Computer Sciences Department

When Is "Nearest Neighbor" Meaningful?

Kevin Beyer
Jonathan Goldstein
Raghu Ramakrishnan
Uri Shaft

Technical Report #1377

June 1998

UNIVERSITY OF
WISCONSIN
MADISON

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
The contact author is Kevin Beyer

Abstract

We explore the effect of dimensionality on the "nearest neighbor" problem. We show that under a broad set of conditions (much broader than independent and identically distributed dimensions), as dimensionality increases, the distance to the nearest data point approaches the distance to the farthest data point. To provide a practical perspective, we present empirical results on both real and synthetic data sets that demonstrate that this effect can occur for as few as 10-15 dimensions.

These results should not be interpreted to mean that high-dimensional indexing is never useful; we illustrate this point by identifying some high-dimensional workloads for which this effect does not occur. However, our results *do* emphasize that the methodology used almost universally in the database literature to evaluate high-dimensional indexing techniques is flawed, and should be modified. In particular, most such techniques proposed in the literature are not evaluated versus simple linear scan, and are evaluated over workloads for which nearest neighbor is not meaningful. Often, even the reported experiments, when analyzed carefully, show that linear scan would outperform the techniques being proposed on the workloads studied in high (10-15) dimensionality!

Our results also suggest that users must be able to assess how meaningful the answer to a high dimensional nearest neighbor query is. Since the "nearest neighbor" is in general at the same distance as the "farthest neighbor", before we attach significance to a nearest neighbor answer, we should have a measure of how many data points are significantly farther than the answer points. We propose an intuitive variant of the problem that provides such a measure.

1 Introduction

In recent years, many researchers have focused on 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 (e.g., see [22]) complex data—such as images (e.g. [13, 24, 25, 19, 25, 21, 23, 16, 3]), sequences (e.g. [2, 1]), video (e.g. [13]), and shapes (e.g. [13, 26, 23, 20])—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.

This paper makes the following contributions:

- We show that under certain broad 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, the *contrast* in distances to different data points becomes nonexistent. The conditions we have identified in which this happens are much broader than the independent and identically distributed (IID)

*This work was partially supported by a "David and Lucile Packard Foundation Fellowship in Science and Engineering", a "Presidential Young Investigator" award, NASA research grant NAGW-3921, ORD contract 144-ET33, and NSF grant 144-GN62.

dimensions assumption that other work assumes. Our result characterizes the problem itself, rather than specific algorithms that address the problem. In addition, our observations apply equally to the k-nearest neighbor variant of the problem. When one combines this result with the observation that most applications of high dimensional NN are heuristics for similarity in some domain (e.g. color histograms for image similarity), serious questions are raised as to the validity of many mappings of similarity problems to high dimensional NN problems.

- 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. In addition, we present results from a real image database that clearly indicate that these dimensionality effects occur in practice. Our observations suggest that high-dimensional feature vector representations for multimedia similarity search must be used with caution, in particular, checking that the workload yields a clear separation between nearest and farthest neighbors for typical queries (e.g., through sampling). To place these observations in perspective, we identify special workloads for which the concept of nearest neighbor continues to be meaningful in high dimensionality.

- Our results underscore the point that evaluation of a technique for nearest-neighbor search should be based on meaningful workloads. We observe that the database 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 almost always out-performs their techniques in high dimensionality on the examined datasets. This is unsurprising as the workloads used to evaluate these techniques are in the class of "badly behaving" workloads identified by our results; the proposed methods may well be effective for appropriately chosen workloads, but this is not examined in their performance evaluation.

- 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 the size of the answer set varies in our reformulation, we also provide a mechanism for estimating the size of the answer set. While this reformulation resembles the reformulation suggested in [6], their reformulation is used to obtain an approximate nearest neighbor algorithm; in contrast, our goal is to ensure that there is a meaningful answer set for a given query.

In summary, our results suggest that more care be taken when thinking of nearest neighbor approaches and high dimensional indexing algorithms; we supplement our theoretical results with experimental data and a careful discussion.

2 On the Significance of "Nearest Neighbor"

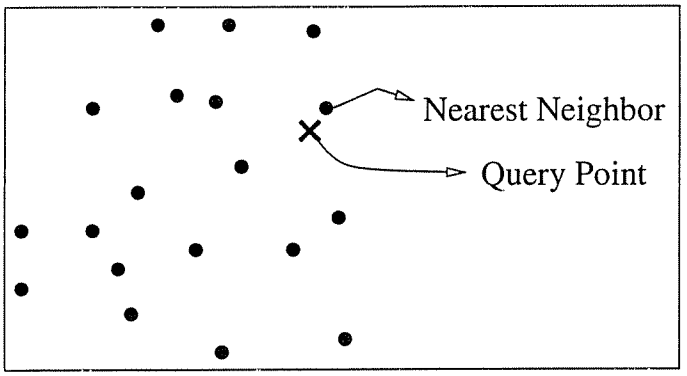

Figure 1: Query point and its nearest neighbor.

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?"

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 3. 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. Furthermore, consider the scenario where the position of each point is thought to lie in some circle with high confidence (see Figure 2). Such a situation can come about either from numerical error in calculating the location, or "heuristic error", which derives from the algorithm used to deduce the point (e.g. if a flat rather than a spherical map were used to determine distance). In this scenario, the determination of a nearest neighbor is impossible with any reasonable degree of confidence!

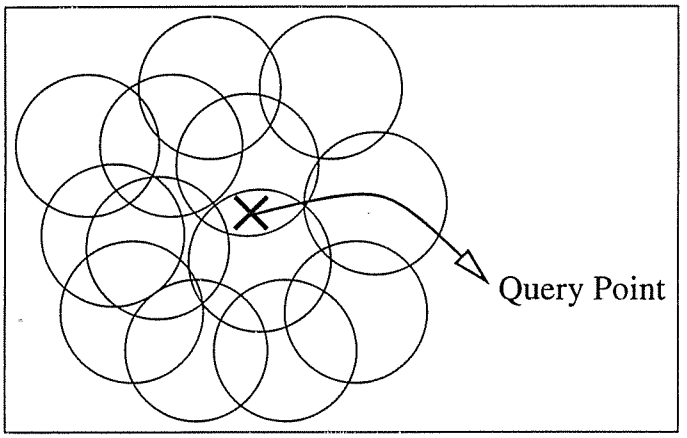

Figure 2: The data points are approximations. Each circle denotes a region where the true data point is supposed to be.

While the scenario depicted in Figure 3 is very contrived for a geographical database (and for any practical two dimensional application of NN), 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 4). 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 use the notion *instability* for describing this phenomenon.

Definition 1 *A nearest neighbor query is* **unstable** *for a given ε if the distance from the query point to most data points is less than $(1+\varepsilon)$ times the distance from the query point to its nearest neighbor.*

We show that in many situations, for **any** fixed $\varepsilon > 0$, as dimensionality rises, the probability that a query is unstable converges to 1. Note that the points that fall in the enlarged query region are the valid answers to the approximate nearest neighbors problem (described in [6]).

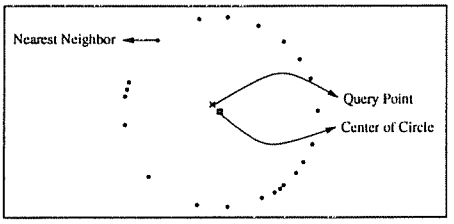

Figure 3: Another query point and its nearest neighbor.

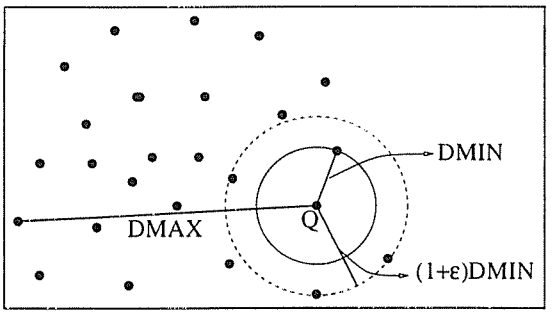

Figure 4: Illustration of query region and enlarged region. (DMIN is the distance to the nearest neighbor, and DMAX to the farthest data point.)

3 NN in High-Dimensional Spaces

This section contains our formulation of the problem, our formal analysis of the effect of dimensionality on the meaning of the result, and some formal implications of the result that enhance understanding of our primary result.

3.1 Notational Conventions

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

A vector: $\vec{x}$

Probability of an event e**:** $P[\mathrm{e}]$.

Expectation of a random variable X**:** $\mathbf{E}[X]$.

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

IID: Independent and identically distributed. (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.

3.2 Some Results from Probability Theory

Definition 2 *A sequence of random vectors (all vectors have the same arity)* $\vec{A_1}, \vec{A_2}, \ldots$ **converges in probability** *to a constant vector* $\vec{c}$ *if for all* $\varepsilon > 0$ *the probability of* $\vec{A_m}$ *being at most* ε *away from* $\vec{c}$ *converges to 1 as* $m \to \infty$*. In other words:*

$$\forall \varepsilon > 0\,, \quad \lim_{m \to \infty} P\left[\left\|\vec{A_m} - \vec{c}\right\| \leq \varepsilon\right] = 1$$

We denote this property by $\vec{A_m} \to_p \vec{c}$*. We also treat random variables that are not vectors as vectors with arity 1.*

Lemma 1 *If* $B_1, B_2, \ldots$ *is a sequence of random variables with finite variance and* $\lim_{m\to\infty} \mathbf{E}[B_m] = b$ *and* $\lim_{m\to\infty} var(B_m)$ 0 *then* $B_m \to_p b$*.*

A version of Slutsky's theorem *Let* $\vec{A_1}, \vec{A_2}, \ldots$ *be random variables (or vectors) and* g *be a continuous function. If* $\vec{A_m} \to_p \vec{c}$ *and* $g(\vec{c})$ *is finite then* $g(\vec{A_m}) \to_p g(\vec{c})$*.*

Corollary 1 (to Slutsky's theorem) *If* $X_1, X2, \ldots$ *and* $Y_1, Y2, \ldots$ *are sequences or random variables s.t.* $X_m \to_p a$ *and* $y_m \to_p b \neq 0$ *then* $X_m/Y_m \to_p a/b$*.*

Although we do not make use of it, we include the Weak Law of Large Numbers below for completeness, since we refer to it in discussions.

The Weak Law of Large Numbers: *Let $A_1, A_2, \ldots$ be IID random variables and $\mathbf{E}[A_i] = \mu$ is finite then*

$$\frac{1}{m}\sum_{i=1}^{m} A_i \rightarrow_p \mu$$

3.3 Nearest Neighbor Formulation

Given a dataset and a query point, we want to analyze how much the distance to the nearest neighbor differs from the distance to other data points. We do this by evaluating the number of points that are no farther away than a factor ε larger than the distance between the query point and the NN, as illustrated in Figure 4. When examining this characteristic, we assume nothing about the structure of the distance calculation.

We will study this characteristic by examining the distribution of the distance between query points and data points as some variable m changes. Note that eventually, we will interpret m as dimensionality. However, nowhere in the following proof do we rely on that interpretation. One can view the proof as a convergence condition on a series of distributions (which we happen to call distance distributions) that provides us with a tool to talk formally about the "dimensionality curse".

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

Definition 3 *:*

m is the variable that our distance distributions may converge under (m ranges over all positive integers).

$F_{data_1}, F_{data_2}, \ldots$ is a sequence of data distributions.

$F_{query_1}, F_{query_2}, \ldots$ is a sequence of query distributions.

n is the (fixed) number of samples (data points) from each distribution.

$\forall m \quad P_{m,1}, \ldots, P_{m,n}$ are n independent data points per m such that $P_{m,i} \sim F_{data_m}$.

$Q_m \sim F_{query_m}$ is a query point chosen independently from all $P_{m,i}$.

$0 < p < \infty$ is a constant.

$\forall m, d_m$ is a function that takes a data point from the domain of F_{data_m} and a query point from the domain of F_{query_m} and returns a non-negative real number as a result.

$DMIN_m = \min\{d_m(P_{m,i}, Q_m) \,|\, 1 \leq i \leq n\}$.

$DMAX_m = \max\{d_m(P_{m,i}, Q_m) \,|\, 1 \leq i \leq n\}$.

3.4 Instability Result

Our main theoretical tool is presented below. In essence, it states that assuming the distance distribution behaves a certain way as m increases, the difference in distance between the query point and all data points becomes negligible (i.e., the query becomes unstable). Future sections will show that the necessary behavior described in this section identifies a large class (larger than any other classes we are aware of for which the distance result is either known or can be readily inferred from known results) of workloads. More formally, we show:

Theorem 1 *Under the conditions in Definition 3, if*

$$\lim_{m\to\infty} var\left(\frac{(d_m(P_{m,1}, Q_m))^p}{\mathbf{E}[(d_m(P_{m,1}, Q_m))^p]}\right) = 0 \tag{1}$$

Then for every $\varepsilon > 0$

$$\lim_{m\to\infty} P[DMAX_m \leq (1+\varepsilon)DMIN_m] = 1$$

Proof Let $\mu_m = \mathbf{E}[(d_m(P_{m,i}, Q_m))^p]$. (Note that the value of this expectation is independent of i since all $P_{m,i}$ have the same distribution.)

Let $V_m = (d_m(P_{m,1}, Q_m))^p / \mu_m$.

Part 1: We'll show that $V_m \rightarrow_p 1$.

It follows that $\mathbf{E}[V_m] = 1$ (because V_m is a random variable divided by its expectation.)

Trivially, $\lim_{m\to\infty} \mathbf{E}[V_m] = 1$.

The condition of the theorem (Equation 1) means that $\lim_{m\to\infty} \text{var}(V_m) = 0$. This, combined with $\lim_{m\to\infty} \mathbf{E}[V_m] = 1$, enables us to use Lemma 1 to conclude that $V_m \rightarrow_p 1$.

Part 2: We'll show that if $V_m \rightarrow_p 1$ then $\lim_{m\to\infty} P[\text{DMAX}_m \leq (1+\varepsilon)\text{DMIN}_m] = 1$.

Let $\vec{X}_m = (d_m(P_{m,1}, Q_m)/\mu_m, \ldots, d_m(P_{m,n}, Q_m)\mu_m)$ (a vector of arity n).

Since each part of the vector $\vec{X}_m$ has the same distribution as V_m, it follows that $\vec{X}_m \rightarrow_p (1, \ldots, 1)$.

Since min and max are continuous functions we can conclude from Slutsky's theorem that $\min(\vec{X}_m) \rightarrow_p \min(1, \ldots, 1) = 1$, and similarly, $\max(\vec{X}_m) \rightarrow_p 1$.

Using Corollary 1 on $\max(\vec{X}_m)$ and $\min(\vec{X}_m)$ we get

$$\frac{\max(\vec{X}_m)}{\min(\vec{X}_m)} \rightarrow_p \frac{1}{1} = 1$$

Note that $\text{DMIN}_m = \mu_m \min(\vec{X}_m)$ and $\text{DMAX}_m = \mu_m \max(\vec{X}_m)$. So,

$$\frac{\text{DMAX}_m}{\text{DMIN}_m} = \frac{\mu_m \max(\vec{X}_m)}{\mu_m \min(\vec{X}_m)} = \frac{\max(\vec{X}_m)}{\min(\vec{X}_m)}$$

Therefore

$$\frac{\text{DMAX}_m}{\text{DMIN}_m} \rightarrow_p 1$$

By definition of convergence in probability we have that for all $\varepsilon > 0$,

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

Also,

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

($P[\text{DMAX}_m \geq \text{DMIN}_m] = 1$ so the absolute value in the last term has no effect.) Thus,

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

■

In summary, the above theorem says that if the precondition holds (i.e., if the distance distribution behaves a certain way as m increases), all points converge to the same distance from the query point. Thus, under these conditions, the concept of nearest neighbor is no longer useful.

We may be able to use this result by directly showing that $V_m \rightarrow_p 1$ and using part 2 of the proof. (For example, for IID distributions, $V_m \rightarrow_p 1$ follows readily from the Weak Law of Large Numbers.) Later sections demonstrate that our result provides us with a handy tool for discussing scenarios resistant to analysis using law of large numbers arguments. From a more practical standpoint, there are two issues that must be addressed to determine the theorem's impact:

- How restrictive is the condition

$$\lim_{m\to\infty} \text{var}\left(\frac{(d_m(P_{m,1}, Q_m))^p}{\mathbf{E}\left[(d_m(P_{m,1}, Q_m))^p\right]}\right) = \lim_{m\to\infty} \frac{\text{var}\left((d_m(P_{m,1}, Q_m))^p\right)}{\left(\mathbf{E}\left[(d_m(P_{m,1}, Q_m))^p\right]\right)^2} = 0 \tag{2}$$

  which is necessary for our results to hold? In other words , it says that as we increase m and examine the resulting distribution of distances between queries and data, the variance of the distance distribution scaled by the overall magnitude of the distance converges to 0. To provide a better understanding of the restrictiveness of this condition, Sections 3.5 and 4 discuss scenarios that do and do not satisfy it.

- 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? This issue is more difficult to tackle analytically. 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 5.1.

3.5 Application of Our Theoretical Result

This section analyses the applicability of Theorem 1 in formally defined situations. This is done by determining, for each scenario, whether the condition in Equation 2 is satisfied.

All of these scenarios define a workload and use an L_p distance metric over multidimensional query and data points with dimensionality m. (This makes the data and query points vectors with arity m.) It is important to notice that this is the first section to assign a particular meaning to d_m (as an L_p distance metric), p (as the parameter to L_p), and m (as dimensionality). Theorem 1 did not make use of these particular meanings.

We explore some scenarios that satisfy Equation 2 and some that do not. We start with basic IID assumptions and then relax the these assumptions in various ways. We start with two "sanity checks": we show that distances converge with IID dimensions (Section 3.5.1), and we show that Equation 2 is not satisfied when the data and queries fall on a line (Section 3.5.2). We then discuss examples involving correlated attributes and differing variance between dimensions, to illustrate scenarios where the Weak Law of Large Numbers cannot be applied (Sections 3.5.3, 3.5.4, and 3.5.5).

3.5.1 IID Dimensions with Query and Data Independence

Assume the following:

- The data distribution and query distribution are IID in all dimensions.
- All the appropriate moments are finite (i.e., up to the $\lceil 2p \rceil$'th moment).
- The query point is chosen independently of the data points.

We will prove that the conditions of Theorem 1 are satisfied under these assumptions. We take a random data point $\vec{P}_m = (P_1, \ldots, P_m)$ and a random query point $\vec{Q}_m = (Q_1, \ldots, Q_m)$. We have that $(d_m(\vec{P}_m, \vec{Q}_m))^p = \sum_{i=1}^m |P_i - Q_i|^p$. To evaluate whether

$$\lim_{m\to\infty} \frac{\text{var}\left((d_m(\vec{P}_m, \vec{Q}_m))^p\right)}{\left(\mathbf{E}\left[(d_m(\vec{P}_m, \vec{Q}_m))^p\right]\right)^2} = 0$$

is satisfied (Equation 2 in vector notation), we observe that under the IID assumptions $|P_i - Q_i|^p$ are IID variables with some expectation μ and some variance σ^2 that are the same regardless of i and m. Therefore, $\mathbf{E}\left[(d_m(\vec{P}_m, \vec{Q}_m))^p\right] = \mathbf{E}\left[\sum_{i=1}^m |P_i - Q_i|^p\right] = m\mu$ and $\text{var}\left((d_m(\vec{P}_m, \vec{Q}_m))^p\right) = \text{var}\left(\sum_{i=1}^m |P_i - Q_i|^p\right) = m\sigma^2$.

Substituting these values back into our original limit yields convergence to zero, and thus meets our condition. Therefore, the assumptions above lead to a situation in which "nearest neighbor" is not useful. The assumptions used

above are typical in empirical comparisons of high-dimensional NN query processing techniques; they are somewhat 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 in high dimensionality.

Another way to prove that $V_m \rightarrow_p 1$ would be to apply the Law of Large Numbers, and use only Part 2 of Theorem 1's proof. While this result is not original, it is a nice "sanity check."

The assumptions of this example are by no means necessary for Theorem 1 to be applicable. Throughout this section, there are examples of workloads which cannot be discussed using the Weak Law of Large Numbers. While there are innumerable slightly stronger versions of the Weak Law of Large Numbers, Section 3.5.5 contains an example which meets our condition, and for which the Weak Law of Large Numbers is inapplicable.

3.5.2 Identical Dimensions with no Independence

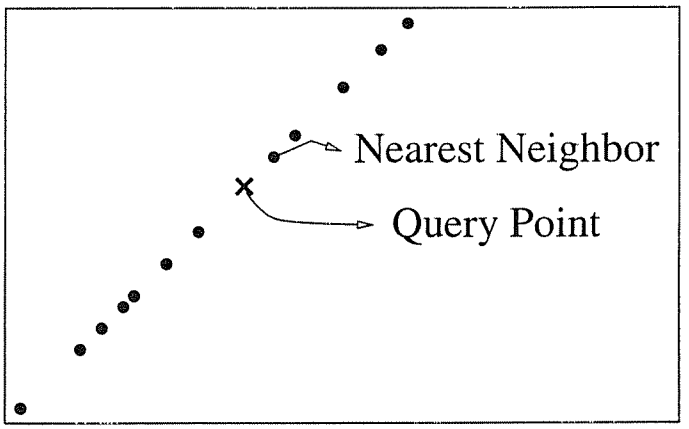

Figure 5: Identical but Dependent Dimensions

We use the same notation as in the previous example. 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 and a query point on a diagonal line (See Figure 5). 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 2 no longer converges to zero under these assumptions. This is easy to see when one considers that while the denominator is unaffected by the dependence we have introduced (since expected values are unaffected by dependence), the numerator is no longer $m\sigma^2$. Instead, since all dimensions have the same value, the sum is performed inside the variance, yielding a numerator of:

$$\text{var}\left(m\,|P_i - Q_i|^p\right)) = m^2 \cdot \text{var}\left(|P_i - Q_i|^p\right) = m^2\sigma^2$$

Substituting 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.

3.5.3 Unique Dimensions with Correlation Between All Dimensions

In this example, we intentionally break many assumptions underlying the IID case. Not only is every dimension unique, but all dimensions are correlated with all other dimensions and the variance of added dimension increases. The following is a description of the problem and the outline of our analysis showing that this workload behaves badly.

We generate an m dimensional data point (or query point) $\vec{X}_m = (X_1, \ldots, X_m)$ as follows:

- First we take independent random variables $U_1, \ldots, U_m$ such that $U_i \sim \text{Uniform}(0, \sqrt{i})$.
- We define $X_1 = U_1$.
- For all $2 \leq i \leq m$ define $X_i = U_i + (X_{i-1}/2)$.

Clearly there is correlation between every pair of dimensions. Let $\vec{P}_m$ and $\vec{Q}_m$ be independent points generated in this way and suppose we use the L_2 metric. The distance squared is $\sum_{i=1}^{m} |P_i - Q_i|^2$. However, the components $|P_i - Q_i|^2$ are all different. Furthermore, the variance of the components increases with i and converges to infinity (i.e., each added dimensions contributes more to the distance between points than the previous dimension).

The expectation of the distance squared is

$$\mathbf{E}\left[\left\|\vec{P}_m - \vec{Q}_m\right\|^2\right] = \frac{1}{9}m^2 + \frac{1}{3}m + o(m)$$

Where $o(m)$ is something much smaller than m. The variance of the distance squared is

$$\text{var}\left(\left\|\vec{P}_m - \vec{Q}_m\right\|^2\right) = \frac{4}{3645}m^3 + \frac{58}{18225}m^2 + \frac{8713}{72900}m + o(m)$$

Substituting these values back into our original limit produces convergence to zero, and thus meets our condition. We explore this example further in the simulation study (Section 5.1).

3.5.4 Variance Converging to 0

This example illustrates that there are workloads that meet the preconditions of Theorem 1, even though the variance of the distance in each added dimension converges to 0. One would expect that only some finite number of the earlier dimensions would dominate the distance. Again, this is not the case.

Suppose we choose a point $\vec{X}_m = (X_1, \ldots, X_m)$ such that the X_i's are independent and $X_i \sim N(0, 1/i)$. Let $\vec{P}_m$ and $\vec{Q}_m$ be such independent points and suppose we use the L_2 metric. Then $\mathbf{E}\left[|P_i - Q_i|^2\right] = 2/i$. Also, $\text{var}\left(|P_i - Q_i^2\right) = 12/i^2$. Using the independence of dimensions we get

$$\mathbf{E}\left[\left\|\vec{P}_m - \vec{Q}_m\right\|^2\right] = \sum_{i=1}^{m} \frac{2}{i} \quad \text{and} \quad \text{var}\left(\left\|\vec{P}_m - \vec{Q}_m\right\|^2\right) = \sum_{i=1}^{m} 12/i^2$$

Since

$$\lim_{m\to\infty} \sum_{i=1}^{m} \frac{2}{i} = \infty \quad \text{and} \quad \lim_{m\to\infty} \sum_{i=1}^{m} 12/i^2 = 2\pi^2$$

we have

$$\lim_{m\to\infty} \frac{\text{var}\left(\left\|\vec{P}_m - \vec{Q}_m\right\|^2\right)}{\left(\mathbf{E}\left[\left\|\vec{P}_m - \vec{Q}_m\right\|^2\right]\right)^2} = \lim_{m\to\infty} \frac{\sum_{i=1}^{m} 12/i^2}{\left(\sum_{i=1}^{m} \frac{2}{i}\right)^2} = 0$$

and the conditions of the theorem are met.

3.5.5 Marginal Data and Query Distributions Change with Dimensionality

In this example, the marginal distributions of data and queries change with dimensionality. Thus, the distance distribution as dimensionality increases cannot be described as the distance in a lower dimensionality plus some new component from the new dimension. As a result, the weak law of large numbers, which implicitly is about sums of increasing size, cannot provide insight into the behavior of this scenario. The distance distributions must be treated,

as our technique suggests, as a series of random variables whose variance and expectation can be calculated and examined in terms of dimensionality.

Let the m dimensional data space S_m be the boundary of an m dimensional unit hyper-cube. (i.e., $S_m = [0,1]^m - (0,1)^m$). In addition, let the distribution of data points be uniform over S_m. In other words, every point in S_m has equal probability of being sampled as a data point. Lastly, the distribution of query points is identical to the distribution of data points.

If $\vec{X}_m = (X_1, \ldots, X_m)$ is a random data point, then for $i \neq j$ the variables X_i and X_j are not independent although their covariance is 0. More precisely, $P[X_i = 0 \text{ and } X_j = 0] = 0$ but $P[X_i = 0] \cdot P[X_j = 0] = 1/(4m^2)$. If X_i and X_j were independent, both expressions would have the same value.

Let $\vec{P}_m$ and $\vec{Q}_m$ be two independent m dimensional points and suppose we use the L_2 metric. The expectation of the square distance is

$$\mathbf{E}\left[\left\|\vec{P}_m - \vec{Q}_m\right\|^2\right] = \frac{1}{6}m + \frac{1}{3}$$

The variance of the squared distance is

$$\text{var}\left(\left\|\vec{P}_m - \vec{Q}_m\right\|^2\right) = \frac{7}{180}m + \frac{1}{10} + \frac{1}{9m}$$

Substituting these values back into our original limit produces convergence to zero, and thus meets our condition.

In essence, this example shows that even though the marginal distributions change when we increase dimensionality, the variance and expected value of the distance behave in a quantifiable way. This allows us to prove that the preconditions of Theorem 1 are satisfied.

4 Meaningful Applications of High Dimensional Indexing

In this section, we place Theorem 1 in perspective and observe that it should *not* be interpreted to mean that high-dimensional indexing is never useful. We do this by identifying scenarios that arise in practice and that are likely to have good separation between nearest and farthest neighbors.

4.1 Classification and Approximate Matching

To begin with, 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 $\text{DMIN}_m = 0$. 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 remain meaningful as dimensionality increases, the query point must be increasingly closer to some data point.

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 6). 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 7 depicts a typical distance distribution in such a scenario. There is 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

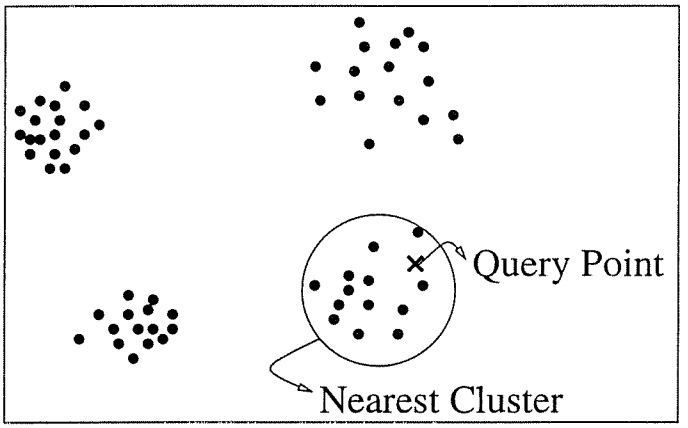

Figure 6: Nearest neighbor query in clustered data.

8 depicts this scenario. The data distributions in both figures are identical. The only difference is the choice of query point distribution (which is uniform in this case).

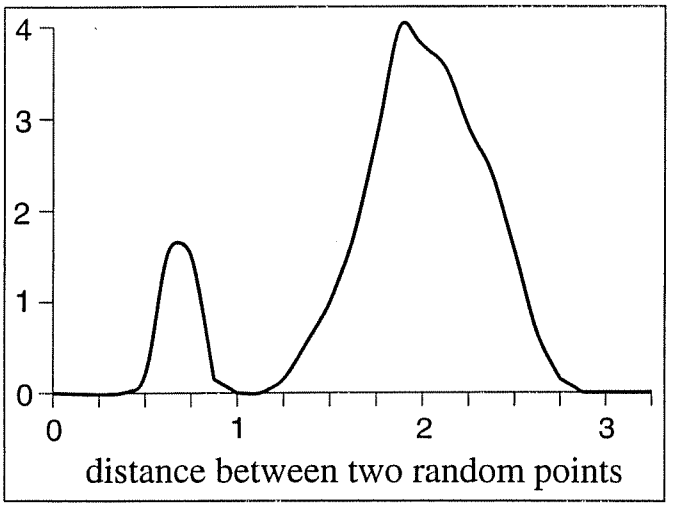

Figure 7: Probability density function of distance between random clustered data and query points.

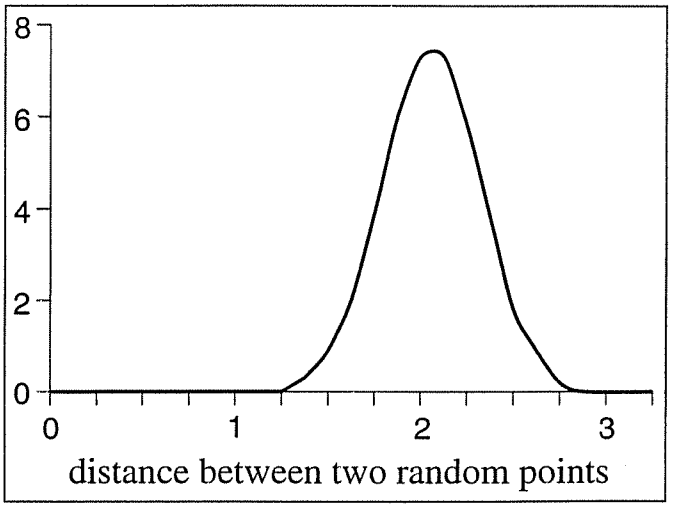

Figure 8: Probability density function of distance between random clustered data point and random uniform query point.

4.2 Implicitly Low Dimensionality

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 these situations (e.g. [15, 7, 14]) and determining the useful dimensions (e.g. [18], which uses principal component analysis to identify meaningful dimensions). Of course, these techniques are only useful if NN in the underlying dimensionality is meaningful.

5 Experimental Studies of NN

Theorem 1 only holds in the limit as dimensionality increases. In practice, at what dimensionality do the problems that we anticipate become significant? We address this issue through empirical studies. First, in Section 5.1, we

generate a range of synthetic datasets that satisfy the preconditions of Theorem 1, and show that the problems are significant as early as 15 dimensions. Next, in Section 5.2, we examine two real datasets from an image database application, and show that the problems manifest themselves in practice.

5.1 Nearest Neighbor Simulations

This section examines the 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, $\text{DMAX}_m/\text{DMIN}_m$ (the ratio of distance of furthest point to distance of nearest point) and the number of points that lie within ε distance of the query point were measured.

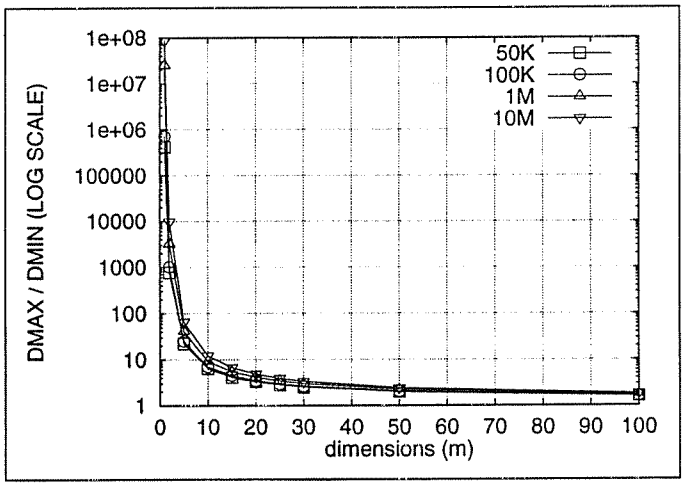

Figure [illegible] ries, uniform distribution.

Figure 9 shows $\text{DMAX}_m/\text{DMIN}_m$ as dimens[illegible] increases, with every dimension following a uniform distribution. The information was collected for all dataset[illegible]mension 1, the ratio is on the order of 10^8, providing plenty of contrast between the nearest object and t[illegible]nest 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 10, which is very similar to Figure 9, shows the effect of varying data distribution. The "mix" distribution was a (round-robin) combination of the other three, and therefore the dimensions were independent, but not identical. While all methods degraded roughly at the same rate, exponential had inherently higher contrast than normal and uniform, and mix scored in the middle. However, as dimensionality decreases, the contrast is reduced so drastically that the absolute difference quickly becomes negligible.

Figure 11 shows an entirely different kind of graph. This graph shows the percentage of data contained in the expanded query region as ε increases. Queries over one, two, and five dimensional data retrieve insignificant amounts of the dataset when $\varepsilon < 10$. The percentage of data retrieved rises quickly, however, as dimensionality is increased. This graph allows one to predict, for a particular distribution, the answer set size of a sphere query.

We ran experiments with two different correlated workloads (see Figure 12). The "uniform" line represents an IID uniform(0,1) workload that is presented as a baseline. The workload for the "recursive" line has correlation between every pair of dimensions and every new dimension had a larger variance. This is the workload described in

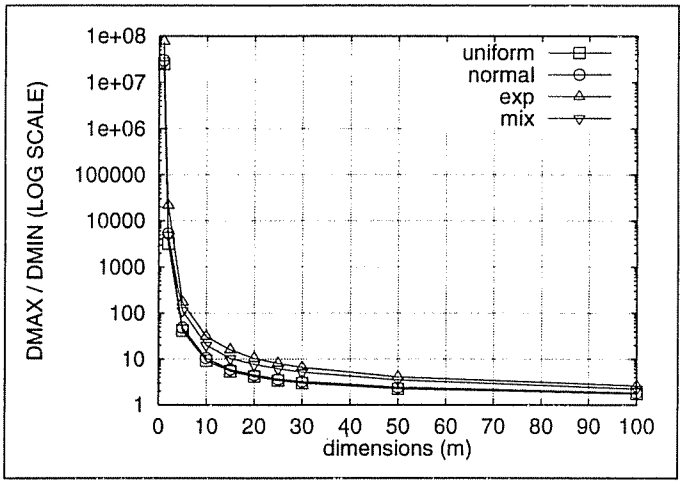

Figure 10: Distribution varies, one million tuples.

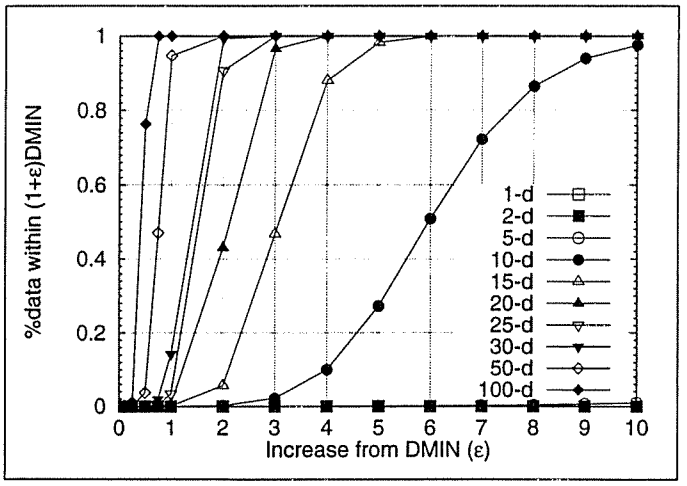

Figure 11: Epsilon varies, uniform distribution, one million tuples.

Section 3.5.3. The "two degrees of freedom" workload generates query and data points on a two dimensional plane, and was generated as follows:

- Let $a_1, a_2, \ldots$ and $b_1, b_2, \ldots$ be constants in (-1,1).
- Let U_1, U_2 be independent uniform(0,1).
- For all $1 \leq i \leq m$ let $X_i = a_i U_1 + b_i U_2$.

This last workload does not satisfy Equation 2. Figure 12 shows that the "two degrees of freedom" workload behaves similarly to the (one or) two dimensional uniform workload, regardless of the dimensionality. However, the recursive workload (as predicted by our theorem) was affected by dimensionality. More interestingly, even with all the correlation and changing variances, the recursive workload behaved almost the same as the IID uniform case!

These graphs 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.

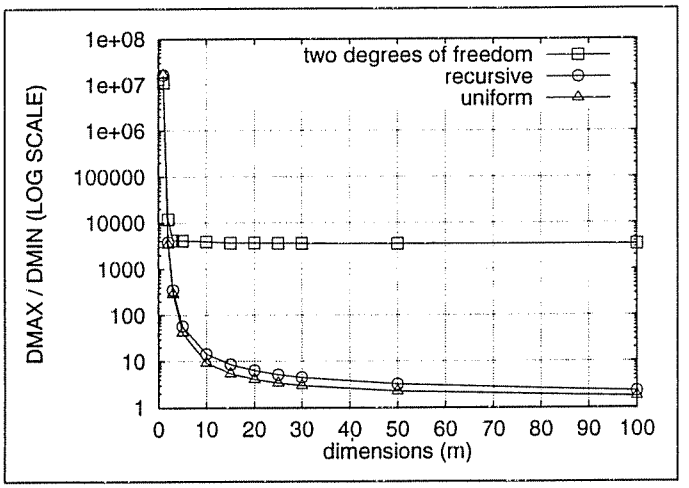

Figure 12: Correlated distributions, one million tuples.

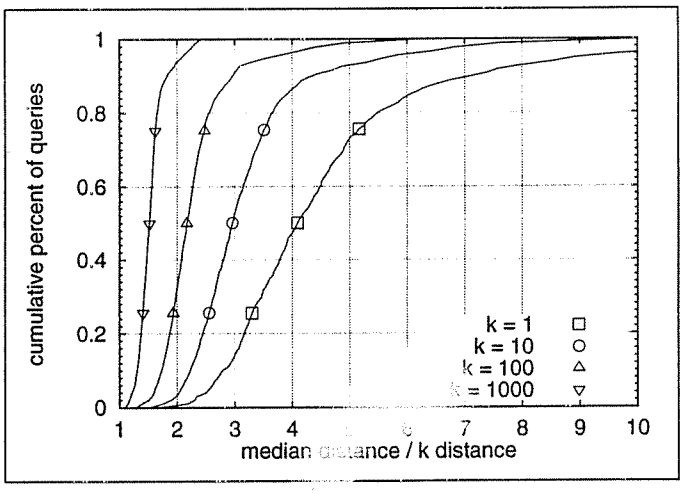

Figure 13: 64-D color histogram data.

5.2 Case Study: Two Datasets from an Image Database System

In this section, we examine the quality of k-NN queries for two image database datasets. We examine k-NN rather than NN because this is the traditional application of these datasets. The first dataset was a 256 dimensional color histogram dataset (one tuple per image) that was reduced to 64 dimensions by principal components analysis. There were approximately 13,500 tuples in the dataset.

To determine the quality of answers for NN queries, we examined the percentage of queries in which at least half the data points were within some factor of the nearest neighbor. The results are in Figure 13. Examine the graph at *median distance/k distance* = 3. The graph says that for $k = 1$ (normal NN problem), 15% of the queries had at least half the data within a factor of 3 of the distance to the NN. For $k = 10$, 50% of the queries had at least half the data within a factor of 3 of the distance to the 10th nearest neighbor. It is easy to see that the effect of changing k on the quality of the answer is most significant for small values of k.

Does this dataset provide meaningful answers to the 1-NN problem? the 10-NN problem? the 100-NN problem? Perhaps, but keep in mind that intuitively, most people would expect that the *median distance/k distance* ratios (on the X-axis) be more in the range of 1000-10,000.

Our other dataset is taken from the same underlying image dataset and contains vectors of 324 dimensions. These dimensions correspond to a combination of vectorization techniques that the image database designers felt reflected

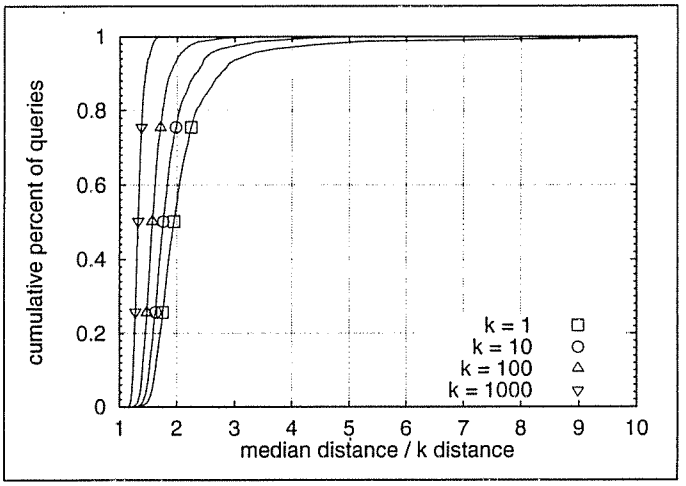

Figure 14: 324-D image feature data.

the meaningful attributes of an image better than just a color histogram. For instance, it contains information about shapes found within the image. The same experiments were run as in the previous dataset. The results in Figure 14 show that half the data typically lies much closer to the NN than in the previous dataset. This lack of contrast in this dataset suggests that this dataset is not as good at grouping similar things as the other, contrary to the database designers' intuition.

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., [27, 17, 12, 9, 10]) 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, most papers do not compare against the trivial linear scan algorithm. Given our results, which suggest that most of the data must be examined as dimensionality increases, it is not surprising to discover that at relatively few dimensions, linear scan handily beats these (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 [9] 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.

[27] 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 [17], 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 [12], 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 4 *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. One can use the existing methods of processing k nearest neighbors queries for solving this new variant of the problem.

Graphs like Figure 11 in Section 5.1 provide an 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. 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 Related Work

8.1 Computational Geometry

The nearest neighbor problem has been studied in computational geometry (e.g., [4, 5, 6, 8, 11]). However, the usual approach is to take the number of dimensions as a constant and find algorithms that behave well when the number of points is large enough. They observe that the problem is hard and define the approximate nearest neighbor problem as a weaker problem. In [6] there is an algorithm that retrieves an approximate nearest neighbor in $O(\log n)$ time for any data set. In [8] there is an algorithm that retrieves the true nearest neighbor in constant expected time under the IID dimensions assumption. However, the constants for those algorithms are exponential in dimensionality. In [6] they recommend not to use the algorithm in more than 12 dimensions. It is impractical to use the algorithm in [8] when the number of points is much lower than exponential in the number of dimensions.

8.2 Boundary Effects and Index Structure Utility

In [10, 5] it was observed that in some high dimensional cases, the estimate of NN query cost (using some index structure) can be very poor if "boundary effects" are not taken into account. The boundary effect is that the query region (i.e., a sphere whose center is the query point) is mainly outside the hyper-cubic data space. When one does not take into account the boundary effect, the query cost estimate can be much higher than the actual cost.

In this paper we do not discuss the cost of using an index structure. Instead, we discuss the utility of the query itself (regardless of the processing method). Moreover, in situations where nearest neighbors queries are unstable its hardly useful to process them or even estimate the cost of processing them. In addition, boundary effects are not explicitly related to distance distributions.

8.3 Fractal Dimensions

In [15, 7, 14] it was suggested that real data sets usually have fractal properties (self-similarity, in particular) and that fractal dimensionality is a good tool in determining the performance of queries over the data set.

The following example illustrates that the fractal dimensionality of the data space from which we sample the data points may not always be a good indicator for the utility of nearest neighbor queries. Suppose the data points are sampled uniformly from the vertices of the unit hypercube. The data space is 2^m points (in m dimensions), so its fractal dimensionality is 0. However, this situation is one of the worst cases for nearest neighbor queries. (This is actually the IID Bernoulli(1/2) which is even worse than IID uniform.) When the number of data points in this scenario is close to 2^m, nearest neighbor queries become stable, but this is impractical for large m.

However, are there real data sets for which the (estimated) fractal dimensionality is low, yet there is no separation between nearest and farthest neighbors? This is an intriguing question which we intend to explore in future work.

(We used the technique described in [7] on the two real data sets described in Section 6. However, the fractal dimensionality of those data sets could not be estimated (when we divided the space once in each dimension, most of the data points occupied different cells). We used the same technique on an artificial 100 dimensional data set that has known fractal dimensionality 2 and about the same number of points as the real data sets (generated like the "two degrees of freedom" workload in Section 5.1, but with less data). The estimate we got for the fractal dimensionality is 1.6 (which is a good estimate). Our conclusion is that the real data sets we used are inherently high dimensional; another possible explanation is that they do not exhibit fractal behavior.)

9 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 situations in which NN queries do not break down in high dimensionality. In particular, the ideal datasets and workloads for classification/clustering algorithms seem reasonable in high dimensionality. Although, if the scenario is deviated from (for instance, if the query point does not 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 between a query point and the nearest and farthest data points drops below a factor of 3. In addition to simulated workloads, we also examined two real datasets that behaved similarly.

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 can 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. Arya. *Nearest Neighbor Searching and Applications.* PhD thesis, Univ. of Maryland at College Park, 1995.

[5] S. Arya, D. M. Mount, and O. Narayan. Accounting for boundary effects in nearest neighbors searching. In *Proc. 11th ACM Symposium on Computational Geometry*, pages 336–344, 1995.

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

[7] A. Belussi and C. Faloutsos. Estimating the selectivity of spatial queries using the 'correlation' fractal dimension. In *Proc. VLDB*, pages 299–310, 1995.

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

[9] 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.

[10] 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.

[11] M. Bern. Approximate closest point queries in high dimensions. *Information Processing Letters*, 45:95–99, 1993.

[12] 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.

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

[14] 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.

[15] C. Faloutsos and I. Kamel. Beyond uniformity and independence: Analysis of R-trees using the concept of fractal dimension. In *Proc. 13th ACM SIGACT-SIGMOD-SIGART Symposium on PODS*, pages 4–13, 1994.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

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

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