(1+\varepsilon)DMIN_m$ centered at the query point.*

Our main result is presented below. In essence, it states that as dimensionality increases, for a broad class of data distributions almost all data points are about as close to the query point as the nearest neighbor. More precisely, for any fixed enlargement factor ε, the probability that $J_{m,\varepsilon}$ (the number of data points inside the enlarged query region) is n (the total number of points in the dataset) converges to 1 as dimensionality increases ($m \to \infty$).

Theorem 1 *Let $\vec{Q}$ be a query point chosen independently of all data points $\vec{X}_i$ where $1 \leq i \leq n$, and let $D_m = d(\vec{Q}, \vec{X})$ be a random variable that follows the distribution of distance between query point $\vec{Q} \sim F_{query}$ and data point $\vec{X} \sim F_{data}$.*

If $\lim_{m\to\infty} \left(var\left(D_m^p\right) / (\mathcal{E}\left[D_m^p\right])^2 \right) = 0$, then for any $\varepsilon > 0$: $\lim_{m\to\infty} P\left[J_{m,\varepsilon} = n\right] = 1$.

Proof :
Given independence between $\vec{Q}$ and $\vec{X}_i$ where $1 \leq i \leq n$, and that $\lim_{m\to\infty} \left(\mathrm{var}\left(D_m^p\right) / (\mathcal{E}\left[D_m^p\right])^2 \right) = 0$, we need to show that for any $\varepsilon > 0$, $\lim_{m\to\infty} P\left[J_{m,\varepsilon} = n\right] = 1$.

Note that by Definition 1,

$$\mathrm{DMIN}_m = \min\{d(\vec{Q}, \vec{X}_i) \mid 1 \leq i \leq n\}$$

$$\mathrm{DMAX}_m = \max\{d(\vec{Q}, \vec{X}_i) \mid 1 \leq i \leq n\}$$

To show $\lim_{m\to\infty} P\left[J_{m,\varepsilon} = n\right] = 1$, we will identify small intervals $[l_m, h_m] \subset (0, \infty)$ (one interval for each m, as defined in Equations 2 and 3) such that $h_m/l_m = 1 + \varepsilon$, and we will show that

$$\lim_{m\to\infty} P\left[l_m \leq \mathrm{DMIN}_m \leq \mathrm{DMAX}_m \leq h_m\right] = 1 \qquad (1)$$

Step 1. In this step we prove that in order to establish the consequent of the theorem, it is sufficient to show that Equation 1 holds. By definition, $J_{m,\varepsilon} = n$ is equivalent to the statement that every data point is within distance $(1+\varepsilon)\mathrm{DMIN}_m$ from the query point (i.e., all the n data points are in the enlarged query region). Therefore,

$$\lim_{m\to\infty} P\left[J_{m,\varepsilon} = n\right] = \lim_{m\to\infty} P\left[\mathrm{DMAX}_m \leq (1+\varepsilon)\mathrm{DMIN}_m\right]$$

Note that if $l_m \leq \mathrm{DMIN}_m \leq \mathrm{DMAX}_m \leq h_m$ holds, then $\mathrm{DMAX}_m/\mathrm{DMIN}_m \leq h_m/l_m = 1+\varepsilon$. Therefore,

$$\lim_{m\to\infty} P\left[\mathrm{DMAX}_m \leq (1+\varepsilon)\mathrm{DMIN}_m\right] \geq$$

$$\lim_{m\to\infty} P\left[l_m \leq \mathrm{DMIN}_m \leq \mathrm{DMAX}_m \leq h_m\right]$$

This means that showing Equation 1 holds is sufficient.

Step 2. In this step we define the intervals $[l_m, h_m]$. We know that the distances $\left(d(\vec{Q}, \vec{X}_i)\right)^p$ $(1 \leq i \leq n)$ are near $\mathcal{E}\left[D_m^p\right]$ since the condition of the theorem says that the variance of distances is small compared to the mean squared distance. Therefore, we will use small intervals $[l_m, h_m]$ that contain $(\mathcal{E}\left[D_m^p\right])^{1/p}$. For $0 < \delta < 1$ let us define:

$$l_m^p = (1-\delta)\mathcal{E}\left[D_m^p\right] \qquad (2)$$

$$h_m^p = (1+\delta)\mathcal{E}\left[D_m^p\right] \qquad (3)$$

We need to find $0 < \delta < 1$ s.t. $h_m/l_m = 1+\varepsilon$. The solution is

$$\delta = \frac{(1+\varepsilon)^p - 1}{(1+\varepsilon)^p + 1}$$

Step 3. From Step 1 we know that it is sufficient to show Equation 1. We now use the definition of DMIN_m and DMAX_m. The minimum and maximum of a set are inside an interval if and only if all items of the set are inside that interval. In this case, the items of the set are $d(\vec{Q},\vec{X}_i)$ (where $1 \leq i \leq n$).

$$\lim_{m\to\infty} P\left[l_m \leq \mathrm{DMIN}_m \leq \mathrm{DMAX}_m \leq h_m\right] =$$

$$\lim_{m\to\infty} P\left[\forall i : l_m \leq d(\vec{Q},\vec{X}_i) \leq h_m\right] =$$

$$\lim_{m\to\infty} P\left[\forall i : l_m^p \leq \left(d(\vec{Q},\vec{X}_i)\right)^p \leq h_m^p\right]$$

Step 4. Using the definition of l_m^p and h_m^p:

$$\lim_{m\to\infty} P\left[\forall i : l_m^p \leq \left(d(\vec{Q},\vec{X}_i)\right)^p \leq h_m^p\right] =$$

$$\lim_{m\to\infty} P\left[\forall i : (1-\delta)\mathcal{E}\left[D_m^p\right] \leq \left(d(\vec{Q},\vec{X}_i)\right)^p \leq (1+\delta)\mathcal{E}\left[D_m^p\right]\right]$$

$$= \lim_{m\to\infty} P\left[\forall i : \left|\left(d(\vec{Q},\vec{X}_i)\right)^p - \mathcal{E}\left[D_m^p\right]\right| \leq \delta\mathcal{E}\left[D_m^p\right]\right]$$

Step 5. Using the property $P\left[\forall i : A_i\right] = 1 - P\left[\exists i \text{ s.t. } \neg A_i\right]$ (Here, A_i stands for $\left|(d(\vec{Q},\vec{X}_i))^p - \mathcal{E}\left[D_m^p\right]\right| \leq \delta\mathcal{E}\left[D_m^p\right]$):

$$\lim_{m\to\infty} P\left[\forall i : \left|\left(d(\vec{Q},\vec{X}_i)\right)^p - \mathcal{E}\left[D_m^p\right]\right| \leq \delta\mathcal{E}\left[D_m^p\right]\right] =$$

$$1 - \lim_{m\to\infty} P\left[\exists i \text{ s.t. } \left|\left(d(\vec{Q},\vec{X}_i)\right)^p - \mathcal{E}\left[D_m^p\right]\right| > \delta\mathcal{E}\left[D_m^p\right]\right]$$

Step 6. Use the property $P\left[\exists i \text{ s.t. } B_i\right] \leq \sum_{i=1}^n P\left[B_i\right]$ (Here, B_i stands for $\left|(d(\vec{Q},\vec{X}_i))^p - \mathcal{E}\left[D_m^p\right]\right| > \delta\mathcal{E}\left[D_m^p\right]$):

$$1 - \lim_{m\to\infty} P\left[\exists i \text{ s.t. } \left|\left(d(\vec{Q},\vec{X}_i)\right)^p - \mathcal{E}\left[D_m^p\right]\right| > \delta\mathcal{E}\left[D_m^p\right]\right]$$

$$\geq 1 - \lim_{m\to\infty} \sum_{i=1}^n P\left[\left|\left(d(\vec{Q},\vec{X}_i)\right)^p - \mathcal{E}\left[D_m^p\right]\right| > \delta\mathcal{E}\left[D_m^p\right]\right]$$

Step 7. Observe that the query point $\vec{Q}$ is independent of the data points $\vec{X}_i$ (for all $1 \leq i \leq n$), and all $\vec{X}_i$'s have the same distribution. Hence, all the distance random variables $d(\vec{Q},\vec{X}_i)$ have the same distribution as D_m. Hence:

$$1 - \lim_{m\to\infty} \sum_{i=1}^n P\left[\left|\left(d(\vec{Q},\vec{X}_i)\right)^p - \mathcal{E}\left[D_m^p\right]\right| > \delta\mathcal{E}\left[D_m^p\right]\right] =$$

$$1 - \lim_{m\to\infty} \sum_{i=1}^n P\left[\left|D_m^p - \mathcal{E}\left[D_m^p\right]\right| > \delta\mathcal{E}\left[D_m^p\right]\right] =$$

$$1 - n \cdot \lim_{m\to\infty} P\left[\left(D_m^p - \mathcal{E}\left[D_m^p\right]\right)^2 > \delta^2\left(\mathcal{E}\left[D_m^p\right]\right)^2\right]$$

Step 8. Use Markov's inequality: for any positive random variable Y and for any constant $c > 0$, if $\mathcal{E}\left[Y\right]$ is defined then $P\left[Y > c\right] \leq \mathcal{E}\left[Y\right]/c$. (Here, Y stands for $\left(D_m^p - \mathcal{E}\left[D_m^p\right]\right)^2$, and c stands for $\delta^2\left(\mathcal{E}\left[D_m^p\right]\right)^2$.)

$$1 - n \cdot \lim_{m\to\infty} P\left[\left(D_m^p - \mathcal{E}\left[D_m^p\right]\right)^2 > \delta^2\left(\mathcal{E}\left[D_m^p\right]\right)^2\right] \geq$$

$$1 - n \cdot \lim_{m\to\infty} \frac{1}{\delta^2\left(\mathcal{E}\left[D_m^p\right]\right)^2}\mathcal{E}\left[\left(D_m^p - \mathcal{E}\left[D_m^p\right]\right)^2\right]$$

Step 9. By definition of variance we have $\mathrm{var}\left(D_m^p\right) = \mathcal{E}\left[\left(D_m^p - \mathcal{E}\left[D_m^p\right]\right)^2\right]$. Use also the condition of the theorem: $\lim_{m\to\infty} \mathrm{var}\left(D_m^p\right) / \left(\mathcal{E}\left[D_m^p\right]\right)^2 = 0$ to get:

$$1 - n \cdot \lim_{m\to\infty} \frac{1}{\delta^2\left(\mathcal{E}\left[D_m^p\right]\right)^2}\mathcal{E}\left[\left(D_m^p - \mathcal{E}\left[D_m^p\right]\right)^2\right] =$$

$$1 - n \cdot \lim_{m\to\infty} \frac{1}{\delta^2\left(\mathcal{E}\left[D_m^p\right]\right)^2}\mathrm{var}\left(D_m^p\right) = 1 - n\cdot 0 = 1$$

We have shown that

$$\lim_{m\to\infty} P\left[J_{m,\varepsilon} = n\right] =$$

$$\lim_{m\to\infty} P\left[\mathrm{DMAX}_m \leq (1+\varepsilon)\mathrm{DMIN}_m\right] \geq 1$$

Since $P\left[J_{m,\varepsilon} = n\right] \leq 1$ (because it is a probability) we have that

$$\lim_{m\to\infty} P\left[J_{m,\varepsilon} = n\right] =$$

$$= \lim_{m\to\infty} P\left[\mathrm{DMAX}_m \leq (1+\varepsilon)\mathrm{DMIN}_m\right] = 1 \quad (4)$$

■

Lemma 2 *Suppose that the same conditions used in Theorem 1 hold.*

If $\lim_{m\to\infty}\left(var(D_m^p)/(\mathcal{E}\left[D_m^p\right])^2\right) = 0$ *then the ratio* $DMAX_m/DMIN_m$ *converges in probability to the constant 1. That is, for any* $\varepsilon > 0$*, we have:*

$$\lim_{m\to\infty} P\left[\left|(DMAX_m/DMIN_m) - 1\right| > \varepsilon\right] = 0$$

Proof : From equation 4 in the proof of Theorem 1, we have that for any $\varepsilon > 0$:

$$\lim_{m\to\infty} P\left[\mathrm{DMAX}_m \leq (1+\varepsilon)\mathrm{DMIN}_m\right] = 1$$

DMIN_m is defined as a minimum of distances (which are positive), so $\mathrm{DMIN}_m \geq 0$. Also, $\mathrm{DMAX}_m \geq \mathrm{DMIN}_m$, so $\mathrm{DMAX}_m/\mathrm{DMIN}_m \geq 1$. Therefore,

$$P\left[\mathrm{DMAX}_m \leq (1+\varepsilon)\mathrm{DMIN}_m\right] = P\left[\frac{\mathrm{DMAX}_m}{\mathrm{DMIN}_m} \leq 1+\varepsilon\right]$$

$$= P\left[\left|\frac{\mathrm{DMAX}_m}{\mathrm{DMIN}_m} - 1\right| \leq \varepsilon\right] = 1 - P\left[\left|\frac{\mathrm{DMAX}_m}{\mathrm{DMIN}_m} - 1\right| > \varepsilon\right]$$

Thus, we have that for all $\varepsilon > 0$:

$$\lim_{m\to\infty} P\left[\left|\frac{\mathrm{DMAX}_m}{\mathrm{DMIN}_m} - 1\right| > \varepsilon\right] =$$

$$1 - \lim_{m\to\infty} P\left[\mathrm{DMAX}_m \leq (1+\varepsilon)\mathrm{DMIN}_m\right] = 1 - 1 = 0$$

■

In summary, there are two theoretical results of interest. The first is Theorem 1, which states that even though the nearest neighbor may be well-defined, if we look a little further away, we will find the rest of our data, rendering our determination of nearest neighbor highly questionable in utility.

Our second theoretical result, Lemma 2, makes the same point slightly differently. It says that the minimum and maximum distances from the query point to points in the dataset become closer and closer as dimensionality increases. Obviously, in such a situation, it becomes meaningless to identify a "nearest neighbor". Both results are independent of the method used to compute the nearest neighbor.

While these results are theoretically very interesting, there are two issues that must be addressed to determine their practical impact:

- How restrictive is the condition
$$\lim_{m\to\infty}\left(\mathrm{var}\left(D_m^p\right)/(\mathcal{E}\left[D_m^p\right])^2\right)=0$$
which is necessary for our results to hold?
- For situations in which the condition is satisfied, at what rate do distances between points become indistinct as dimensionality increases? In other words, at what dimensionality does the concept of "nearest neighbor" become meaningless?

Consider the condition

$$\lim_{m\to\infty}\left(\mathrm{var}\left(D_m^p\right)/(\mathcal{E}\left[D_m^p\right])^2\right)=0$$

Intuitively, it says that as we add dimensions and examine the resulting distribution of distances between queries and data, the standard deviation of the distance distribution must rise significantly more slowly than the expected value of the distance distribution. This makes perfect sense when one considers that we are using the condition to prove that as dimensionality increases, more and more data falls into our enlarged query region. Thus, the standard deviation of the distance distribution must increase at a slower rate than the distance to the nearest neighbor. To provide a better understanding of the restrictiveness of this condition, Sections 3.1 and 4 discuss scenarios that do and do not satisfy it.

The second issue is more difficult to tackle analytically, because many conservative approximations made in the proof are insignificant in the limit. (The approximations are in Steps 1, 6, and 8.) We therefore performed a set of simulations that examine the relationship between m and the ratio of minimum and maximum distances with respect to the query point. The results of these simulations are presented in Section 5.

3.1 Applicability of Our Result

This section analyses the applicability of Theorem 1 in several situations. This is done by determining, for each situation, whether the condition

$$\lim_{m\to\infty}\left(\mathrm{var}\left(D_m^p\right)/(\mathcal{E}\left[D_m^p\right])^2\right)=0$$

is satisfied.

3.1.1 IID Dimensions with Query and Data Independence

Suppose that the following assumptions are valid:

- The data distribution is identical and independent in each dimension.
- The query distribution is identical and independent in each dimension.
- The query point is chosen independently of the data points.

We will prove that the conditions of Theorem 1 are satisfied under these assumptions. Let $\vec{Q}$ and $\vec{W}$ be random variables such that $\vec{Q}\sim F_{\text{query}}$ and $\vec{W}\sim F_{\text{data}}$. To evaluate whether

$$\lim_{m\to\infty}\left(\mathrm{var}\left(D_m^p\right)/(\mathcal{E}\left[D_m^p\right])^2\right)=0$$

is satisfied, we apply the definition of distance between points. This leads to the following condition:

$$\lim_{m\to\infty}\frac{\mathrm{var}\left(\sum_{i=1}^m|Q_i-W_i|^p\right)}{\left(\mathcal{E}\left[\sum_{i=1}^m|Q_i-W_i|^p\right]\right)^2}=0 \tag{5}$$

Note that identical per dimension characteristics in our assumptions allow us to say that for all i, $|Q_i-W_i|^p$ is some random variable V_i, and all V_i's are IID with mean μ and variance σ^2. Thus, $\mathcal{E}\left[\sum_{i=1}^m V_i\right]=m\mu$. Since the dimensions in the query and data distributions are independent of one another, and the query and data distributions are independent,
$var\sum_{i=1}^m V_i=m\sigma^2$.

Plugging these values back into our original limit produces convergence to zero, and thus meets our condition. Therefore, the assumptions above lead to a situation in which "nearest neighbor" becomes meaningless. The assumptions used above are typical in empirical comparisons of high-dimensional NN query processing techniques; they are quite general, in that they allow any kind of data and query distribution (e.g., normal, Zipf, uniform) to be used, as long as they are IID. Our results therefore call into question the value of such empirical comparisons of NN query processing techniques.

Note that the assumptions of this section are by no means necessary for Theorem 1 to be applicable; we present empirical results for a "mix" data distribution in Section 5 that illustrates this point.

3.1.2 Identical Dimensions with no Independence

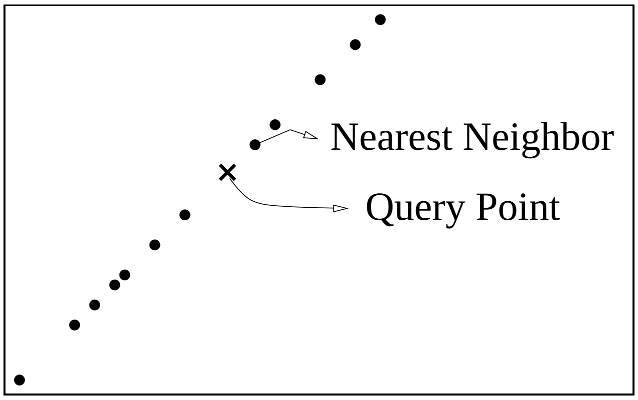

Figure 4: Identical but Dependent Dimensions

In contrast to the previous case, consider the situation where all dimensions of both the query point and the data points follow identical distributions, but are completely dependent (i.e., value for dimension 1 = value for dimension 2 = ...). Conceptually, the result is a set of data points on a diagonal line, and a query point somewhere on that same line (See Figure 4). Observe that no matter how many dimensions are added, the underlying query can actually be converted to a one-dimensional nearest neighbor problem.

The condition in Equation 5 no longer converges to zero under these assumptions. This is easy to see when one considers that while the divisor is unaffected by the dependence we have introduced (since expected values are unaffected by dependence), the quotient is no longer $m\sigma^2$. Instead, since all dimensions have the same value, the sum is performed inside the variance, yielding a quotient of:

$$\mathrm{var}\left(m\left|Q_i - W_i\right|^p\right)\right) = m^2 \cdot \mathrm{var}\left(\left|Q_i - W_i\right|^p\right) = m^2\sigma^2$$

Plugging this into the original condition leads to convergence of the limit to the non-zero value σ^2/μ^2. Thus the scenario doesn't meet our condition.

4 Meaningful Applications of High Dimensional Indexing

In considering potentially realistic scenarios in which solving the high dimensional nearest neighbor problem is meaningful, we observed that exact match and approximate match queries can be reasonable. For instance, if there is dependence between the query point and the data points such that there exists some data point which matches the query point exactly, then $\mathrm{DMIN}_m = 0$. Thus, assuming that most of the data points aren't duplicates, a meaningful answer can be determined. Furthermore, if the problem statement is relaxed to require that the query point be within some small distance δ of a data point (instead of being required to be identical to a data point), we can still call the query meaningful. Note, however, that staying within some δ becomes more and more difficult as m increases since we are adding terms to the sum in the distance metric. For this version of the problem to be meaningful, therefore, as dimensionality increases the query point must be increasingly closer to some data point.

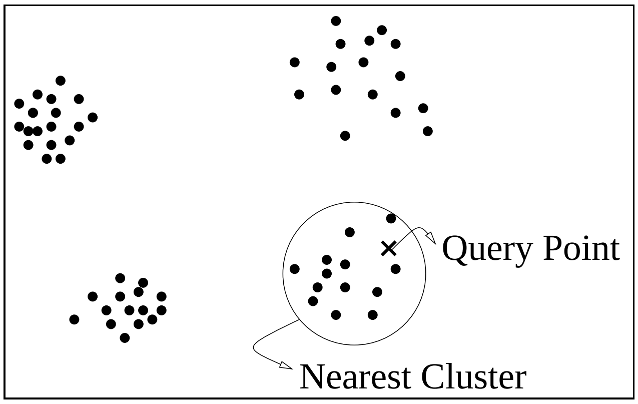

Figure 5: Nearest neighbor query in clustered data.

We can generalize the situation further as follows: The data consists of a set of randomly chosen points together with additional points distributed in clusters of some radius δ around one or more of the original points, and the query is required to fall within one of the data clusters (see Figure 5). This situation is the perfectly realized classification problem, where data naturally falls into discrete classes or clusters in some potentially high dimensional feature space. Figure 6 depicts a typical distance distribution in such a scenario. Note that there is clearly a cluster (the one into which the query point falls) that is closer than the others, which are all, more or less, indistinguishable in distance. Indeed, the proper response to such a query is to return all points within the closest cluster, not just the nearest point (which quickly becomes meaningless compared to other points in the cluster as dimensionality increases).

Observe however, that if we don't guarantee that the query point falls within some cluster, then the cluster from which the nearest neighbor is chosen is subject to the same meaningfulness limitations as the choice of nearest neighbor in the original version of the problem; Theorem 1 then applies to the choice of the "nearest cluster". Figure 7 depicts this scenario. Note that the data distributions in both figures are identical. The only difference is the choice of query point distribution (which is uniform in this case).

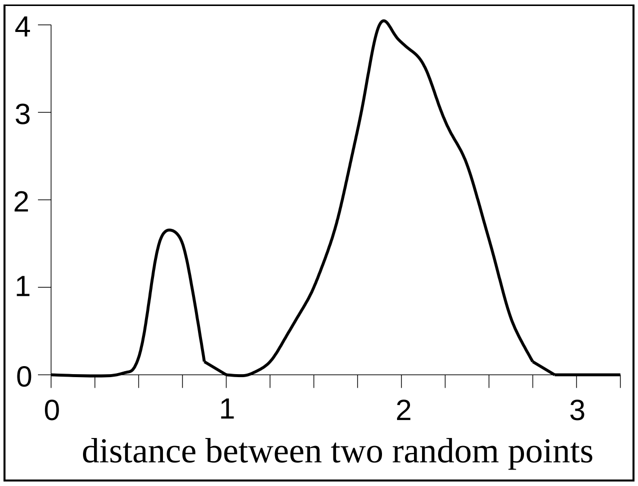

Figure 6: Probability density function of distance between random clustered data and query points.

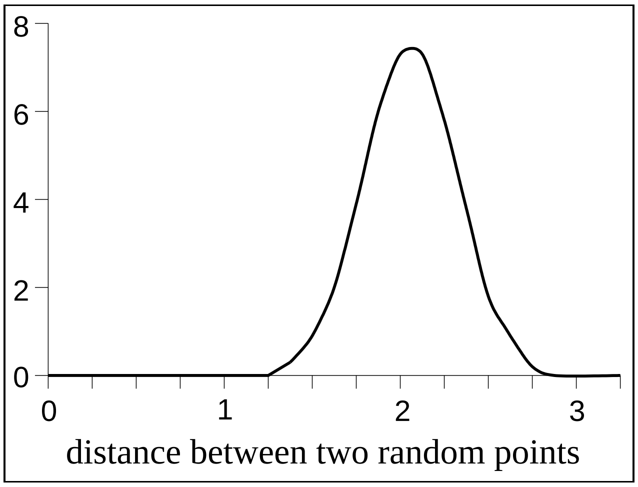

Figure 7: Probability density function of distance between random clustered data point and random uniform query point.

Another possible scenario where high dimensional nearest neighbor queries are meaningful occurs when the underlying dimensionality of the data is much lower than the actual dimensionality. There has been recent work on identifying (e.g. [8]) these situations and determining the useful dimensions in [11], which uses principal component analysis to identify meaningful dimensions. Note that these techniques are only useful if NN in the underlying dimensionality is meaningful.

5 Nearest Neighbor Simulations

This section examines the precise effect of dimensionality on the quality of answers to nearest neighbor queries by performing nearest neighbor query simulations. The variables considered were data distribution, dimensionality, dataset size, and ε. The data distributions included $uniform[0, \sqrt{12}]$, $N(0, 1)$, and $Exp(1)$. We chose one of these distributions for all dimensions, except in one experiment in which we assigned distributions to dimensions in round-robin fashion. All parameters of the distributions were chosen so that the variance of the distributions was 1. Dimensionality varied between 1 and 100. The dataset sizes included 50 thousand, 100 thousand , 1 million, and 10 million tuples(4 Gbytes). ε varied between 0 and 10. For a particular setting of the above variables, $\mathrm{DMAX}_m/\mathrm{DMIN}_m$ (the ratio of distance of furthest point to distance of nearest point) and $J_{m,\varepsilon}$ (the number of points that lie within ε distance of the query point) were measured.

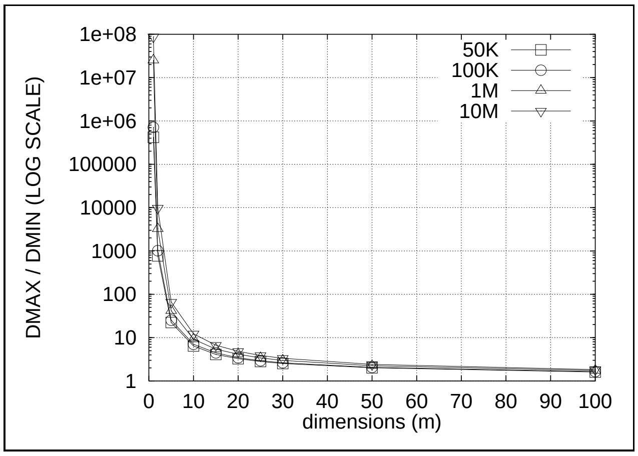

Figure 8: Size varies—uniform distribution.

Figure 8 shows $\mathrm{DMAX}_m/\mathrm{DMIN}_m$ as dimensionality increases, with every dimension following a uniform distribution. The information was collected for all dataset sizes. First, note that at dimension 1, the ratio is on the order of 10^8, providing plenty of contrast between the nearest object and the farthest object. At 10 dimensions, this contrast is already reduced by 7 orders of magnitude! Clearly things are looking bad by 20 dimensions, and abysmal at 30. Also of interest is that dataset size, which was varied widely, had little effect on the overall trend. All subsequent graphs will be for dataset sizes of one million tuples.

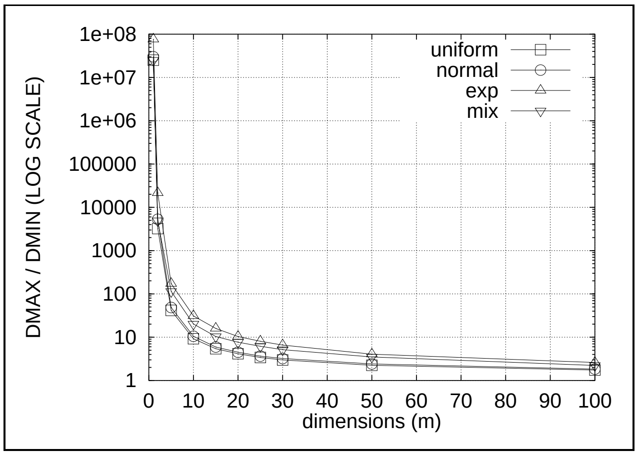

Figure 9: Distribution varies—one million tuples.

Figure 9, which is very similar to Figure 8, shows the effect of varying data distribution. Note that while all methods degraded at the same rate, exponential had inherently higher contrast than normal and uniform. Of course, the contrast is still reduced so drastically as dimensionality decreases that the absolute difference quickly becomes negligible. Also note that the "mix" workload, which mixed different distributions in round robin fashion, scored somewhere in the middle.

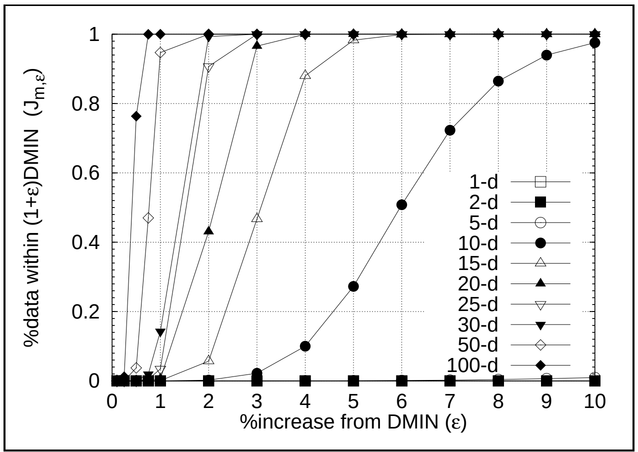

Figure 10: Epsilon varies—uniform distribution with one million tuples.

Figure 10 shows an entirely different kind of graph. This graph shows the percentage of data contained in the expanded query region as ε increases. First, note that queries over one, two, and five dimensional data retrieve insignificant amounts of the dataset at $\varepsilon < 10$. The percentage of data retrieved rises very quickly, however, as dimensionality is increased. Note that these graphs allow one to predict, for a particular distribution, the answer set size of a sphere query.

These graphs clearly demonstrate that our geometric intuition for nearest neighbor, which is based on one, two and three dimensions, fails us at an alarming rate as dimensionality increases. The distinction between nearest and farthest points, even at ten dimensions, is a tiny fraction of what it is in one, two, or three dimensions.

6 Analyzing the Performance of a NN Processing Technique

In this section, we discuss the ramifications of our results when evaluating techniques to solve the NN problem; in particular, many high-dimensional indexing techniques have been motivated by the NN problem. An important point that we make is that all future performance evaluations of high dimensional NN queries must include a comparison to linear scans. Finally, we explain how some of our results may be used to predict performance of individual queries.

First, our results indicate that while there exist situations in which high dimensional nearest neighbor queries are meaningful, they are very specific in nature and are quite different from the "independent dimensions" basis that most studies in the literature (e.g., [20, 10, 6, 4, 5]) use to evaluate techniques in a controlled manner. In the future, these NN technique evaluations should focus on those situations in which the results are meaningful. For instance, answers are meaningful when the data consists of small, well-formed clusters, and the query is guaranteed to land in or very near one of these clusters.

In terms of comparisons between NN techniques, we haven't found a single paper that compares their performance to that of a linear scan. Given our results, which suggest that most of the data must be examined as dimensionality increases, it is not surprising to note that at relatively few dimensions, linear scan handily beats these, in many cases, very complicated indexing structures. (Linear scan of a set of sequentially arranged pages is much faster than unordered retrieval of the same pages; so much so that secondary indexes are ignored by query optimizers unless the query is estimated to fetch less than 10% of the data pages. Fetching a large number of data pages through a multi-dimensional index usually results in unordered retrieval.)

For instance, the performance study of the parallel solution to the k-nearest neighbors problem presented in [4] indicates that their solution scales more poorly than a parallel scan of the data, and never beats a parallel scan in any of the presented data.

[20] provides us with information on the performance of both the SS tree and the R* tree in finding the 20 nearest neighbors. Conservatively assuming that linear scans cost 15% of a random examination of the data pages, linear scan outperforms both the SS tree and the R* tree at 10 dimensions in all cases. In [10], linear scan vastly outperforms the SR tree in all cases in this paper for the 16 dimensional synthetic dataset. For a 16 dimensional real dataset, the SR tree performs similarly to linear scan in a few experiments, but is usually beaten by linear scan. In [6], performance numbers are presented for NN queries where bounds are imposed on the radius used to find the NN. While the performance in high dimensionality looks good in some cases, in trying to duplicate their results we found that the radius was such that few, if any, queries returned an answer.

While performance of these structures in high dimensionality looks very poor, it is important to keep in mind that all the reported performance studies examined situations in which the difference in distance between the query point and the nearest neighbor differed little from the distance to other data points. Ideally, they should be evaluated for meaningful workloads. These workloads include low dimensional spaces and clustered data/queries as described in Section 4. Some of the existing structures may, in fact, work well in appropriate situations.

7 A New Variant of the Nearest Neighbor Problem

Our results show that the nearest neighbor might well be indistinguishable from other data points, with respect to distance from the query point. For a broad class of distributions, this phenomenon sets in fairly quickly with increasing dimensionality. For certain special situations, as discussed in Section 4, the nearest neighbor may be well-separated from other data points for relatively high dimensions, but only if some important conditions are satisfied by the data and query distributions.

These observations suggest that it is important for the user to be able to recognize when the nearest neighbor identified by the DBMS is indeed well-separated from the data. The standard formulations of the nearest neighbor and k-nearest neighbors problems, unfortunately, do not enable the user to estimate by how much the nearest neighbor is closer than other data points, or to estimate what fraction of the data is appreciably farther.

We therefore propose a new variant of the problem, designed to provide the user with such an estimate.

Definition 2 *Given a set of m-dimensional data points, an m-dimensional query point, and a fraction ε, the **ε-Radius Nearest Neighbors** problem is to find all points that are within a hypersphere of radius $DMIN_m(1+\varepsilon)$ of the query point, together with their distances from the query point.*

By specifying ε, the user can control the fraction of the distance to the nearest neighbor that is considered significant; intuitively, the smaller the set of returned points is as a fraction of the total dataset size, the more meaningful the nearest neighbor is.

Graphs like Figure 10 in Section 5 provide a nice estimate of the answer size for such a query. Given a dataset, we can randomly choose a collection of query points, and construct such a graph (based on averages over the chosen query points) using just two passes over the dataset. Note that the answer size estimate can be used in two ways. First, it can be used to determine if the nearest neighbor is likely to be meaningless. Second, if the nearest neighbor is likely to be well-separated from most of the other data points, the answer size estimate can be used to choose between a linear scan or some indexing structure.

8 Conclusions

In this paper, we studied the effect of dimensionality on NN queries. In particular, we identified a broad class of workloads for which the difference in distance between the nearest neighbor and other points in the dataset becomes negligible. This class of distributions includes distributions typically used to evaluate NN processing techniques.

In addition to providing intuition and examples of distributions in that class, we also discussed several situations in which NN queries do not break down in high dimensionality. In particular, the ideal datasets and work loads for classification/clustering algorithms seem reasonable in high dimensionality. It is interesting to note that if the scenario is deviated from (for instance, if the query point doesn't lie in a cluster), the queries become meaningless.

To find the dimensionality at which NN breaks down, we performed extensive simulations. The results indicated that the distinction in distance decreases fastest in the first 20 dimensions, quickly reaching a point where the difference in distance to query point between the nearest and farthest points drops below a factor of 3.

Finally, we discussed the manner in which performance should be measured when evaluating a NN processing technique. We observed that linear scans are never included among the empirical comparisons of various techniques, and that linear scans beat these techniques handily in high dimensionality. In addition, we noted that previous technique comparisons assumed scenarios in which the meaning of NN decreases with dimensionality, and that these techniques, may, in fact, work well for situations in which high dimensional NN is meaningful. Any future empirical studies should be based on meaningful scenarios.

Also discussed was an alternative to the k-nearest neighbor problem, where ε is specified instead of k. This is useful since different problem domains may require different ε . Our reformulated problem has the advantage that with a little precomputation on a per dataset basis, both performance estimation and meaningfulness checks may be done prior to the execution of individual queries.

References

[1] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In *Proc. 4th Inter. Conf. on FODO*, pages 69–84, 1993.

[2] S. F. Altschul, W. Gish, W. Miller, E. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[3] Y. H. Ang, Zhao Li, and S. H. Ong. Image retrieval based on multidimensional feature properties. In *SPIE vol. 2420*, pages 47–57, 1995.

[4] S. Berchtold, C. Böhm, B. Braunmüller, D. A. Keim, and H.-P. Kriegel. Fast parallel similarity search in multimedia databases. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 1–12, 1997.

[5] S. Berchtold, C. Böhm, D. A. Keim, and H.-P. Kriegel. A cost model for nearest neighbor search in high-dimensional data space. In *Proc. 16th ACM SIGACT-SIGMOD-SIGART Symposium on PODS*, pages 78–86, 1997.

[6] T. Bozkaya and M. Ozsoyoglu. Distance-based indexing for high-dimensional metric spaces. In *Proc. 16th ACM SIGACT-SIGMOD-SIGART Symposium on PODS*, pages 357–368, 1997.

[7] C. Faloutsos et al. Efficient and effective querying ny image content. *Journal of Intelligent Information Systems*, 3(3):231–262, 1994.

[8] C. Faloutsos and V. Gaede. Analysis of n-dimensional quadtrees using the Housdorff fractal dimension. In *Proc. ACM SIGMOD Int. Conf. of the Management of Data*, 1996.

[9] U. M. Fayyad and P. Smyth. Automated analysis and exploration of image databases: Results, progress and challenges. *Journal of intelligent information systems*, 4(1):7–25, 1995.

[10] N. Katayama and S. Satoh. The SR-tree: An index structure for high-dimensional nearest neighbor queries. In *Proc. 16th ACM SIGACT-SIGMOD-SIGART Symposium on PODS*, pages 369–380, 1997.

[11] K.-I. Lin, H. V. Jagadish, and C. Faloutsos. The TV-Tree: An index structure for high-dimensional data. *VLDB Journal*, 3(4):517–542, 1994.

[12] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. In *IEEE Trans. on Pattern Analysis and Machine Learning*, volume 18(8), pages 837–842, 1996.

[13] R. Mehrotra and J. E. Gary. Feature-based retrieval of similar shapes. In *9th Data Engineering Conference*, pages 108–115, 1992.

[14] H. Murase and S. K. Nayar. Visual learning and recognition of 3D objects from appearance. *Int. J. of Computer Vision*, 14(1):5–24, 1995.

[15] S. A. Nene and S. K. Nayar. A simple algorithm for nearest neighbor search in high dimensions. In *IEEE Trans. on Pattern Analysis and Machine Learning*, volume 18(8), pages 989–1003, 1996.

[16] A. Pentland, R. W. Picard, and S. Scalroff. Photobook: Tools for content based manipulation of image databases. In *SPIE Volume 2185*, pages 34–47, 1994.

[17] M. J. Swain and D. H. Ballard. Color indexing. *Inter. Journal of Computer Vision*, 7(1):11–32, 1991.

[18] D. L. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. In *IEEE Trans.*

on Pattern Analysis and Machine Learning, volume 18(8), pages 831–836, 1996.

[19] G. Taubin and D. B. Cooper. Recognition and positioning of rigid objects using algebraic moment invariants. In *SPIE Vol. 1570*, pages 318–327, 1991.

[20] D. A. White and R. Jain. Similarity indexing with the SS-Tree. In *ICDE*, pages 516–523, 1996.