Nearest-Neighbor Searching and Metric Space Dimensions

Kenneth L. Clarkson[*]

October 23, 2005[†]

Abstract

Given a set S of points in a metric space with distance function D, the *nearest-neighbor searching* problem is to build a data structure for S so that for an input query point q, the point $s \in S$ that minimizes $D(s, q)$ can be found quickly. We survey approaches to this problem, and its relation to concepts of metric space dimension. Several measures of dimension can be estimated using nearest-neighbor searching, while others can be used to estimate the cost of that searching. In recent years, several data structures have been proposed that are provably good for low-dimensional spaces, for some particular measures of dimension. These and other data structures for nearest-neighbor searching are surveyed.

1 Introduction

The problem of *nearest-neighbor search* is to build a data structure for a set of objects so that, given a query object q, the nearest object in the set to q can be found quickly.

That is, suppose $\mathbb{U}$ is a set and D is a *distance measure* on $\mathbb{U}$, a function that takes pairs of elements of $\mathbb{U}$ and returns a nonnegative real number. Then given a set $S \subset \mathbb{U}$ of size n, the *nearest-neighbor searching* problem is to build a data structure so that, for an input query point $q \in \mathbb{U}$, an element $a \in S$ is found with $D(q, a) \leq D(q, x)$ for all $x \in S$. We will call the members of S *sites*, to distinguish them from other members of $\mathbb{U}$, and say that the answer a is *nearest* in S to q. Put another way, if we define $D(x, S)$ as $\min\{D(x, s) \mid s \in S\}$, then we seek the site s such that $D(q, s) = D(q, S)$.

This problem has been studied for a long time, and has many names in a large and diverse literature. In an early proposal for a solution, due to McNutt (as discussed by Knuth[Knu98]), it was called the *post office problem*. In another

[*]Bell Labs; 600 Mountain Avenue; Murray Hill, New Jersey 07974; `clarkson@research.bell-labs.com`

[†]Previous version: April 2005; Original: February 21, 2005

Contents

early proposal, it was called *best-match file searching* [BK73]. In the database or information-retrieval literature, it might be called the problem of *building an index for similarity search* [HS03]. In the information theory literature, it arises as the problem of *building a vector quantization encoder* [LBG80, GN93]. In the pattern recognition (or statistics or learning theory) literature, it might be called the problem of building a *fast nearest-neighbor classifier* [FJ51, DGL96].

This chapter surveys the problem of nearest neighbor searching in the general metric space setting, together with additional dimensionality properties that make instances of the problem tractable. The meaning of "tractable" here is vague, but mainly refers to data structures that are not too big, but allow queries that are fast, where "not too big" means roughly $O(n)$, and "fast" means $o(n)$, as $n \to \infty$.

Some basic constructions of metric spaces are also reviewed, as are some ways of "repairing" a distance measure that does not quite satisfy all the properties needed for a metric space.

Some concepts of dimension we consider include the *Assouad* dimension, the *box* dimension, and a dimension based on *doubling measures.* These concepts have been studied in measure theory and harmonic analysis. As discussed in Subsection 5.2, the theoretical computer science community has in recent years begun to study spaces of bounded Assouad dimension, and some of their results include provably efficient data structures for nearest-neighbor searching [Cla99, KL04, HPM05]. Some stronger efficiency results have also been given for spaces satisfying a stronger condition based on *doubling measure* [KR02, BKL04, HKMR04]. Some of these algorithms, or related ones, have been implemented, with promising results [Cla03, BKL04]. Also, some experimental results have been obtained regarding the correlation dimension of a space and the cost of some nearest-neighbor searching problems [BF98].

These results are described here in the context of the large variety of algorithms and data structures that have been proposed for nearest-neighbor searching. It is remarkable that there are so many such algorithms, especially considering that the distance measure is used simply as a "black box" function that takes two objects and returns a nonnegative real number.

To put these applications of dimensional properties in perspective, we survey a variety of dimensional concepts for metric spaces and for metric measure spaces, and relate them to nearest neighbors. In addition to the box and Assouad dimensions, we consider also for metric spaces the Hausdorff and packing dimensions. For metric measure spaces, the pointwise, energy, and quantization dimensions are discussed, as well as doubling measures, and also the general Rènyi dimensions, which include the information and correlation dimensions. Nearest-neighbor searching is a key component of several estimators of these dimensions, while some estimates of dimension allow bounds for costs related to nearest-neighbor queries. These relations are discussed in Section 5.

As an example of the interplay of dimensions and nearest neighbors, suppose a metric space has a measure. Here we will assume that the measure is a probability distribution. The *pointwise dimension* at point x tells how quickly the measure of a ball $B(x, \epsilon)$ centered at x goes to zero, as its radius ϵ goes to zero. The pointwise dimensions of points in the space are closely related to the Hausdorff, information, and energy dimensions of the measure. Now suppose a set of n sites S comprises random points that are independently distributed according to the given measure. It turns out that the pointwise dimension at x can be estimated as the ratio of $\log n$ to the distance of x to its nearest neighbor in S. That is, the pointwise dimension is related to how fast the nearest-neighbor distance goes to zero as n goes to infinity. This is discussed in Subsection 4.2. The pointwise dimension has been proposed in the database literature as a way to determine how large the answer to a fixed-radius neighbor query is likely to be. (Such a query asks for all sites inside a given ball.) It is a basic concept of *multifractal* analysis, as used in studying dynamical systems [Pes97].

As another example: ϵ-nets are a kind of well-distributed subset of a metric space, such that every point in the space is within distance ϵ of the net. The box dimension of the space determines the rate at which the size of such nets increases, as a function of $1/\epsilon$. There is a greedy algorithm for finding ϵ-nets that has been applied to building data structures for nearest-neighbor searching,[Bri95, Woj03, Cla03, HPS05] as well as other optimization problems [Gon85]. These relations are discussed in Section 4 and Subsection 5.2.4.

1.1 Scope, and Survey of Surveys

There are many important aspects of nearest-neighbor searching that are not covered here, but have been surveyed elsewhere.

Several surveys of nearest-neighbor searching in $\Re^d$ have been done: one focuses on high-dimensional spaces[Ind04]; another on closest-pair problems, including insertions and deletions of sites [Smi00]; and another [AGE$^+$02] on data structures to allow *moving* sites to be handled efficiently [Ata85, Kah91, BGH97]. A recent survey [BBK01] and book [PM05] describe nearest-neighbor searching from a database perspective.

There are at least two prior surveys of searching in general metric spaces [CNBYM01, HS03]. These surveys discuss in detail many algorithms that have arisen in practice.

The primary concern here is with reducing the number of distance evaluations needed to answer a query. There is a substantial body of work on increasing the efficiency of search by speeding up distance evaluations. Some of these techniques are simple and practical, such as avoiding the evaluation of square roots, or stopping distance evaluations when the distance value is known accurately enough. Other techniques show that distances can be estimated quickly using randomization. Related techniques involve the “embedding” of the metric space in a low-dimensional space, a very active area of research [IM04].

The basic conditions for provably fast search that we consider relate to various concepts of dimension, many of which include the possibility of nonintegral, or *fractal* dimension [Fal90, Edg98]. A rigorous, thorough, and accessible introduction to fractal dimension, including statistical considerations for estimators of dimension, is given by Cutler[Cut93].

1.2 Caveat Lector

The algorithms described in Section 3 are given only the most cursory overview. The discussion of dimension in Section 4 may well neglect some basic conditions needed for mathematical rigor, while the discussion of algorithms in Section 5 may not accurately reflect the data structures that are being abstracted and simplified. The bibliography should be helpful, however.

1.3 Related Problems and Applications

Nearest-neighbor problems arise in many application areas, including information retrieval, classification, data compression, databases, and dynamical systems. The tasks of vector quantization and nearest-neighbor classification are illustrative.

The *vector quantization* problem is the following: let X denote a random variable in $\mathbb{U}$, with distribution μ. An *n-quantizer* for μ is a function f on $\mathbb{U}$ that takes a point X to one of a set of at most n points in $\mathbb{U}$. That set is called the *codebook*. Let $\mathcal{F}_n$ be the set of all n-quantizer functions. Then the nth quantization error for $\mathbb{U}$ is

$$V_n(\mu) := \inf_{f \in \mathcal{F}_n} E\rho(D(X, f(X))), \tag{1}$$

the cheapest expected cost of representing X by $f(X)$, where the cost is a function $\rho()$ of the distance of X to $f(X)$, and the expectation is with respect to the distribution μ. If the value X was information to transmit over a channel, then an identifying number for $f(X)$ from 1 to n could be transmitted instead, and $f(X)$ recovered from that number on the other end of the channel. The quantization error is the expectation for how badly the received value $f(X)$ distorts the true value X.

Often $\mathbb{U}$ is $\Re^d$, the distance is Euclidean, and $\rho(a) = a^v$ for some value $v > 0$, in which case the nth quantization error is of *order v*.

Nearest-neighbor searching arises in vector quantization as the task of implementing f, after the codebook set has been chosen. Sometimes the codebook is *structured* in a way that allows fast and easy search, as for example when it comprises points on a regular grid. However, for a given distribution, the optimal codebook may be *unstructured*, and look like nothing but an arbitrary point set. (For such unstructured codebooks, the need for fast encoding has led to the use of *tree* encoders, which are not guaranteed to answer a query with the nearest site, or even a near one. However, such encoders work well in practice.) Vector quantization, and the *quantization dimension*, are discussed further in §4.3.

As noted above, the problem of classification has long been approached using nearest-neighbor searching [DGL96]. Points in the space (typically $\Re^d$) correspond to sets of objects, and the point coordinates encode various properties of the objects. Each object also has a "color," say red or blue, corresponding to some additional important property. The sites S are a "training set," each of known color. A nearest-neighbor classifier takes as input a query point of unknown color, and returns the color of the nearest site in the training set, or the color of the majority of the nearest k sites. That returned value is a prediction of the true color of the query point.

Thus the nearest-neighbor searching problem arises in finding the closest sites in the training set. Note that, however, it is not necessary to find the nearest sites, but only their colors. It is sometimes possible to use this simplification of the problem to obtain a faster algorithm [MNSW00, LMG04].

We turn now to other computational problems, closely related to nearest-neighbor searching, that arise in applications. One mentioned for classification is *k-nearest neighbors* (k-NN): given an integer k and query point q, find the k sites that are closest to q. That is, nearest-neighbor searching is the special case of k-NN searching with $k = 1$. Another related problem is distance *range searching*: build a data structure so that given distance value r and query point q, all sites $p \in S$ with $D(q, p) \leq r$ can be found quickly. If we were given the nearest-neighbor distance $D(q, S)$ by an oracle, then answering the range query with parameter $r = D(q, S)$ would answer the nearest neighbor query.

Approximate Queries. Sometimes it may not be necessary to find the nearest neighbor, but only a *(δ)-near* neighbor, that is, one whose distance is within a δ factor of the nearest distance, for some $\delta > 1$. (Note the distinction between "k-nearest" and "(δ)-near.") Such *approximate* nearest neighbor queries are of interest in their own right, and may have much faster algorithms than those for nearest neighbor queries. Moreover, near neighbors can sometimes be used to find nearest neighbors, as discussed in Subsection 5.2.

Reverse Queries. Another related problem is that of building a data structure for *reverse* or *inverse* nearest neighbor queries, where the input is similar, but the answer is not the site nearest to query point q, but rather the sites that have q as their (second) nearest neighbor in $S \cup \{q\}$, that is, the answer is the set of sites

$$\{s \in S \mid D(s, q) \leq D(s, S \setminus \{s\})\}.$$

As with (forward) nearest-neighbor searching, this problem also can be generalized with k and ϵ: given k and site q, find all sites that have q as kth nearest neighbor, or given $\epsilon > 0$, find all sites such that q is within $1 + \epsilon$ of closest. This problem has arisen as a computational bottleneck in event-driven astrophysical simulations [AT01], and as a notion of "influence" in decision support and referral systems[KM00, SFcT03]. It also arises as a subproblem in building data structures for nearest-neighbor queries, as mentioned in Subsection 5.2.4.

A key property for reverse queries is that in some circumstances, such as those given for Lemma 5.1, the answer size is bounded by a value independent of $|S|$, the number of elements of S. The intuition, considering points in the plane, is that as more and more sites have q as a nearest neighbor, at some point two of the sites must be closer to each other than to q.

Batched Queries. There are several general problems that might be solved using a data structure for nearest-neighbor searching, or k-NN searching. For example, the *closest-pair* problem is to find the two sites s and s' such that

$D(s, s') = \min\{D(p, p') \mid p, p' \in S, p \neq p'\}$. This could be solved by applying a data structure for 2-nearest-neighbors to each site in turn. (Here we must have 2-nearest-neighbors because the closest site to a site s is s itself.) Similarly, the *all-k-nearest-neighbor* (all-k-NN) problem is to find, for each site s, the k sites closest to s. Solving this problem is a common preprocessing step for "manifold reconstruction" in the computational geometry[FR02], learning theory[SR03], and computer graphics[Hor03] literatures. Note that the answer to the closest-pair problem can easily be found using the answer to the all-k-NN problem. Similarly, the max-min distance

$$\max_i \min_j D(s_i, s_j),$$

which has been proposed as a diversity measure [AL99], can be found among the all-k-NN output. The *correlation integral* problem is a range query analog of all-nearest neighbors: given a value $r > 0$, find all pairs of sites within distance r of each other. This problem arises in computing the *correlation dimension* of S, discussed in Subsection 4.2.

Bichromatic Problems In addition to the "chromatic" problem of nearest-neighbor classification mentioned above, another class of problems is *bichromatic*. The input is two sets S and S', and the closest pair of sites, one from each set, is desired. (That is, sites in S are "red," and those in S' are "blue," and the closest two-color pair is wanted, hence the problem is bichromatic.) Another bichromatic problem is called a kind of *spatial join* in the database literature: given distance value D, find for each site $s \in S$, the sites in S' that are within distance D of s [BF98].

2 Metric Space Properties, Construction, and Repair

So far, we have only described nearest-neighbor searching problems in great generality, and have not even given any properties of the distance measure D, except that it maps from pairs of points to real numbers. In instances of these problems, $\mathbb{U}$ and D have many properties that can be used to obtain solutions to the nearest-neighbor problem. The main property that often holds is that $(\mathbb{U}, D)$ is a *metric space*, described next. An additional condition that often applies is that $(\mathbb{U}, D)$ has bounded dimension, for some concept of dimension, as discussed in Section 4.

2.1 Metric Spaces

The distance function D of a metric space $(\mathbb{U}, D)$ satisfies the following conditions, for all $x, y, z \in \mathbb{U}$:

1. nonnegativity: $D(x, y) \geq 0$;
2. small self-distance: $D(x, x) = 0$;

3. isolation: $x \neq y$ implies $D(x, y) > 0$;
4. symmetry: $D(x, y) = D(y, x)$;
5. the triangle inequality: $D(x, z) \leq D(x, y) + D(y, z)$.

2.2 Distance Measure Repairs

A great many instances of nearest-neighbor searching naturally have an associated metric space. Moreover, it is worth noting that if any one of the conditions 3 to 5 fails, while the others hold, there is a natural associated function that is a metric, described next.

Condition 3, isolation, fails: here $(\mathbb{U}, D)$ is a called a *pseudometric*. Partition $\mathbb{U}$ into equivalence classes based on D, where x and y are equivalent if and only if $D(x, y) = 0$. With the natural distance $D([x], [y]) = D(x, y)$ on the classes, the result is a metric space.

Condition 4, symmetry, fails: $(\mathbb{U}, D)$ is a *quasi-metric*. The related measure $\hat{D}(x, y) := (D(x, y) + D(y, x))/2$ will satisfy symmetry, and so yield a metric space.

Condition 5, the triangle inequality, fails: a *semimetric*, or *positively-weighted undirected graph*. A related metric can be found using *shortest paths*: let

$$\hat{D}(x, y) := \inf \sum_i D(z_i, z_{i+1}),$$

where the infinum is taken over all sequences in $\mathbb{U}$ of the form

$$x = z_1, z_2, \ldots, z_N = y,$$

for all $N > 1$. Note that $\hat{D}$ satisfies the triangle inequality, and is a metric, possibly after patching up the "small self-distance" condition. This is the shortest path distance in the graph whose vertices are the points, a *graph metric*.

This repair of the triangle inequality is often used in the other direction: given a finite metric space $(\mathbb{U}, D)$, a graph with vertex set $\mathbb{U}$ and with few edges is found, such that the resulting graph metric is a good approximation to the original metric D. Such graphs are called *spanners*; these have been the focus of considerable research and application.

Another conceivable repair for the triangle inequality is to use $\hat{D}(x, y) := D(x, y)^{1/w}$; for sufficiently large w, $\hat{D}$ satisfies the triangle inequality. If only $w = \infty$ will suffice, then the uniform metric ($D(x, y) = 1$ if $x \neq y$) is the resulting $\hat{D}$. Otherwise, with $w < \infty$, this approach might be of interest, since it preserves inequalities among distances, so the nearest neighbor in D is also the nearest in $\hat{D}$. For finite spaces, $\max_{x,y,z \in \mathbb{U}, x \neq y} \log_2(D(x, z)/D(x, y))$ would be large enough, for example; this quantity is bounded by the *spread*, which is discussed in Subsection 5.2. Note that this transformation preserves distance rank: if y is farther from x than z, it will also be under the repaired version. So distance measures that do not obey the triangle inequality might be transformed into metrics, with the answers to nearest-neighbor queries preserved. On the other hand, this transformation flattens the distance, and so may make clusters less distinct, and degrade some searching algorithms.

The repair of quasi-metrics given above is computationally trivial. Pseudometrics do not really need "repair" for nearest-neighbor searching: it is only necessary to keep in mind that the answer is a representative of an equivalence class, and the possibility that distinct sites have distance zero. The repair of the triangle inequality may be difficult to apply in the context of nearest-neighbor searching (although see the string edit distance below). However, graph metrics are a concept of considerable interest and importance in optimization [IM04]. Given an arbitrary distance function (mapping from ordered pairs to the nonnegative reals) that has $D(x, x) = 0$, an associated metric could be found by using shortest paths to obtain a function that satisfies the triangle inequality, then averaging to enforce symmetry, and finally grouping into equivalence classes to achieve isolation.

2.3 Metric Space Constructions

One very basic metric space for any given set $\mathbb{U}$ is, as noted, the uniform metric, where for all $x, y \in \mathbb{U}$, $D(x, y) = 1$ if $x \neq y$, and $D(x, x) = 0$. Another basic space is the set of real numbers $\Re$, with distance $|x - y|$ for $x, y \in \Re$. Moreover, metric spaces can be constructed from other spaces. In the following, suppose $(\mathbb{U}, D)$ is a metric space, as are some $(\mathbb{U}_1, D_1) \dots (\mathbb{U}_d, D_d)$.

- Submetrics. Plainly, any $(\mathbb{U}', D')$, where $\mathbb{U}' \subset \mathbb{U}$ and D' is D restricted to $\mathbb{U}' \times \mathbb{U}'$, is a metric space.

- Products. Let $\hat{\mathbb{U}}$ be the cross-product $\mathbb{U}_1 \times \mathbb{U}_2 \times \dots \mathbb{U}_d$, that is, the d-tuples over the $\mathbb{U}_i$. For some value p with $1 \leq p \leq \infty$, define $\hat{D}$ as follows: for $x, y \in \hat{\mathbb{U}}$, let

  $$\hat{D}(x, y) := \left(\sum_i D_i(x_i, y_i)^p \right)^{1/p},$$

  the product metric. When all $\mathbb{U}_i = \Re$ and $D_i(x, y) = |x - y|$, this yields $\Re^d$ with the ℓ_p distance measures, so $\hat{D}(x, y) = \|x - y\|_p$. When $p = d = 2$, this is simply the Euclidean plane. When $p = 1$ and all the D_i are the uniform metric, the result is the Hamming distance.

- Strings. Let $\mathbb{U}^*$ denote the strings over $\mathbb{U}$. Suppose $\hat{D}$ is a distance measure on $\mathbb{U}^*$ defined as follows: when deletion or addition of one character from x yields y, then $\hat{D}(x, y) = 1$; when replacement of a character a in x by a character b yields y, then $\hat{D}(x, y) = D(a, b)$. Then $(\mathbb{U}^*, \hat{D})$ is a semimetric, and its shortest path "repair," as discussed above, is called the string edit, or Levenshtein distance. In other words, the string edit distance between $x, y \in \mathbb{U}^*$ is the minimum cost sequence of deletion, insertion, or replacement operations to obtain y from x. If deletion and insertion have infinite cost, then this is a kind of Hamming distance on strings. This measure might be used for spelling correction, and for comparing genetic sequences.

- Subsets. The Hausdorff distance between subsets of $\mathbb{U}$ is

  $$\hat{D}(S,T) := \min\{D'(S,T), D'(T,S)\},$$

  where

  $$D'(S,T) := \sup_{s\in S}\inf_{t\in T} D(s,t).$$

  Such a distance might be used for geometric shapes. (Technically, this is only a pseudometric, but it is a metric for all closed bounded subsets.)

  Another commonly used distance between subsets is

  $$D(S,T) := \inf_{s\in S, t\in T} D(s,t).$$

  Note that this is not a metric.

  When $\mathbb{U}$ has a measure μ, the distance $\mu(A\Delta B)$ has been studied, where $A\Delta B$ is the symmetric difference of A and B; this metric generalizes the Hamming distance.

- Nonnegative combinations. Suppose the $\mathbb{U}_i$ are all equal, a set $\mathbb{U}$, but the D_i are different. Given $\alpha_1 \ldots \alpha_d$ with $\alpha_i \geq 0$, define $\hat{D}$ by $\hat{D}(x,y) := \sum_i \alpha_i D_i(x,y)$. Then $(\mathbb{U}, \hat{D})$ is a metric, a nonnegative combination of the originals. (In particular, scaling a single metric by a positive constant also gives a metric.) In other words, the set of metrics on $\mathbb{U}$ is closed under nonnegative combination, and forms a cone; such cones are well studied [DL97].

- Metric Transforms. If f is a real-valued function of the nonnegative reals, and $f(0) = 0$, and $f(z)$ is monotone increasing and concave for $z \geq 0$, then $\hat{D}(x,y) := f(D(x,y))$ is a metric [DL97]. For example, if f is twice differentiable, $f'(z) \geq 0$, and $f''(z) \leq 0$ for $z \geq 0$, then f is monotone increasing and concave. One such function is $f(z) := z^\epsilon$, for any given ϵ with $0 < \epsilon < 1$. The new metric $D(x,y)^\epsilon$ is sometimes called the *snowflake* or *power transform* of the original. The function with $f(z) = z/(1+z)$ also satisfies the given conditions, and yields a bounded distance measure.

- Steinhaus Transform. If (U, D) is a metric space and $a \in U$, then $(U, \hat{D})$ is also a metric space, where

  $$\hat{D}(x,y) := \frac{2D(x,y)}{D(x,a) + D(y,a) + D(x,y)}.$$

  This is sometimes called the *Steinhaus transform*[DL97].

When this transform is applied to the distance $D(A, B) = \mu(A\Delta B)$, and with a being the null set Φ, the result is

$$\begin{aligned}\hat{D}(A, B) &= \frac{2\mu(A\Delta B)}{\mu(A\Delta\Phi) + \mu(B\Delta\Phi) + \mu(A\Delta B)} \\ &= \frac{2\mu(A\Delta B)}{\mu(A) + \mu(B) + \mu(A\Delta B)} = \frac{2\mu(A\Delta B)}{2\mu(A \cup B)} \\ &= \frac{\mu(A\Delta B)}{\mu(A \cup B)},\end{aligned}$$

which is called the *Steinhaus distance* [DL97]. The special case for finite sets $|A\Delta B|/|A \cup B|$ is called the *Tanimoto distance* [RT60], *resemblance* [Bro97], *set similarity distance* [Cha02], *Jaccard distance* [Jac01, Spa80], or *Marczewski-Steinhaus distance* [MS58]. It has been proven a metric in several ways [DL97, Spa80, XA03, Cha02].

The above follows in part Deza and Laurent [DL97], and also Indyk and Matoušek [IM04]; the latter describe other metric constructions, including the *earth-mover* (or *Mallows*[LB01]), *Fréchet*, and *block-edit* distances.

The above hardly exhausts the distances and metrics that have been considered, even by applying the constructions repeatedly. For example, for two probability distributions on $\mathbb{U}$ with density functions f and g, the α-divergence of f and g is

$$\frac{1}{\alpha - 1} \ln \int f^{\alpha} g^{1-\alpha},$$

which has the Kullback-Leibler ($\alpha \to 1$) and Hellinger ($\alpha = 1/2$) divergences as special cases. This is not a metric, however.

There is even a distance measure between metric spaces, which can be defined for spaces Z and Z' as the Hausdorff distance between $\kappa(Z)$ and $\kappa(Z')$, where these are Kuratowski embeddings of Z and Z', as mentioned in Subsection 3.1, and the embeddings are chosen to minimize the Hausdorff distance, among all such embeddings [Hei03].

The Pearson Correlation Distance. A distance measure on $\Re^d$ commonly used for biological sequences is derived from the *Pearson correlation*: For point $x = (x_1, \ldots, x_d) \in \Re^d$, let $\eta := \sum_i x_i/d$, and $x' := (x_1 - \eta, x_2 - \eta, \ldots, x_d - \eta)$, and finally $\hat{x} := x'/\|x'\|_2$. That is, the coordinates of x are normalized to have mean zero, and to have $\|\hat{x}\|_2 = 1$, a unit vector in the Euclidean norm. The Pearson correlation of $x, y \in \Re^d$ is then the dot product $\hat{x} \cdot \hat{y}$. The commonly-used derived distance measure is $1 - \hat{x} \cdot \hat{y}$. While this measure does not satisfy the small self-distance or triangle inequality conditions for a metric, note that

$$\|\hat{x} - \hat{y}\|_2^2 = \hat{x} \cdot \hat{x} + \hat{y} \cdot \hat{y} - 2\hat{x} \cdot \hat{y} = 2(1 - \hat{x} \cdot \hat{y}).$$

That is, the square root of the commonly used measure is proportional to the ordinary Euclidean distance between $\hat{x}$ and $\hat{y}$. Therefore, only the small self-distance condition fails for this variant, and metric space (even Euclidean space) methods can be used.

3 Using Triangle Inequalities

3.1 Triangle Inequality Bounds

The properties of metric spaces allow some basic observations that can yield significantly faster algorithms for nearest-neighbor searching. These follow from the triangle inequality, which allows bounds on a distance we may not have computed, say $D(q,s)$, to be derived from two distances we may already know, say $D(q,p)$ and $D(p,s)$. The following simple properties hold.

Lemma 3.1 *For $q, s, p \in \mathbb{U}$, any value r, and any $P \subset \mathbb{U}$,*

1. $|D(p,q) - D(p,s)| \le D(q,s) \le D(q,p) + D(p,s)$;
2. $D(q,s) \ge D_P(q,s) := \max_{p \in P} |D(p,q) - D(p,s)|$;
3. *if* $D(p,s) > D(p,q) + r$ *or* $D(p,s) < D(p,q) - r$, *then* $D(q,s) > r$;
4. *if* $D(p,s) \ge 2D(p,q)$, *then* $D(q,s) \ge D(q,p)$.

Proof: Applying the triangle inequality in the three possible ways,

$$\begin{aligned} D(q,s) &\le D(q,p) + D(p,s) \\ D(p,s) &\le D(p,q) + D(q,s) \\ D(q,p) &\le D(q,s) + D(s,p) \end{aligned}$$

The first of these is the upper bound for $D(q,s)$ in (1), and the other two imply the lower bound of (1). Claim (2) follows from (1), the two parts of Claim (3) follow from the last two inequalities, respectively, and Claim (4) follows from Claim (3) with $r = D(p,q)$. ■

The value $D_P(q,s)$, that is a lower bound for $D(q,s)$, is used in the AESA algorithm, as discussed below(§3.2.2).

If sites in $\mathbb{U}$ are represented by the vector of their distances to P, then $D_P(q,s)$ is the ℓ_∞ (coordinate-wise maximum) distance between the representatives of q and s. Because $D_P(q,s) \le D(q,s)$, the mapping from the original $(\mathbb{U}, D)$ to $(\Re^{|P|}, D_\infty)$ is said to be *contractive*; such contractive mappings can be helpful in distance range searching: if the problem is mapped to the vector representation, then the answer to a query corresponds to a superset of the answer in the original space [HS00].

Moreover, $D_P(q,s) \le D(q,s)$ is an equality if both q and s are in P. That is, the sites $s \in P$ can be represented by the vector of their distances to P, and the ℓ_∞ (coordinate-wise maximum) distance between those vectors is the original distance. This shows that any finite space of m sites can be embedded in the ℓ_∞ space of dimension m. This embedding is due to Kuratowski [Hei03, IM04].

3.2 Orchard's Algorithm, AESA, Metric Trees

The above bounds from the triangle inequality give a way to avoid computing the distance from a query point q to many of the sites, by giving bounds on their distance that allow the sites to be ruled out as nearest.

3.2.1 Orchard's Algorithm

For example, consider the following simple scheme. For each site p, create a list of sites in increasing order of distance to p.

To find the closest site to a query point q, pick some site c as an initial candidate for the nearest site. Compute $D(c, q)$, and walk along the list for c, computing distances to the sites on the list. If some site s is closer to q than c, set $c := s$. Now repeat the same procedure, using the new c and its list, and so on. Suppose some such list is traversed to a site s with $D(c, s) > 2D(c, q)$. Then by Lemma 3.1(4), c is the closest site: any remaining site on the list for c must be farther from q than c is. (Here c takes the role of p in the lemma.)

This algorithm, due to Orchard[Orc91], is simple and fast[ZJL02], particularly in high dimension ($\mathbb{U}$ is $\Re^{64}$, for example). However, it needs $\Omega(n^2)$ preprocessing, making it inappropriate for large databases. Even worse, it needs $\Omega(n^2)$ storage. For many applications this is fatal. However, for the target application of vector quantization, the preprocessing and storage costs can be acceptable.

Orchard's algorithm is an instance of a "traversal" method, and so can be accelerated using the *skip list* technique, as discussed in Subsection 5.2.3.

One refinement for Orchard's algorithm is to ensure that the distance from q to any given site is computed only once per query; one way to do this is to keep a mark bit for each site, which is initially zero for all sites. When the distance to a site is computed, the mark bit is set to one, and the site is entered in a linked list. When a site is considered for distance computation, if the mark bit is set to one, the site can be ignored: it cannot be closer than the current site. After a query, the linked list is walked, and the mark bits are set to zero for sites on the list. Such a scheme allows the mark bits to be maintained in a time proportional to the number of distance evaluations.

The Annulus Method. To ease the storage burden, a different scheme is to keep only one of the sorted lists for Orchard's algorithm, proceeding as follows. For some site p^*, build a list of the other sites and their distances to p^*, sorted by increasing distance. As in Orchard's algorithm, maintain a candidate closest site c. To find sites closer to q than c, walk on the list for p^* from the position of c, alternately in each direction, and compute distances. As in Orchard's algorithm, if a site s is found that is closer to q than c, set $c := s$ and continue. If a site s on the lower side has $D(p^*, s) < D(p^*, q) - D(c, q)$, then no further sites on the lower side need be considered, by Lemma 3.1(3). (Here the r of the lemma is $D(c, q)$, and the p of the lemma is p^*.) Similarly, if a site on the higher side has $D(p^*, s) > D(p^*, q) + D(c, q)$, then no further sites on the higher side need be considered. If both conditions hold, then the current candidate c is closest.

Orchard's method, the annulus method, and other methods are discussed and tested in [ZJL02].

3.2.2 AESA

The Approximating and Eliminating Search Algorithm, or AESA [Vid86, Vid94], applies the bounds of Lemma 3.1 in a more thorough way than Orchard's algorithm or the annulus method, and like Orchard's method, uses $\Omega(n^2)$ preprocessing and storage. The AESA algorithm precomputes and stores distances $D(x,y)$ for all $x, y \in S$, and uses the lower bound function D_P defined in Lemma 3.1. When AESA answers a query for point q, every site $x \in S$ is in one of three states:

- *Known*, so that $D(x,q)$ has been computed; the Known sites form a set P;
- *Unknown*, so that only a lower bound $D_P(x,q)$ is available;
- *Rejected*, so that $D_P(x,q)$ is larger than the distance of the closest Known site.

The algorithm starts with all sites x Unknown, with $D_P(x,q) = \infty$, and repeats the following steps until all sites are Rejected or Known:

1. pick the Unknown site x with the smallest $D_P(x,q)$;
2. compute $D(x,q)$, so that x becomes Known;
3. update the smallest distance r known to q;
4. set $P := P \cup \{x\}$, and for all Unknown x', update $D_P(x',q)$; make x' Rejected if $D_P(x',q) > r$.

Based on its definition,

$$D_{P\cup\{x\}}(x',q) = \max\{D_P(x',q), |D(x,q) - D(x,x')|\},$$

so it is easy to maintain its value as sites are added to P.

There will be a need to break ties in the picking step 1, as at the beginning, when all sites have $D_P(x,q) = \infty$. This might be done at random.

While this scheme is simple and answers queries quickly, the quadratic preprocessing and storage limit its applicability. The *Linear* Approximating and Eliminating Search Algorithm, or LAESA [MOV94], reduces these needs by precomputing and storing the distances from all sites to only a subset V of the sites, called *pivots*. The algorithm proceeds as in AESA, but only applies the update step 4 when $x \in V$. The algorithm therefore picks the pivots preferentially in step 1.

The LAESA algorithm works best when the pivots are well separated [MOV94]; similar observations motivate many algorithms, as discussed in Subsection 5.2.2, to use ϵ-nets (defined in the next section) in a way similar to pivot sets.

While AESA makes very thorough use of bounds that are implied by the triangle inequality, perhaps the ultimate in that direction is the work of Shasha and Wang[SW90], whose algorithm considers a matrix of upper and lower bounds on the distances among points in $S\cup\{q\}$, and finds the closure of the bounds implied by the distance evaluations. The set of evaluated distances gives a semimetric, or nonnegatively weighted undirected graph. The triangle inequality gives an upper bound on the distance between two sites by way of the shortest path in the graph, and a lower bound by way of such upper bounds and evaluated distances.

3.2.3 Metric Trees

While the storage needs of the data structures of the last section are considerable, those of *metric trees* are quite modest. A metric tree $T(S)$ can be built as follows: if $|S| = 1$, the tree has one node; otherwise,

1. pick a ball B, with a site as center;
2. recursively construct $T(S \cap B)$ and $T(S \setminus B)$;
3. make these two trees the children of the root;
4. store a description of B at the root, including its center site.

Each node of a metric tree thus corresponds to the intersection of the balls and ball-complements stored at its ancestors in the tree. When answering a query for point q, the tree is traversed and distances to the ball centers of nodes are computed. As the traversal progresses, the minimum of the computed distances gives an upper bound on the nearest-neighbor distance, and thus the radius of a ball B_q centered at q. When a node in the tree is visited, the regions of the two children of the node are considered; if B_q can be proven not to meet the region of a child, based on the ball data in the path to the root, then the child need not be visited. Otherwise, the child is visited. (Here Lemma 3.1(3) gives a means for such a proof, using the current upper bound on the nearest-neighbor distance as r.) The cost of answering a query is proportional to the number of nodes explored.

In the seventies, McNutt(as discussed by Knuth [Knu98]) proposed a data structure similar to metric trees, where children of a node are $T(S \cap B)$ and $T(S \setminus B')$, and B' is a ball with the same center, but slightly smaller radius than B. Thus some sites might be stored in both subtrees. This overlap makes for a data structure that needs more space, but allows some queries to be answered with less backtracking, that is, without needing to explore both children.

Burkhard and Keller [BK73] proposed a multibranch version for discrete-valued metrics. Metric trees, in many variations, were also invented by Omohundro [Omo89], by Uhlmann[Uhl91], and by Yianilos[Yia93], and they have a large literature. For further discussion of them, prior surveys can be consulted [HS03, CNBYM01].

4 Dimensions

While it is easy to construct or encounter metric spaces for which brute-force search is the fastest possible, it is still useful to consider situations in which something faster can be done. Moreover, it may be that the properties of the space that make it desirable to do nearest-neighbor search also make it possible to do the search quickly.

One such property is that of bounded dimension of the metric space, for a wide variety of definitions of the term *dimension*. Such a definition gives a way of assigning a real number to a metric space; all the definitions we consider coincide (assign the same number) for "simple" sets. So the dimension of $\Re^d$, or an open subset of it, is d for any of these definitions, and the dimension of a d-manifold in $\Re^{d'}$ will always be d, regardless of how big d' is. That is, the dimensions are generally "intrinsic," and rely on properties of the given metric space itself, not on any space in which the given space happens to reside.

In physics and statistics, there has long been interest in the use of nearest-neighbor searching for the purpose of estimating the dimension of a space. The correlation integral and correlation dimension were mentioned in Subsection 1.3 above. The k-NN problem is intimately related to the information dimension, as discussed below. The correlation and information dimensions are both instances of the generalized, or Rènyi, dimension spectrum; here there is a numerical parameter v so that $\dim_v Z$ is a measure of dimension, where the information dimension corresponds to $v = 1$, and the correlation dimension corresponds to $v = 2$. The Rènyi spectrum is much-studied in the area of chaotic, multifractal systems, such as turbulence, the web, network traffic [FGW98], and Bayesian belief networks [GH04].

Another dimension value on the Rènyi spectrum can be computed by way of minimum spanning trees, or other extremal geometric graphs, as discussed in Section 5.

This section will survey some of these concepts of dimension, and the relations among them. Only a glimpse will really be given here; as mentioned earlier, for a more thorough understanding the survey by Cutler is helpful [Cut93].

4.1 Dimensions of Metric Spaces

To discuss the many concepts of dimension, the notions of coverings and packings are crucial. These concepts will also appear in algorithms, as discussed in §5.2 below.

Coverings and packings. We will consider bounded metric spaces $Z = (\mathbb{U}, D)$, so that there is some r with $D(x, y) < r$ for all $x, y \in \mathbb{U}$. Given $\epsilon > 0$, an *ϵ-cover* (by balls) of Z is a set $Y \subset \mathbb{U}$ with the property that for every $x \in \mathbb{U}$, there is some $y \in Y$ with $D(x, y) < \epsilon$. Put another way, let

$$B(y, \epsilon) := \{x \in \mathbb{U} \mid D(x, y) < \epsilon\}.$$

Then Y is an ϵ-cover if and only if $\mathbb{U} = \cup_{y \in Y} B(y, \epsilon)$. Put still another way, Y is an ϵ-cover of $\mathbb{U}$ if and only if the Hausdorff distance (cf. §2.1) of $\mathbb{U}$ to Y is less than ϵ.

The *covering number* $\mathcal{C}(\mathbb{U}, \epsilon)$ is the size of the smallest ϵ-covering of $\mathbb{U}$. (Here the dependence of the covering number on the distance function D is implicit.)

For example, if $\mathbb{U}$ is the unit square in the plane, the covering number is $\Theta(1/\epsilon^2)$ as $\epsilon \to 0$, since a disk of radius ϵ can cover only an area proportional to ϵ^2. In general, the covering number of a unit hypercube in $\Re^d$ is $\Theta(1/\epsilon^d)$, for similar reasons.

The quantity $\log_2 \mathcal{C}(\mathbb{U}, \epsilon)$ is called the *ϵ-entropy* or *metric entropy*, a function of ϵ. This measures the number of bits needed to identify an element of the space, up to distortion ϵ. Referring to Subsection 1.3, the elements of the cover could constitute a codebook for an n-quantizer with $n = \mathcal{C}(\mathbb{U}, \epsilon)$. Such a quantizer would need $\log_2 n$ bits to transmit an approximation to a member $x \in \mathbb{U}$, such that the worst-case (not expected) distortion $D(x, f(x))$ is no more than ϵ.

A subset $Y \subset \mathbb{U}$ is an *ϵ-packing* if and only if $D(x, y) > 2\epsilon$ for every $x, y \in Y$. That is, the set of balls $\{B(y, \epsilon) \mid y \in Y\}$ are disjoint.

The *packing number* $\mathcal{P}(\mathbb{U}, \epsilon)$ is the size of the largest ϵ-packing. The packing number is closely related to the covering number, as shown in the following lemma.

Lemma 4.1 *[KT61] For given $\epsilon > 0$ and metric space $(\mathbb{U}, D)$, if $\mathcal{P}(\mathbb{U}, \epsilon)$ and $\mathcal{C}(\mathbb{U}, \epsilon)$ are finite, then*

$$\mathcal{P}(\mathbb{U}, \epsilon) \leq \mathcal{C}(\mathbb{U}, \epsilon) \leq \mathcal{P}(\mathbb{U}, \epsilon/2).$$

Proof: A maximal $(\epsilon/2)$-packing P has the property that no point $s \in \mathbb{U}$ has $D(s, P) > \epsilon$; otherwise such a site could be added to P. That is, a maximal $(\epsilon/2)$-packing P is an ϵ-cover, and so the smallest ϵ-cover can be no larger.

On the other hand, for a given ϵ-cover Y, and ϵ-packing P, every point in P must be in $B(y, \epsilon)$ for some $y \in Y$. However, no two $p, p' \in P$ can be in the same such ball: then $D(p, p') < 2\epsilon$ by the triangle inequality, contradicting the assumption that P is an ϵ-packing. So every ϵ-packing is no larger than any ϵ-cover. ■

Nets and the Greedy Algorithm. The close relation of packing and covering is illuminated by the fundamental concept of *ϵ-nets*. A set $Y \subset \mathbb{U}$ is an *ϵ-net* of $(\mathbb{U}, D)$ if it is both an ϵ-cover and an $(\epsilon/2)$-packing.

An ϵ-net can be constructed by the following greedy algorithm, whose input is $\epsilon \geq 0$ and maximum allowed size k, as well as the metric space $(\mathbb{U}, D)$. The algorithm: pick $s \in \mathbb{U}$ arbitrarily, and set $Y := \{s\}$. Repeat the following: pick an $s \in \mathbb{U}$ that maximizes $D(s, Y) = \min\{D(s, y) \mid y \in Y\}$. If $D(s, Y) < \epsilon$ or $|Y| \geq k$, stop. Otherwise, set $Y := Y \cup \{s\}$, and continue.

The returned Y is an ϵ'-cover for some ϵ', with $\epsilon' < \epsilon$ if k is large enough. Let the ith site added to Y be denoted s_i, and let Y_i denote the set Y before s_i is added. Since the sequence $D(s_i, Y_i)$, for $i = 2 \ldots, |Y|$, is nonincreasing, every member of Y is at least ϵ' from every other member, and so Y is an $(\epsilon'/2)$-packing, and hence an ϵ'-net. Since Y is an $(\epsilon'/2)$-packing, by the Lemma above, any $(\epsilon'/2)$-cover must have at least $|Y|$ members. If this greedy algorithm is run with input $\epsilon = 0$, then the output Y will have size k, and any $(\epsilon'/2)$-cover must have at least k members; that is, the algorithm gives a cover distance ϵ' no more than twice the best possible for k sites: it is an approximation algorithm for the *k-center problem*, of finding the k points whose maximum distance to any point in $\mathbb{U}$ is minimized. Gonzalez [Gon85], and, independently, Hochbaum and Shmoys [HS85], showed that this is the best possible approximation factor for a polynomial-time algorithm on a general metric space, unless $P = NP$.

As mentioned, this algorithm has been used in building nearest-neighbor data structures [Bri95, Woj03, Cla03, HPM05]. It has also been used in computational chemistry[Wil96], where it is one version of the *Bawden-Lajiness algorithm*.

Box Dimension. The *box dimension* $\dim_B(Z)$ of $Z = (\mathbb{U}, D)$ can be defined as follows: it is the d such that the covering number satisfies

$$\mathcal{C}(\mathbb{U}, \epsilon) = 1/\epsilon^{d+o(1)} \tag{2}$$

as $\epsilon \to 0$, if such a d exists. That is, the covering (and packing) numbers depend roughly polynomially on the scale of measurement ϵ, and $\dim_B(Z)$ is the limiting degree of that polynomial. The above condition on d is often expressed as

$$d = \lim_{\epsilon \to 0} \frac{\log \mathcal{C}(\mathbb{U}, \epsilon)}{\log(1/\epsilon)}.$$

The box dimension need not be an integer; sets with nonintegral dimension are often called *fractals*. A set can also have zero measure but be fully dimensioned; for example, space-filling curves in the plane have box dimension two, but area zero. The rational numbers have box dimension one, but length zero. (This last property is generally viewed, mathematically, as "bad" in that for other dimensions, the dimension of a countable union $\cup_i U_i$ is no more than $\sup_i \dim U_i$, so the rationals "should" have dimension zero. This can be patched up, resulting in the *modified box dimension*, which turns out to be equal to the *packing dimension*.)

Another view of the box dimension is that it is the critical value for the box *t-content* $\mathcal{C}(\mathbb{U}, \epsilon)\epsilon^t$. That is, suppose each ball in the cover has volume proportional to at most ϵ^t, as would be true in $\Re^t$. Then the box t-content is a rough overestimate of the volume of $\mathbb{U}$, since it is the sum of volumes of a small collection of sets whose union contains $\mathbb{U}$. Suppose the covering number is $1/\epsilon^{d+o(1)}$; then the t-content is $\epsilon^{t-d+o(1)}$, as $\epsilon \to 0$, which goes to 0 for $t > d$, and ∞ for $t < d$. That is, d is the supremum of the t for which the t-content is infinite, or the infinum of the t for which the t-content is zero.

Hausdorff and Packing Dimensions. A similar relationship holds for some other concepts of dimension: the dimension is the critical value for a t-content function. For example, generalizing on ϵ-covers slightly, suppose we call a collection $\mathcal{E}$ of balls an ϵ-cover when $\mathbb{U} \subset \cup_{B \in \mathcal{E}} B$, and $\operatorname{diam}(B) \leq \epsilon$ for all $B \in E$, where $\operatorname{diam}(B) := \sup_{x,y \in B} D(x, y)$. Now consider the t-content

$$\inf\{\sum_{B \in \mathcal{E}} \operatorname{diam}(B)^t \mid \mathcal{E} \text{ an } \epsilon\text{-cover of } \mathbb{U}\}.$$

This is a *Hausdorff t-measure*, and the corresponding critical value is the *Hausdorff dimension* $\dim_H Z$. (Really, this is the Hausdorff *ball* t-measure, as discussed below.) Note that a cover $\mathcal{E}$ could contain many more balls than $\mathcal{C}(Z, \epsilon)$, but balls smaller than ϵ count less in the sum.

A similar construction as for Hausdorff measure, but with covers $\mathcal{E}$ replaced with packings, and the infinum replaced with a supremum, leads to the *packing* t-measure, and the *packing dimension*[Tri82, Sul84]. (The Hausdorff and packing t-measures are better behaved mathematically than the box t-content, since they are *outer measures*, hence the different names.)

Variations. There are many variations of these constructions; for example, the limit may not always exist, so $\limsup$ or $\liminf$ are used instead, leading to *upper* or *lower* versions of these dimensions, respectively. These are denoted as $\overline{\dim}_H(Z)$ for the upper Hausdorff dimension, and $\underline{\dim}_H(Z)$ for the lower Hausdorff dimension, and similarly for other dimensions. The Hausdorff dimension, and other dimensions, exist if and only if the upper and lower versions are equal.

The Hausdorff measure is usually defined with the covers $\mathcal{E}$ allowed to include arbitrary subsets; this changes the t-measure by some factor, but not the basic dependence on ϵ, and so the dimension is the same. In $\Re^d$, the covering or packing can be done with, for example, boxes for all three versions of dimension, and similarly the measures change by a factor but the dimension remains the same.

Furthermore, it is not necessary to consider boxes of all possible sizes and shapes. For $Z = (\mathbb{U}, D)$ with $\mathbb{U} \subset \Re^d$, and D an ℓ_p distance, an equivalent definition of box dimension can be made using *quadtrees* (also known as hyperoctrees, dyadic cubes, or Besicovitch nets), as follows: put the set $\mathbb{U}$ in a cube; divide the cube into 2^d equal-sized subcubes, divide those into equal sized cubes, and so on, so that at the kth step, there are 2^{kd} equal-sized cubes. Let $\mathcal{B}_{2^{-k}}(\mathbb{U})$ denote the minimum number of such cubes at step k needed to contain $\mathbb{U}$. Then the upper box dimension $\overline{\dim}\, Z$ is the d' such that $\mathcal{B}_{2^{-k}}(\mathbb{U}) = 2^{kd'+o(k)}$ as $k \to \infty$. The occupied cubes in this description correspond to the nodes of a quadtree data structure for $\mathbb{U}$.

Assouad dimension, a.k.a. Uniform Metric Dimension, Doubling Constant. The *Assouad dimension* $\dim_A(Z)$, for space $Z = (\mathbb{U}, D)$ is related to the box dimension, but satisfies a stronger, more uniform condition: it is the value d, if it exists, such that

$$\sup_{x \in \mathbb{U}, r > 0} \mathcal{C}(B(x, r), \epsilon r) = 1/\epsilon^{d+o(1)} \tag{3}$$

as $\epsilon \to 0$ [Lar67, Ass83].

So $\dim_A(Z)$ is at least as large as the box dimension of any ball from the space. This dimension is bounded if and only if $Z = (\mathbb{U}, D)$ is a *doubling* space, meaning that there is a constant C so that that any ball $B(x, 2r)$ is contained in the union of at most 2^C balls of radius r; that is, any $2r$-ball has an r-cover. Sometimes C itself is termed the *doubling dimension*, and 2^C the *doubling constant*. Let $\operatorname{doub}_A(Z)$ denote this version of the doubling dimension. The numbers are related by $\dim_A(Z) \leq \operatorname{doub}_A(Z)$; in fact we can say that the cover size in (3) above is bounded by $O(1/\epsilon^{\operatorname{doub}_A(Z)})$.

From the close relation of packings and coverings, another way to express the doubling condition is that no ball $B(x,r)$ contains an $(r/2)$-packing with more than 2^C members.

Coping with Finiteness. If $\mathbb{U}$ is finite, then $(\mathbb{U}, D)$ has dimension zero for any of these dimensions except $\operatorname{doub}_A Z$, since $\mathcal{C}(Z, \epsilon) \leq n$ for any ϵ. However, the box dimension can be estimated by considering the covering number as a function of ϵ, over some range of ϵ values, and fitting $\log \mathcal{C}(Z, \epsilon)$ to $\log \epsilon$ within that range. Moreover, if $S \subset \mathbb{U}$, then $\dim S \leq \dim \mathbb{U}$ for any of these dimensions, and $\operatorname{doub}_A S \leq \operatorname{doub}_A \mathbb{U}$, that is, a bound on the dimension of $\mathbb{U}$ gives a bound on the dimension of its subsets.

4.2 Dimensions of Metric Measure Spaces

Another category of dimensions applies to metric measure spaces $(\mathbb{U}, d, \mu)$, where a set $\mathbb{U}$ is equipped with both a metric D and a measure μ. (We will assume that $\mu(\mathbb{U}) = 1$, so μ defines a probability distribution, and the measure of a set is the probability mass that it contains.)

Recall that a measure μ is a nonnegatively valued function on subsets of $\mathbb{U}$, with at least the following properties: the empty set ϕ has $\mu(\phi) = 0$; if $A \subset B$, then $\mu(A) \leq \mu(B)$; and $\mu(\cup_i A_i) \leq \sum_i \mu(A_i)$, for $A_1, A_2, \dots \subset \mathbb{U}$.

The metric spaces that are input for nearest-neighbors problems are of course finite, and a given finite set has measure zero for many measures of interest. However, the counting measure μ_c can be used, for which $\mu_c(A) = |A|$, the number of elements of A. A common input of interest is a random sample from μ, with the sites independently generated with distribution μ. Moreover, often we consider such sets of independently identically distributed sites $S = \{x_1, x_2, \dots, x_n\}$ as $n \to \infty$. Since the *empirical measure* $\mu_S(A) := \mu_c(A \cap S)/n$ satisfies

$$\lim_{n\to\infty} \mu_S(A) = \mu(A)$$

with probability one, for any given $A \subset \mathbb{U}$, various properties of μ can be estimated using S. Also, some properties of metric measure spaces can be defined also for finite spaces, for example, the *doubling measure* property.

Doubling Measure. A metric measure space $Z = (\mathbb{U}, D, \mu)$ with a doubling measure[Hei03] is one for which there is a number $\operatorname{doub}_M(Z)$ such that $\mu(B(x, 2r)) \leq \mu(B(x,r)) 2^{\operatorname{doub}_M(Z)}$ for all x and r. Such a space is also called a *growth-restricted metric*[KR02] or *Federer measure* or a *diametrically regular* measure [Fed69]. The definition is sometimes relaxed, so that only balls $B(x,r)$ with $\mu(B(x,r))$ sufficiently large need satisfy the doubling condition.

For such a space there is a smallest number $\dim_D(Z)$ such that

$$\sup_{x\in\mathbb{U}, r>0} \mu(B(x,r))/\mu(B(x,\epsilon r)) = 1/\epsilon^{\dim_D(Z)+o(1)},$$

as $\epsilon \to 0$. (cf. (3).) It is not hard to show that $\operatorname{doub}_A(Z) \leq 4 \operatorname{doub}_M(Z)$[KL04] and that $\dim_A(Z) \leq \dim_D(Z)$.

If the inputs to a nearest-neighbor searching problem are such that $(S \cup \{q\}, D, \mu_c)$ is a doubling measure, then several provably good data structures exist for searching S, as discussed below (§5.2). Note that it is *not* the case that any subset of a growth-restricted space is growth-restricted. However, this relation can hold approximately for *random* subsets.

Rènyi Spectrum and Dimensions. For a metric measure space $(\mathbb{U}, D, \mu)$, and a value $\epsilon \geq 0$, define μ_ϵ by $\mu_\epsilon(x) = \mu(B(x, \epsilon))$, and for a value v, let

$$\|\mu_\epsilon\|_v := \left[\int_{\mathbb{U}} \mu_\epsilon^v d\mu\right]^{1/v} = \left[\int_{\mathbb{U}} \mu(B(y, \epsilon))^v d\mu(y)\right]^{1/v}.$$

That is, μ_ϵ is a "smoothed" version of μ, and $\|\mu_\epsilon\|_v$ is its L_v norm with respect to μ. For integral $v \geq 1$, it is not too hard to see that for random points $X_1 \ldots X_{v+1}$ with distribution μ, $\|\mu_\epsilon\|_v^v$ is the probability that $X_2 \ldots X_{v+1}$ are all within distance ϵ of X_1. So $\|\mu_\epsilon\|_v^v$ is the probability distribution (as a function of ϵ) for the vth nearest-neighbor distance of $v+1$ points.

In particular, $\|\mu_\epsilon\|_1$ is the probability that X_1 and X_2 are within distance ϵ of each other. Since this is a kind of spatial correlation, $\|\mu_\epsilon\|_1$ is also known as the *correlation* integral. For S a random sample with distribution μ, the correlation integral can be estimated by the number of pairs of sites in S at distance less than ϵ, divided by $\binom{n}{2}$. The expectation of this estimate is $\|\mu_\epsilon\|_1$.

The *generalized Rènyi dimension* $\dim_v(\mu)$ is the value d, if it exists, such that

$$\|\mu_\epsilon\|_{v-1} = \epsilon^{d+o(1)},$$

as $\epsilon \to 0$. The Rènyi entropy of order v is $\log\|\mu_\epsilon\|_{v-1}$, and so the Rènyi dimension is the limit of the ratio of the Rènyi entropy to $\log \epsilon$, as $\epsilon \to 0$ [R̀70, Cut93].

The Rènyi dimension can be defined even for $v = 1$, by considering the limiting value of $\|\mu_\epsilon\|_v$ as $v \to 0$. If the limit exists, the result is equal to the *information dimension* of μ, which will be denoted as $\dim_1(\mu)$, and equal to the d such that

$$\int_{\mathbb{U}} \mu_\epsilon(y) \log(\mu_\epsilon(y)) d\mu(y) = \epsilon^{d+o(1)}, \qquad (4)$$

as $\epsilon \to 0$.

The family of values $\dim_v(\mu)$, for $v \in \Re$, is called the *Rènyi spectrum.*

Pointwise Dimension, $f_\mu(\alpha)$. The information dimension is closely related to the *pointwise dimension* $\alpha_\mu(x)$ for $x \in \mathbb{U}$, also known as the *local dimension* or *Hölder exponent.* It is defined as the d, if it exists, such that

$$\mu(B(x, \epsilon)) = \epsilon^{d+o(1)},$$

as $\epsilon \to 0$. That is,

$$\alpha_\mu(x) = \lim_{\epsilon \to 0} \frac{\log \mu(B(x, \epsilon))}{\log \epsilon}, \qquad (5)$$

with a lower version $\underline{\alpha}_\mu(x)$ defined using lim inf instead of lim, and similarly an upper version. The definition of information dimension suggests that $\dim_1(\mu) =$

$E[\alpha_\mu(x))]$, taking the expectation with respect to μ, and indeed for bounded measures the equality holds, under some general conditions [Cut93].

The pointwise dimension is also related to the Hausdorff and packing dimensions of μ: the upper Hausdorff dimension $\overline{\dim}_H(\mu)$ is the infinum of the Hausdorff dimensions of all subsets of $\mathbb{U}$ with measure 1, that is,

$$\overline{\dim}_H(\mu) := \inf\{\dim_H A \mid A \subset \mathbb{U}, \mu(A) = 1\}.$$

(Recalling here our assumption that μ is a probability distribution, so $\mu(\mathbb{U}) = 1$.) The packing dimension of a measure can be defined analogously. The upper Hausdorff dimension can also be expressed in terms of the pointwise dimension:

$$\overline{\dim}_H(\mu) = \inf\{\beta \mid \underline{\alpha}_\mu(x) \le \beta, \text{almost all } x\}$$

So the upper Hausdorff dimension is an upper bound for the pointwise dimension of all points, except for a set of measure zero. (See Edgar [Edg98],3.3.14.) The lower Hausdorff dimension, and upper and lower packing dimensions, also can be expressed as bounds for the pointwise dimension.

Some spaces are *exact-dimensional*, meaning that all points have the same pointwise dimension, except for a set of measure zero. For $(\mathbb{U}, D, \mu)$, let $f_\mu(\alpha)$ denote the Hausdorff dimension of the subset of $\mathbb{U}$ comprising those points with pointwise dimension α. (Sometimes the box dimension is used instead here.) For exact-dimensional spaces, $f_\mu(\alpha)$ is zero for all but one value of α, but other spaces have a more elaborate structure under this *multifractal* analysis. Moreover, the values of $f_\mu(\alpha)$ can be computed from the Rènyi spectrum, using the Legendre transform.

Box-Counting Versions. Just as with the dimensions of metric spaces, for subsets of $\Re^d$ these dimensions are also readily defined in a box-counting, or "quadtree" form: the level i cubes of the quadtree are those cubes of size 2^{-i} that have nonzero measure. Then the value $\sum_{c \text{ level } i} \mu(c)^v$ is within a constant factor of $\|\mu_{2^{-i}}\|_{v-1}^{v-1}$, and yields the same dimension value. (The difference in exponents, that is, v vs. $v-1$, is due to the implicit additional factor of $\mu(x)$ in the integral defining $\|\mu_\epsilon\|_v$.)

In this formulation, the set function for the information dimension becomes

$$\sum_{c \text{ at level } i} \mu(c) \log \mu(c),$$

an estimate of the Shannon information, while the set function for the correlation dimension is

$$\sum_{c \text{ at level } i} \mu(c)^2,$$

and $\dim_0(Z)$ is seen to equal the box dimension of the support of μ.

Again, a given finite or countable metric space (S, D, μ) will have dimension zero, according to the above definitions, but under the assumption that S is i.i.d. with distribution μ, the empirical measure using S gives a way of estimating the dimension of μ. The quadtree cells of interest for estimating the information dimension are those that contain at least one site, and the estimator for the information dimension set function becomes

$$\sum_{c \text{ at level } i} \mu_S(c) \log \mu_S(c).$$

We would like to use the limit, as $\epsilon \to 0$, of the limit of this function as $n \to \infty$, but instead the value for given $\epsilon = 2^{-i}$ is estimated using "sufficiently large" n. (Using such estimates for a range of ϵ scales leads to a dimension estimate via line-fitting, as mentioned for metric space dimension.) It is the complication of relating ϵ and n that has led to consideration of nearest-neighbor-based estimators, where the scale of measurement is set by the sample size n itself.

Relations Among the Dimensions. As noted above, there are some basic inequalities among these notions of dimension. If $\dim_T(Z)$ denotes the topological dimension, we have

$$\dim_T(Z) \leq \dim_H(Z) \leq \dim_0(Z) = \dim_B(Z) \leq \dim_A(Z) \leq \dim_D(Z),$$

when the given values exist. The inequalities can be strict. Also for metric measure space Z, $\dim_q(Z) < \dim_{q'}(Z)$ if $q > q'$, so in particular $\dim_2(Z) \leq \dim_1(Z) \leq \dim_0(Z)$.

Some of these inequalities are clear intuitively. The Assouad dimension is roughly a uniform, homogeneous, worst-case version of the box dimension, so it is not surprising that $\dim_B(Z) \leq \dim_A(Z)$. The box dimension is based on a t-content that is a restricted form of that for the Hausdorff measure, so $\dim_H(Z) \leq \dim_B(Z)$ is intuitively clear. The existence of a doubling measure implies the existence of a doubling constant, implying $\dim_A(Z) \leq \dim_D(Z)$.

4.3 Quantization and Energy Dimensions

For completeness, we note yet two more concepts of dimension for measures. Each is equal to the box dimension in a limiting case. The *quantization* dimension is related to the ability to apply the procedure of vector quantization. Let X denote a random variable in $\mathbb{U}$, with distribution μ. As discussed in §1.3, an *n-quantizer* for μ is a function f on $\mathbb{U}$ that takes a point X to one of at most n points in $\mathbb{U}$. Also $\mathcal{F}_n$ is the set of all such n-quantizer functions, and the nth quantization error for $\mathbb{U}$ of order v is

$$V_{n,v}(\mu) := \inf_{f \in \mathcal{F}_n} ED(X, f(X))^v, \tag{6}$$

the cheapest expected cost of representing X by $f(X)$, where the cost is the vth power of the distance of X to $f(X)$. The *quantization dimension of order* v of $Z = (\mathbb{U}, D, \mu)$ is

$$\dim_Q(Z) := \lim_{n\to\infty} -v \frac{\log n}{\log V_{n,v}(\mu)},$$

that is,

$$V_{n,v}(\mu) = n^{-v/\dim_Q(Z) + o(1)} \tag{7}$$

as $n \to \infty$.

The quantization dimension can also be defined for $v = \infty$, and with "upper" and "lower" versions, and these are equal to the upper and lower box dimension of the support of μ.

Graf and Luschgy [GL00] discuss the quantization dimension in detail.

The *energy* dimension is defined as follows. The *Riesz t-energy* of a measure is

$$I_t(\mu) := \int \int \frac{1}{D(x,y)^t} d\mu(x) d\mu(y),$$

and the energy dimension is $\sup\{t \mid I_t(\mu) < \infty\}$.

This energy is related to the pointwise dimension: for given x, it can be shown [Mat95] that

$$\int \frac{1}{D(x,y)^t} d\mu(y) = t \int_0^\infty r^{-t-1} \mu(B(x,r)) dr.$$

If $\mu(\mathbb{U})$ is bounded, and the upper pointwise dimension is bounded everywhere by some $v > t$, that is, for all $x \in \mathbb{U}$, $\mu(B(x,r)) = O(r^v)$, then $I_t(\mu)$ is bounded.

A discrete version of the energy is, for $S = \{x_1 \dots x_n\}$ with distribution μ,

$$I_t(S) = \frac{1}{n^2} \sum_{i \neq j} \frac{1}{D(x_i, x_j)^t}.$$

Minimizing this energy is a way to produce "well-distributed" points [HS04]. Note that for large t, the small distances will dominate, and a minimizer will be approximately a packing. The results of Hardin and Saff [HS04] imply that the minimum energy $I_t(S)$ for n points in a d-manifold contained in $\Re^{d'}$ is

$$n^{t/d-1+o(1)} \tag{8}$$

as $n \to \infty$, for $t > d$.

5 Dimensions and Nearest-Neighbor Searching

5.1 Dimension Estimation

Dimension measures and nearest-neighbor searching are related in both directions: the computation of some dimensional measures can be done using nearest-neighbor searching, and spaces with bounded dimension can have faster nearest-neighbor searching data structures, both theoretically and empirically.

Nearest-neighbor Searching for Dimension Estimation. In the former direction, we have already seen that the correlation integral can be estimated using a fixed-radius all-sites query. Historically, the quadtree-based view was proposed first, and the distance-based version was proposed as a more accurate empirical estimator [GP83].

For a given set of sample points, the quadtree estimate is easier to compute than the correlation integral, and so Belussi and Faloutsos[BF98] use the quadtree estimator in the context of database *spatial joins.* One kind of spatial join is the set of pairs of sites within distance ϵ of each other, for some given ϵ. That is, its size is exactly the distance-based estimate of the correlation integral, times $\binom{n}{2}$. Belussi and Faloutsos propose the quadtree estimator (together with a line fit) to help estimate the answer size for spatial joins.

Pointwise Dimension. So far, estimators based on quadtrees and fixed-radius queries have been considered; a class of estimators even more directly related to nearest-neighbor search are those based on k-NN search. For example, Cutler and Dawson [CD89] showed that the pointwise dimension (5), related to the information dimension, has the kth nearest-neighbor distance as an estimator:

$$\alpha_\mu(x) = \lim_{n\to\infty} \frac{\log(k/n)}{\log \delta_{k:n}(x)}, \tag{9}$$

with probability 1, where n is the sample size and $\delta_{k:n}(x)$ is the distance of x to its kth nearest neighbor in the sample. In other words,

$$\delta_{1:n}(x) = n^{-1/\alpha_\mu(x)+o(1)} \tag{10}$$

as $n \to \infty$. Similar observations were made by Pettis et al. [KTAD79], Verveer and Duin [VD95], and van de Water and Schram [vdWS88]. A derivation of a similar estimator via maximum likelihood was given by Levina and Bickel [LB05].

Heuristically, (9) can be understood by considering ϵ_k such that the ball $B(x, \epsilon_k)$ has probability mass $\mu(B(x, \epsilon_k)) = k/n$. The expected number of points in the sample falling in $B(x, \epsilon_k)$ is k, and so $\delta_{k:n}(x) \approx \epsilon_k$, and therefore

$$k/n = \mu(B(x, \epsilon_k)) \approx \epsilon_k^{\alpha_\mu(x)} \approx \delta_{k:n}(x)^{\alpha_\mu(x)},$$

using the definition of pointwise dimension, and (9) follows after taking logarithms and dividing. This relation to pointwise dimension suggests that nearest-neighbor distances might be helpful in estimating other related dimensional measures, such as the information, energy, and even Hausdorff dimension.

A paper of Tao et al. [TFP03], related to that of Belussi and Faloutsos [BF98], uses estimates of the pointwise dimension for nearest-neighbor query cost and size estimation; given (S, D), the pointwise dimension for each site in a sample $P \subset S$ is estimated, and then for a given query point, the pointwise dimension estimate for a nearby sample site is used. The pointwise dimension estimate is done with the counting measure, and is called a *local power law.*

A worst-case bound on the pointwise dimension of a graph metric is used by Gao and Zhang [GZ04] in the context of routing. In view of the relation of the Hausdorff and pointwise dimensions, perhaps their bound is a kind of graph Hausdorff dimension.

Extremal Graphs as Dimensional Estimators. In the setting of Euclidean manifolds, Costa and Hero [CH04] propose the use as dimension estimators of the minimum spanning tree, matching, k-NN graph, or other extremal graphs.

Suppose G is such a graph for a set of n sites independently, identically distributed on a d-manifold. For v with $0 < v < d$, let

$$L(G, v) := \sum_{e \text{ an edge of } G} \ell(e)^v,$$

an edge length power sum of G. Costa and Hero use the fact, going back to the celebrated results of Beardwood et al. [BHH59], that

$$L(G, v)/n = n^{-v/d+o(1)}$$

as $n \to \infty$, for the extremal graphs just mentioned, and others. (cf. (7), (8), (10)) Yukich's monograph [Yuk98] surveys results in this setting.) This allows the topological dimension d of a manifold to be estimated as a function of $L(G, v)$ and n, so for example,

$$d = \lim_{n\to\infty} \frac{\log(1/n)}{\log(L(G, 1)/n)}$$

with probability one.

This expression matches (9) for the case of the 1-nearest-neighbor graph in a d-manifold, since $L(G, 1)/n$ is the mean nearest-neighbor distance in the graph, and all points in the manifold have pointwise dimension d. Moreover, algorithms to find the extremal graphs involve nearest-neighbor queries. These estimators also provide their own scaling: there is no ϵ to go to zero, as in the definitions of the dimensions, but rather the scale of measurement $1/n$ is a consequence of the nearest-neighbor relations involved.

Kozma et al.[KLS] have shown a somewhat similar relation between the minimum spanning tree and the upper box dimension: for a bounded metric space $(\mathbb{U}, D)$, let the *minimum spanning tree dimension* be the infinum of the values t such that there is some δ with $L(T(S), t) \leq \delta$ for all $S \subset \mathbb{U}$, where $T(S)$ is the minimum spanning tree of S. That is, $\sup_{S\subset\mathbb{U}} L(T(S), t)$ is a t-content of $\mathbb{U}$, and this minimum spanning tree dimension is its critical value. Kozma et al. show that this dimension is equal to the upper box dimension of $(\mathbb{U}, D)$. They do this by way of a series of packings implicitly constructed in the course of building a minimum spanning tree.

A further, heuristic, relation with box dimension: if we change the greedy ϵ-net algorithm of Subsection 4.1 only "slightly," to take at each step the point whose minimum distance is smallest, instead of largest, then the minimum spanning tree results by connecting each newly added point to its nearest neighbor in the current set.

Kégl[K0́3] proposes using an upper bound on the packing number and a simple form of line-fitting as an estimate of the box dimension, although not using the greedy ϵ-net algorithm discussed in Subsection 4.1.

5.2 Constant Dimension and Nearest-Neighbors Search

5.2.1 Basic Properties, Spread

Some basic properties of metric spaces $Z = (S, D)$ with bounded Assouad dimension, that is, constant doubling dimension, are useful in nearest neighbor searching. Recall that for Z with constant doubling dimension, there is a value $d = \mathrm{doub}_A(Z)$ so that any ball of radius r has an $(r\epsilon)$-cover of size at most $O(1/\epsilon^d)$, as $\epsilon \to 0$. As shown below, this implies a reverse nearest-neighbor condition: every site $s \in S$ is nearest neighbor to $O(2^{O(d)} \log \Delta(S))$ sites, where $\Delta(S)$ is the ratio of the distance between the farthest pair of sites to the distance between the closest pair of sites.

Before that nearest-neighbor condition is shown, a brief digression on the ratio $\Delta(S)$: it is known variously as the *distance ratio*, *aspect ratio*, and *spread*, where the last seems to be most common. Algorithms for nearest-neighbor searching problems that depend on the spread have been known for some time: for example, algorithms for the all-nearest-neighbors or all-k-NN in $\Re^d$ that take $O(n \log \Delta(S))$ time[Cla83, GBT84]. Less anciently, combinatorial properties of point sets in $\Re^d$ with very low spread have also been described[EVW97, Eri02], and bounds have been given for classical clustering algorithms in terms of the spread [HPS05]. Note also that the spread gives a bound to the exponent of Subsection 2.1 related to the “repair” of a distance measure for the triangle inequality.

Although it is not as elegant to include a dependence on the spread in a bound, often that dependence is only on the logarithm of the spread. Making an algorithm more complicated to remove such dependence is unlikely to be worth the trouble in practice.

Here is the reverse nearest-neighbor condition mentioned. It holds not just for nearest neighbors, but in the more general setting of “kth (γ)-nearest” neighbors. A site a is *kth (γ)-nearest to a site* b, with respect to S, if there are at most $k-1$ sites in S whose distance to b is within a factor of γ of the distance of the nearest to b in $S \setminus \{b\}$.

Lemma 5.1 *For a metric space $Z = (S, D)$ with doubling dimension $d = \dim_A(Z)$, and any site $s \in S$, the number of sites $s' \in S$ for which s is k-th (γ)-near in S to s' is $O((2\gamma)^d k \log \Delta(S))$, as $1/\gamma \to 0$.*

Proof: First consider $k = 1$, that is, (γ)-near neighbors, and a ball $B(s, 2r)$ for some $r > 0$. As discussed in §4.1, there is an (r/γ)-cover of $B(s, 2r)$ of size $O((2r/(r/\gamma))^d) = O((2\gamma)^d)$. Therefore any site s' with $r < D(s, s') \le 2r$ has some site in the (r/γ)-cover that is closer to s' than $D(s, s')$ by a factor of at least $r/(r/\gamma) = \gamma$, and the number of sites s' with $r < D(s, s') \le 2r$ that have s (γ)-near is $O((2\gamma)^d)$.

If p is the closest point in S to s, at distance r', then consideration of $r = 2r', 4r', 8r', \ldots$, shows that at most $\log \Delta(S)$ values of r need be considered, each contributing at most $O((2\gamma)^d)$ sites with s (γ)-near in S.

For $k = 2$, all sites of the covers in the above construction are removed from S; this leaves a metric space with fewer sites but the same doubling constant.

New covers can be constructed using the remaining sites, showing that there two sites that are closer to a given site s' than $D(s, s')/\gamma$.

For $k > 2$, build such nets k times. ∎

Doubling Constant Spaces vs. Euclidean. For Euclidean spaces, there is a sharper form of the above bound, as applied to kth nearest neighbors: for $S \subset \Re^d$, a site is kth nearest to at most $k2^{O(d)}$ other sites. It is not clear if this condition alone makes searching in Euclidean spaces much easier.

Another condition satisfied by subsets of Euclidean spaces is that for any site s and query q, if s is nearest neighbor to q, then it is possible to prove this using the Delaunay neighbors of s. (Sites a and b are Delaunay neighbors if there is some ball with a and b on its boundary sphere, and no sites in its interior.) If s is closer to q than any Delaunay neighbor of s is to q, then s is closest to q in S. A site may have many Delaunay neighbors, even in the plane, but for random sites under many probability distributions, a site may have $O(1)$ expected Delaunay neighbors. If the nearest neighbor to the query can be "guessed," then in such cases its status can be proven in constant expected time. Any similar condition for metric spaces seems to include a dependence on the spread.

Note that conversely, if s is not nearest to q, then one of its Delaunay neighbors is closer, suggesting a walk from Delaunay neighbor to Delaunay neighbor toward q; analogs of such an approach are discussed in Subsection 5.2.3.

5.2.2 Divide and Conquer

We next consider applying a *divide-and-conquer* approach to building a data structure and answering a query. Under some conditions, it is possible to split the searching problem into subproblems: the set of sites S is expressed as the union of sets $S_1, S_2, \ldots$, so that for any query point q, the nearest in S to q is in one of the sets S_i, and there is an efficient test to verify that condition. A natural tree data structure $\mathcal{T}(S)$ could be found in this setting by recursively finding $\mathcal{T}(S_i)$ for all S_i, and making each of these a child of the root of the tree. The search algorithm is then: apply this "effective test" to choose S_i, and then recursively search $\mathcal{T}(S_i)$.

Such a data structure is appealing, in that it requires no "back-tracking," that is, it is a tree structure for which the search proceeds from the root, along a single path, to a leaf.

Key properties of such an approach are a bound on $\max_i |S_i|$, and on the total $\sum_i |S_i|$. The former determines the number of levels in the data structure, and the latter is needed to determine the size of the data structure.

One example of such a scheme is an algorithm by Clarkson for the Euclidean case [Cla88].

In the examples below, the divide-and-conquer scheme is based on finding the nearest neighbor to q in a subset $P \subset S$. To motivate such approaches, we return to some basic considerations regarding nearest-neighbor search.

Bounds Using the Nearest in a Subset. The task of nearest-neighbor searching can be viewed as having two parts: finding the nearest neighbor, and proving that all other sites are *not* the nearest neighbor. Moreover, any nearest-neighbor algorithm, at any given time while processing a query, has computed the distance from the query point q to some subset P of the sites. So the algorithm needs to use the distance evaluations from q to P to prove that some sites cannot be the answer to the query. What is the most effective way to do this?

Looking at Lemma 3.1(1), to show that $s \in S \setminus P$ is far from q, given that the distance from some $p \in P$ to q is known, the lower bound of

$$|D(p,q) - D(p,s)|$$

for $D(q,s)$ can be used. It is hard to tell, considering different members p of P, which p would maximize this expression. However, to maximize the difference of the two distances in the expression, one might try to make one distance or the other as small as possible. The $p \in P$ that minimizes $D(p,q)$ is of course the nearest in P to q, while the $p \in P$ that minimizes $D(p,s)$ is the nearest in P to s. So if some $p \in P$ is close to q and far from s, or close to s and far from q, then it gives a proof that s cannot be close to q.

These considerations suggest that one major piece of information for a query point q is the closest site in P. Next we will consider how such information can be used, together with the doubling constant and doubling measure conditions, to suggest some data structures for nearest-neighbor searching that have provable properties. These data structures will be inefficient in their resource bounds, but will illustrate the relations involved.

In each of the three examples below, a subset P of S of size m will be found, together with a ball B_p for each $p \in P$. (B_p is typically, but not always, centered at p.) These will have the property that for query point q, if p is nearest to q in P, then up to some conditions, the nearest neighbor to q in S is contained in B_p. Moreover, some progress will be made by this, either because B_p is small, or there are not too many sites in it. The three cases considered are:

- The space has a doubling constant, P is an ϵ-net, and either q is far enough away from p that p itself is approximately nearest, or B_p contains the nearest to q in S. Each ball B_p is smaller by a constant factor than a ball containing S.
- The space is an (empirical) doubling measure, P is a random subset, and B_p contains the nearest to q in S with very high probability. Moreover, B_p contains $O(n(\log n)/m)$ sites.
- The space has a doubling constant and the queries are *exchangeable* with the sites. Here P is a random subset, and B_p contains the nearest to q in S with controllably high probability $1 - 1/K$, for given K. Moreover, B_p is expected to contain $O((Kn/m)\log^2 \Delta(S))$ sites.

A direct approach for using these constructions is, again, to apply them recursively: to build $\mathcal{T}(S)$, find the subset P and balls B_p for each $p \in P$, then recursively build $\mathcal{T}(S \cap B_p)$ for each $p \in P$. To search for the nearest neighbor in S to query point q, find $p \in P$ closest to q, then recursively search $\mathcal{T}(S \cap B_p)$.

The first approach we consider uses ϵ-nets, which were defined in Subsection 4.1.

Divide and Conquer: Doubling Constant Spaces. Consider a metric space $Z = (\mathbb{U}, D)$ with bounded doubling dimension $d = \mathrm{doub}_A(Z)$, input sites $S \subset \mathbb{U}$, and the problem of building a data structure for approximate nearest-neighbor search. Suppose we scale the distance measure so that the sites fit in a ball of radius one, and suppose the subset P is a δ^2-net, for a parameter $\delta > 0$. That means, in particular, that any site in S is within δ^2 of a site in P. Moreover, the doubling dimension condition means that there is a limit on how many sites are in a δ^2-net, namely, $O(1/\delta^{2d})$. Now suppose a query point q has p as nearest neighbor in P, and a as nearest neighbor in S, and also the nearest neighbor of a in P is $p_a \in P$. Then $D(a, p_a) \leq \delta^2$, and so

$$D(q,p) \leq D(q,p_a) \leq D(q,a) + D(a,p_a) \leq D(q,a) + \delta^2.$$

That is, if $D(q,a) > \delta$, then p is $(1+\delta)$-near to q in S. Otherwise, with $D(q,a) \leq \delta$, we have

$$D(p,a) \leq D(p,q) + D(q,a) \leq 2\delta + \delta^2 \leq 3\delta, \qquad (11)$$

for $\delta < 1$. At the cost of searching the sites of P, we have confined the answer to the query to a ball $B_p := B(p, 3\delta)$, unless p itself is an acceptable answer. Suppose we recursively build a data structure, for each $p \in P$, for $S \cap B(p, 3\delta)$, with $\delta < 1/6$. Then at depth t in such a data structure, the sites are in a ball of radius $1/2^t$.

The building of such a data structure must stop when there is only one site in the current set S. Thus the depth of this data structure, and the cost of searching it for a nearest neighbor, is proportional to $\log \Delta(S)$. The data structure sketched above can answer an approximate nearest neighbor query in time $O(2^{O(d)} \log \Delta(S))$, if δ and so $m = |P| = O(1/\delta^{2d})$ are constants.

Divide and Conquer: Doubling Measure Spaces. Consider now a metric space $(\mathbb{U}, D)$ for which the empirical measure μ_C is doubling for $S \subset \mathbb{U}$ and $q \in \mathbb{U}$. Recall from Subsection 4.2 that such a space has the property that $|S \cap B(x,r)| \geq |S \cap B(x,2r)|/2^C$ for a value C, for all $x \in S \cup \{q\}$ and $r > 0$.

Hereafter, we may abbreviate $|S \cap B(x,r)|$ as $|B(x,r)|$.

Fix a query point q, and let P be a random subset of S, obtained by choosing each site of S independently, with probability m/n, for a parameter m. The expected size of P is m. For $p \in P$, consider ϵ_p chosen so that $|B(p,\epsilon_p)| = Kn(\log n)/m$, where $n := |S|$ and the values of K and m are to be determined. For $p \in P$, suppose $D(q,p) \leq \epsilon_p/2$, and p is nearest to q in P. Then the nearest site to q in S is contained in $B(p, \epsilon_p)$, by Lemma 3.1(4). On the other hand, if $\beta := D(q,p) > \epsilon_p/2$, then

$$|B(q,\beta)| \geq |B(q,3\beta)|/4^C \geq |B(p,\epsilon_p)|/4^C \geq Kn(\log n)/m4^C.$$

where the second inequality follows from $B(p, \epsilon_p) \subset B(q, 3\beta)$, which follows, for $x \in B(p, \epsilon_p)$, from

$$D(q, x) \leq D(q, p) + D(p, x) \leq \beta + \epsilon_p \leq 3\beta.$$

The probability that p is nearest to q in P is no more than the probability that $B(q, \beta)$ has no points of P, which is no more than

$$(1 - m/n)^{Kn(\log n)/m4^C} \leq e^{-K(\log n)/4^C} = 1/n^{K/4^C}.$$

If $K/4^C > 10$, for example, then q will have p nearest in P with probability no more than $1/n^{10}$.

We have the following lemma.

Lemma 5.2 *Suppose* $(\mathbb{U}, D)$ *is a metric space,* $S \subset \mathbb{U}$*, and* $q \in \mathbb{U}$*, such that there is some constant* C *for which*

$$|S \cap B(x, r)| \geq |S \cap B(x, 2r)|/2^C$$

for all $x \in S \cup \{q\}$ *and* r*. Suppose* P *is a random subset of* S*, where* $p \in S$ *is chosen independently for* P *with probability* m/n*. Then with probability at least* $1 - 1/n^{K/4^C}$*, the nearest neighbor to* q *in* S *will be contained in a subset of* S *of size* $Kn(\log n)/m$*.*

If $m := 10K \log n$, that size is $n/10$, so if a data structure is built for each subset recursively, the depth will be $\log n$. Choosing $K := 10(\log n)4^C$ then means that the probability that any step in a search for a given q will fail is no more than about $1/n^9$.

Divide and Conquer: Exchangeable Queries. We have seen that for metric spaces with a doubling constant, it is possible to build a data structure for approximate nearest-neighbor searching, and for doubling measure spaces, it is possible to build a data structure for exact searching. While the schemes given above are crude, the best data structures known for metric spaces under these conditions have a similar behavior: approximate for doubling constant, exact for doubling measure. This is dissatisfying, because the doubling measure condition seems very fragile. The doubling constant condition is more robust, but approximation algorithms have the difficulty that for some metric spaces, and some applications, they may have poor precision: for points uniformly distributed in high dimension, *every* site is not much more distant, relatively speaking, than the nearest site. An approximation algorithm might return any site at all.

A better goal, then, would be a data structure for exact queries that is provably good for doubling constant spaces. Unfortunately, no such data structure is known, so it is worth asking for additional conditions under which provably good data structures can be built.

One such condition is known: when the queries have the same distribution as the sites, that is, they are *exchangeable*. The assumption here is of some random generator of sites and queries, such that the following is true: for a presented query point q, the sets $P \cup \{q\}$ and P' have the same distribution, when P and P' are random subsets of S, and P has one less site than P'. This would hold, for example, when the sites and queries are independently, identically distributed random variables, or if the sites and queries were chosen at random from some large discrete set. Such conditions roughly hold, for example, for vector quantization, where the sites are specifically chosen to be representative of the distribution of the queries.

This condition, together with constant doubling dimension, imply some useful bounds. In particular, a divide-and-conquer construction analogous to those previously given is as follows: pick a random subset $P \subset S$ of size m, then pick a random subset $\hat{P} \subset S$ of size Km, where K and m will be determined. For each $p \in P$, consider the site $q_p \in \hat{P}$ that has p nearest in P, but is farthest away among all such sites in $\hat{P}$. We will show that the ball $B_p := B(q_p, 3D(p, q_p))$ is likely to contain the answer site, for exchangeable query points q with p nearest in P. We will also show that there are not too many sites expected in B_p.

Lemma 5.3 *Under the conditions just above, for $s \in \hat{P}$ with p nearest to s in P and a nearest to s in S, it holds that $D(a, q_p) \leq 3D(p, q_p)$.*

Proof: Since q_p is farther from p than s, $D(s, a) \leq D(s, p) \leq D(q_p, p)$, and so

$$D(q_p, a) \leq D(q_p, p) + D(p, s) + D(s, a) \leq 3D(p, q_p),$$

using the triangle inequality and assumptions. ■

Lemma 5.4 *Under the conditions of the previous lemma, if q is an exchangeable query point with p nearest in P, then with probability $1 - 1/K$, the nearest neighbor to q in S is contained in $B(q_p, 3D(p, q_p))$.*

Proof: If $D(q, p) \leq D(q_p, p)$, the previous lemma shows that the nearest neighbor of q in S is contained in $B(q_p, 3D(p, q_p))$, as desired. So the construction fails only if $D(q, p) > D(q_p, p)$, that is, if q is the point in $P_q := \{q\} \cup \hat{P}$ that is farthest from p, among all points in P_q that have p nearest in P; that is q is the "q_p" of P_q. There are m such points in P_q, and since q is exchangeable, the probability that it is chosen to be one of those m points is $m/(Km+1) < 1/K$. The lemma follows. ■

So the probability is at least $1 - 1/K$ that the nearest neighbor to q is in $B_p := B(q_p, 3D(q_p, p))$, that is, is a (3)-near neighbor of q_p. The next lemma bounds the expected number of such (3)-near neighbors.

Lemma 5.5 *For $P \subset S$ a random subset of size m, $\hat{P} \subset S$ a random subset of size Km, and q an exchangeable query, there are an expected*

$$2^{O(d)} O(Kn/m) \log^2 \Delta(S)$$

sites x such that: there is some q' in $\hat{P}$ with x a (3)-near site with respect to P, and some $p \in P$ that is nearest in P to q and q'.

We really only need to bound the expected number of sites x in such a configuration with $q' = q_p$. It seems easier, however, to bound the number with a weaker condition on q'. The set of such sites x, for a given $p \in P$, contains $B(q_p, 3D(q_p, p)) \cap S$.

Proof: Let P' be any subset of S with $m + 2$ sites. Consider any $x \in S \setminus P'$. The number of sites $\hat{q}' \in P'$ with x (3)-near in P' is at most $2^{O(d)} \log \Delta(S)$, from Lemma 5.1. (Here we apply the lemma to $P' \cup \{x\}$ with $\gamma = 3$, and use the fact that the spread of that set is no more than $\Delta(S)$.) Let $p \in P'$ be the nearest to $\hat{q}'$ in P'. The number of sites $\hat{q} \in P'$ with p as nearest or second nearest is $2^{O(d)} \log \Delta(S)$. (It is possible that $\hat{q}'$ is nearest to $\hat{q}$ in P', and we want to be able to discount that, so second nearest p is considered.)

That is, at most $2^{O(d)} \log^2 \Delta(S)$ configurations of sites $\hat{q}'$, p, and $\hat{q}$ in P' satisfy the conditions:

1. $\hat{q}'$ has x (3)-nearest in P',
2. $\hat{q}'$ has p nearest in P', and
3. $\hat{q}$ has p nearest or second nearest in P'.

Consider now an exchangeable query q, and q' a random member of $\hat{P}$. We have that q and q' are random members of $P \cup \{q, q'\}$. Therefore the probability there are also $p \in P'$ and $x \in S \setminus P'$ such that q, q', p, and x satisfy conditions 1-3, with q in the role of $\hat{q}$ and q' in the role of $\hat{q}'$, is the number of such configurations divided by $(m+1)(m+2)$, namely $(2^{O(d)} \log^2 \Delta(S))/(m+2)(m+1)$. The result follows by multiplying by Km, to account for averaging over the members of $\hat{P}$ by picking random $q' \in \hat{P}$, and also multiplying by $n - m$, to account for all choices of x. ■

Nearly-Linear Data Structures. The above claims were not proposed in the literature exactly as given, since their direct application to divide and conquer does not result in the most efficient data structures. However, the ϵ-net scheme roughly follows the ideas of Krauthgamer and Lee[KL04], while the empirical doubling measure (growth-restricted) scheme follows ideas of an earlier paper by Karger and Ruehl[KR02]. Finally, the "exchangeable queries" model follows a still earlier paper [Cla99].

The problem with applying the direct approach, as described in Subsection 5.2.2, is that the sizes of the subproblems are too big: ideally, the sum of the subproblem sizes $|B_p \cap S|$, over $p \in P$, would be n, but it can be much larger than that. As a result, the resulting data structures, if built with small P, use superlinear storage.

However, for the two approaches described above that employ a random subset, it is possible to use a sample size αn, where α is a fixed fraction; applying this approach recursively to P, the resulting data structure needs storage close to linear in n, although exponential in the doubling dimension. The $M(S, Q)$ algorithm of [Cla99] uses a scheme like this.

Applying this approach to the divide-and-conquer scheme for doubling measure would yield, in the course of construction, a sequence of nested random subsets,

$$S = P_0 \supset P_1 \supset P_2 \supset \ldots \supset P_h, \tag{12}$$

where $|P_i| \approx \alpha^i n$, and $h \approx \log_{1/\alpha} n$. Each P_i is a random subset of P_{i-1}, for $i > 0$, and a random subset of S.

Another way to build nested random subsets is via random permutations: if $p_1, p_2, \ldots, p_n$ is a random permutation of S, then any prefix $\{p_1, p_2, \ldots, p_m\}$ is a random subset. Random permutations, and subsets, can be built one site at a time, picking p_{m+1} as a random element of $S \setminus \{p_1, p_2, \ldots, p_m\}$.

A roughly analogous idea for the ϵ-net scheme would be to use the ϵ-net P to divide and conquer, not the searching problem in S, but instead the searching problem for a larger ϵ-net P' (that is, one with smaller ϵ). There is a nested sequence of subsets as in (12) above, but (assuming S is in a ball of radius $1/2$), each P_i is $(1/2^{h-i})$-net, and $h \approx \Delta(S)$. This is roughly the approach taken by Krauthgamer and Lee[KL04], with additional refinements; they obtain also dynamic data structures that support insertion and deletion of sites.

The ϵ-net divide-and-conquer approach can also use a permutation: the one that arises from the greedy ϵ-net construction procedure described in Subsection 4.1. This permutation is used in [Cla03] and [HPM05].

5.2.3 Traversal Data Structures and Skip Lists

Four ways of generating a nested sequence of subsets of the sites were just described, two for the random approaches, and two for the ϵ-net approaches.

The first way could also be described as follows: each site is given a level number $i \geq 0$, and the level is chosen independently for each site, where the probability that level i is assigned is $1/2^{i+1}$. Starting at the highest level, the search algorithm finds the closest site at level i, and then must consider a small number of sites to determine the closest site at level $i-1$ or higher, repeating until the closest site at level 0 is found.

This description shows that the data structure is similar to a *skip list*[Pug90], which is a way to accelerate searching in a linear list of ordered values; such searching is the one-dimensional version of nearest-neighbor searching.

The skip list approach can be applied to a broader set of methods for nearest-neighbor searching, which could be characterized as graph-searching or *traversal* methods.

Orchard's method (see Subsection 3.2.1) is an example: recall that for Orchard's method, each site s has a corresponding list L_s, sorted by increasing distance from s. The search algorithm maintains a candidate closest site c, and repeats the following steps:

1. walk down L_c, until a site c' closer to q than c is found, or the distance of the list entry to c exceeds $2d(c, q)$;

2. if no such site is found, return c as nearest; otherwise $c := c'$.

In Subsection 5.2, a way of searching in Euclidean spaces was described, using Delaunay neighbors: each site s has a list N_s of its Delaunay neighbors.

For every point q and site s, recall that either some Delaunay neighbor site $s' \in N_s$ is closer to q than s, or else s is the closest site. That is, the same traversal procedure as in Orchard's algorithm can be applied to the Delaunay neighbor lists, in the Euclidean case.

While the Delaunay method can be very expensive, because the total number of Delaunay neighbors can be large, in the Euclidean case there are some traversal approximation algorithms. Given $\epsilon > 0$, Arya and Mount[AM93] found an easily computed list L_s of size independent of the number of sites, such that for any q, if s is closer to q than any member of L_s, then s is $(1+\epsilon)$-near to q in s. This yields a traversal approximation algorithm. In this setting, it is possible to find a list with the same properties as described for Arya and Mount, and whose size is within a provably small factor of the smallest possible for such a list[Cla94].

In the metric-space setting, Navarro [Nav02] proposed a heuristic data structure with a similar but somewhat more complicated searching method. His construction is very similar to one of those of Arya and Mount.

Even when the sizes of the lists L_s are small, it may be that the query time is large, because the path from the starting site to the answer must hop over many sites. However, the skip list technique can be applied to accelerate any traversal method; it was first applied in the nearest-neighbor setting by Arya and Mount. It was expressed above in terms of a nested sequence of random subsets, but it could also be described as follows: assign the sites to levels probabilistically, as above, and for a site s at level i, build the search lists $L_{s,j}$ with respect to P_j, for each $j \leq i$. (As before, P_j is the set of sites at level j or lower.) Starting at some site at the highest level h, perform the traversal procedure using lists $L_{s,h}$, until the nearest site at level h is found, then use the level $h-1$ lists, and so on, until the nearest at level 0 is found. In some cases the same effect can be achieved as follows: for s at level i, concatenate its lists $L_{s,i}, L_{s,i-1}, \ldots, L_{s,0}$, and search this grand concatenated list using the basic traversal method. This might be called a skip-list-accelerated traversal method.

Although it was not derived in the same way, the $M(S,Q)$ data structure of [Cla99] behaves something like such a data structure: the data structure comprises, for each site, a list of sites, the searching method is the traversal above, and the search is provably fast. However, the nested sequence of subsets was generated using a random permutation.

Note that Orchard's algorithm might be accelerated in this way, and in the doubling measure setting, each list L_s for Orchard's method need not include all the sites, by an analysis similar to that for the divide-and-conquer construction.

5.2.4 Voronoi Grouping

The storage requirements of data structures are often their critical limitation, and for nearest-neighbor searching that limitation is particularly acute. Even with storage that is $O(n)$, as in some of the data structures cited above, a dependence in the storage on the doubling constant, or other large constant, makes the data structures unsuitable for many applications.

One way to make a data structure that uses less space is to give up on pure divide and conquer. In the examples above, the condition that p is nearest to the query q in P constitutes a certificate that the nearest site a to q in S is contained in B_p, up to the various additional caveats given. One less "greedy" approach is the following: view each $p \in P$ as the "leader" of a "group" of sites, those sites in S for which p is nearest in P. The set of points that has a given site closest is called its *Voronoi region*, and so this approach might be called Voronoi grouping.

A key value associated with a leader $p \in P$ is its *covering radius* r_p, the distance to the farthest site in its group. Thus each leader $p \in P$ has an associated ball $B(p, r_p)$ that contains all the sites in its group, although not all sites in $B(p, r_p)$ are in the group led by p. However, if at some point the nearest distance to a query q is bounded above by some δ, and $D(q,p) > \delta + r_p$, then none of the sites led by p can be nearest. So the covering radius gives a way of ruling out groups, even if it does not give a way to "rule them in," as in divide and conquer.

An early proposal for Voronoi grouping, called "bisector trees"[KM83] had $|P| = 2$; that is, the sites would be split according to which of two sites was closer, and child subtrees built for each set. To search the tree, the closest site currently known is maintained, and a subtree need not be searched if its sites can be ruled out using the covering radius, as above.

(A very early proposal by Fukunaga and Narendra [FN75] for nearest-neighbor searching uses Voronoi grouping with a large branching factor, that is, $|P|$ is large. Their method does not apply to general metric spaces.)

Another data structure that uses Voronoi grouping is GNAT[Bri95], where $|P|$ is typically a large constant, and one proposed way to choose P is find an ϵ-net of a random sample. There are no proven results about its performance, however.

We turn now to sketching data structures related to those for which provable bounds have been found. Here the approach is generally something like that for divide and conquer: when answering a query, a sequence of larger and larger subsets is considered, and for each subset, the sites that cannot be ruled out as the leaders of the answer a are maintained.

To apply this idea to the ϵ-net approach, consider the nested sequence of ϵ_i-nets P_i described above, with $\epsilon_i := 1/2^{h-i}$, for ϵ-net divide and conquer. (Recall that D has been scaled to have maximum value 1.) For each $p \in P_i$, suppose its leader in P_{i+1}, the closest $p' \in P_{i+1}$, has been found in preprocessing, and stored. A way to answer a query is to find, for $i = h, h-1, \ldots, 0$, the set Q_i containing all sites in P_i that are at a distance from the query point q no more than $D(q, P_i) + 2\epsilon_i$. Let $Q_h := P_h$, the coarsest net. Suppose inductively that $Q_i \subset P_i$ satisfies the distance condition. Then Q_{i-1} can be found from Q_i, because if $p' \in P_i$ is the leader of a site $p \in P_{i-1}$ with

$$D(q,p) \leq D(q, P_{i-1}) + 2\epsilon_{i-1} \leq D(q, P_i) + \epsilon_i,$$

then the distance of q to p' is at most

$$D(q,p') \leq D(q,p) + D(p',p) \leq D(q,P_i) + 2\epsilon_i,$$

and so $p' \in Q_i$.

This is roughly the technique of Beygelzimer et al. [BKL04], who prove resource bounds for it in the doubling measure model.

The following lemma gives a slightly different way of applying this idea.

Lemma 5.6 *For query point q, and $P \subset S$, suppose $p' \in P$ is nearest to q in P, and $a \in S \setminus P$ is nearest to q in S. If $p \in P$ is nearest to a in P, then*

$$D(a,p) \leq D(a,p') \leq 2D(q,p'),$$

and

$$D(q,p) \leq 3D(q,p'),$$

and

$$D(p,p') \leq 4D(q,p').$$

So any leader in P of a is a (3)-near neighbor of q in P.

Proof: We have

$$D(a,p) \leq D(a,p') \leq D(a,q) + D(q,p') \leq 2D(q,p'),$$

using the triangle inequality and the assumptions. So

$$D(q,p) \leq D(q,a) + D(a,p) \leq D(q,p') + 2D(q,p') = 3D(q,p').$$

Finally, $D(p,p') \leq D(q,p') + D(q,p) \leq 4D(q,p')$. ■

This lemma was used by Hildrum et al. [HKMR04] to prove bounds for a randomized data structure along generally similar lines: an increasing nested sequence of random subsets $P_h \subset P_{h-1} \subset \ldots \subset P_0 = S$ is considered, generated with the skip-list technique; each $p \in P_i$ has a link to its leader in P_{i+1}; to answer a query, a subset Q_i is maintained such that Q_i contains the (3)-near neighbors of q in P_i.

The same lemma was used in a previous paper for doubling constant spaces in the exchangeable queries model[Cla99], to obtain a provably efficient data structure, and it also figures in an algorithm for *approximate distance oracles* [BS04].

Recently Har-Peled and Mendel [HPM05] have shown, among many other things, that the greedy permutation can be computed using the Voronoi grouping approach, with a near-linear time bound for constant doubling dimension spaces. Clarkson [Cla03] proposed and implemented a roughly similar algorithm, but without analysis. These algorithms proceed site by site, for $j = 1 \ldots n$, processing a site p_j and making it a leader. Each leader, that is, each site in $P_j := \{p_1, \ldots, p_j\}$, has maintained for it the set of sites in S for which it is nearest in P_j. Such a set for each p_i is maintained in a heap, ordered by distance to p_i, with the largest distance on top. The top of each such heap is itself kept in a heap, so that the site $p \in S \setminus P_j$ for which $D(p, P_j)$ is largest can found, and chosen to be p_{j+1}. When p_{j+1} is processed, a key operation is to find the set of sites that it leads, those for which it is closest in P_{j+1}. In other words, a reverse nearest-neighbor query is done for p_{j+1}. Such queries can be answered quickly, using some information acquired while building P_j.

6 Concluding Remarks

The problem of nearest-neighbor searching and various concepts of metric space dimension have been seen to be related in a variety of interesting ways.

A few obvious questions arise: Does constant doubling dimension allow a data structure for exact queries, without the exchangeability condition? Can efficiency be proven for algorithms under weaker dimensional conditions than doubling measure or doubling constant? Can extremal graphs be used to estimate metric measure space dimensions in a broader setting than Euclidean d-manifolds?

References

[AGE$^+$02] P. K. Agarwal, L. J. Guibas, H. Edelsbrunner, J. Erickson, M. Isard, S. Har-Peled, J. Hershberger, C. Jensen, L. Kavraki, P. Koehl, M. Lin, D. Manocha, D. Metaxas, B. Mirtich, D.. Mount, S. Muthukrishnan, D. Pai, E. Sacks, J. Snoeyink, S. Suri, and O. Wolefson. Algorithmic issues in modeling motion. *ACM Computing Surveys*, 34(4):550–572, 2002.

[AL99] D. K. Agrafiotis and V. S. Lobanov. An efficient implementation of distance-based diversity measures based on k-d trees. *J. Chem. Inf. Comput. Sci.*, 39:51–58, 1999.

[AM93] S. Arya and D. M. Mount. Approximate nearest neighbor queries in fixed dimensions. In *SODA '93: Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 271–280. Society for Industrial and Applied Mathematics, 1993.

[Ass83] P. Assouad. Plongements lipschitziens dans R^n. *Bulletin de la Societe Mathematique de France*, 111:429–448, 1983.

[AT01] R. Anderson and B. Tjaden. The inverse nearest neighbor problem with astrophysical applications. In *SODA '01: Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 767–768. Society for Industrial and Applied Mathematics, 2001.

[Ata85] M. J. Atallah. Some dynamic computational geometry problems. *Comput. Math. Appl.*, 11:1171–1181, 1985.

[BBK01] C. Böhm, S. Berchtold, and D. A. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys*, 33(3):322–373, 2001.

[BF98] A. Belussi and C. Faloutsos. Self-spatial join selectivity estimation using fractal concepts. *ACM Trans. Inf. Syst.*, 16(2):161–201, 1998.

[BGH97] J. Basch, L. J. Guibas, and J. Hershberger. Data structures for mobile data. In *SODA '97: Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 747–756. Society for Industrial and Applied Mathematics, 1997.

[BHH59] J. Beardwood, J. Halton, and J. Hammersley. The shortest path through many points. *Proceedings of the Cambridge Philosophical Society*, 55:299–327, 1959.

[BK73] W. A. Burkhard and R. M. Keller. Some approaches to best-match file searching. *Commun. ACM*, 16:230–236, 1973.

[BKL04] A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. `http://hunch.net/~jl/projects/cover_tree/paper/paper.ps`, 2004.

[Bri95] S. Brin. Near neighbor search in large metric spaces. In *VLDB '95: Proceedings of the Twenty-First International Conference on Very Large Data Bases*, pages 574–584. Morgan Kaufmann Publishers Inc., 1995.

[Bro97] A. Broder. On the resemblance and containment of documents. In *SEQUENCES '97: Proc. Compression and Complexity of Sequences 1997*, page 21. IEEE Computer Society, 1997.

[BS04] S. Baswana and S. Sen. Randomized graph data-structures for approximate shortest paths. In D. P. Mehta and S. Sahni, editors, *Handbook on Data Structures*. Chapman & Hall, London, 2004.

[CD89] C. D. Cutler and D. A. Dawson. Estimation of dimension for spatially-distributed data and related theorems. *Journal of Multivariate Analysis*, 28:115–148, 1989.

[CH04] J. Costa and A. O. Hero. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Trans. Signal Proc.*, 2004.

[Cha02] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC '02: Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 380–388. ACM Press, 2002.

[Cla83] K. L. Clarkson. Fast algorithms for the all nearest neighbors problem. In *FOCS '83: Proceedings of the 24th Symp. on Foundations of Computer Science*, 1983.

[Cla88] K. L. Clarkson. A randomized algorithm for closest-point queries. *SIAM J. Comp.*, pages 830–847, 1988.

[Cla94] K. L. Clarkson. An algorithm for approximate closest-point queries. In *SOCG '94:Proceedings of the Tenth Annual ACM Symposium on Computational Geometry*, pages 160–164, 1994.

[Cla99] K. L. Clarkson. Nearest neighbor queries in metric spaces. *Discrete & Computational Geometry*, 22:63–93, 1999.

[Cla03] K. L. Clarkson. Nearest neighbor searching in metric spaces: Experimental results for $sb(S)$. `http://cm.bell-labs.com/who/clarkson/Msb/readme.html`, 2003.

[CNBYM01] E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, 2001.

[Cut93] C. D. Cutler. A review of the theory and estimation of fractal dimension. In H. Tong, editor, *Dimension Estimation and Models*. World Scientific, 1993.

[DGL96] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, 1996.

[DL97] M. Deza and M. Laurent. *Geometry of Cuts and Metrics*. Springer-Verlag, New York, 1997.

[Edg98] G. A. Edgar. *Integral, Probability, and Fractal Measures*. Springer-Verlag, New York, 1998.

[Eri02] J. Erickson. Dense point sets have sparse Delaunay triangulations: or "–but not too nasty". In *SODA '02: Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 125–134. Society for Industrial and Applied Mathematics, 2002.

[EVW97] H. Edelsbrunner, P. Valtr, and E. Welzl. Cutting dense point sets in half. *Discrete & Computational Geometry*, 17:243–255, 1997.

[Fal90] K. J. Falconer. *Fractal Geometry - Mathematical Foundations and Applications*. J. Wiley & Sons, Hoboken, NJ, 1990.

[Fed69] H. Federer. *Geometric Measure Theory*. Springer-Verlag, New York, 1969.

[FGW98] A. Feldmann, A. C. Gilbert, and W. Willinger. Data networks as cascades: investigating the multifractal nature of internet WAN traffic. In *SIGCOMM '98: Proceedings of the ACM SIGCOMM '98 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 42–55. ACM Press, 1998.

[FJ51] E. Fix and J. L. Hodges Jr. Discriminatory analysis, non-parametric discrimination. Technical Report 4, USAF School of Aviation Medicine, 1951. Project 21-49-004.

[FN75] K. Fukunaga and P. M. Narendra. A branch and bound algorithm for computing k-nearest neighbors. *IEEE Trans. on Computers*, C-24:750–753, 1975.

[FR02] S. Funke and E. A. Ramos. Smooth-surface reconstruction in near-linear time. In *SODA '02: Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 781–790. Society for Industrial and Applied Mathematics, 2002.

[GBT84] H. N. Gabow, J. L. Bentley, and R. E. Tarjan. Scaling and related techniques for geometry problems. In *STOC '84: Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, pages 135–143. ACM Press, 1984.

[GH04] H. Guo and W. Hsu. On multifractal property of the joint probability distributions and its application to Bayesian network inference. In *The 7th Asia-Pacific Conference on Complex Systems*, 2004.

[GL00] S. Graf and H. Luschgy. *Foundations of Quantization for Probability Distributions.* Springer-Verlag, New York, 2000.

[GN93] R.M. Gray and D. L. Neuhoff. Quantization. *IEEE Trans. Inform. Theory*, 44:2325–2383, 1993.

[Gon85] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[GP83] P. Grassberger and I. Procaccia. Measuring the strangeness of strange attractors. *Physica*, D9:189–208, 1983.

[GZ04] J. Gao and L. Zhang. Tradeoffs between stretch factor and load balancing ratio in routing on growth restricted graphs. In *PODC '04: Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing*, pages 189–196. ACM Press, 2004.

[Hei03] J. Heinonen. Geometric embeddings of metric spaces. Technical report, University of Jyväskylä, Finland, 2003. `http://www.math.jyu.fi/tutkimus/ber.html`.

[HKMR04] K. Hildrum, J. Kubiatowicz, S. Ma, and S. Rao. A note on the nearest neighbor in growth-restricted metrics. In *SODA '04: Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 560–561. Society for Industrial and Applied Mathematics, 2004.

[Hor03] K. Hormann. From scattered samples to smooth surfaces. In *Proc. of Geometric Modeling and Computer Graphics*, 2003.

[HPM05] S. Har-Peled and M. Mendel. Fast construction of nets in low dimensional metrics, and their applications. In *SOCG '05: Proceedings of the Twenty-First Annual ACM Symposium on Computational Geometry*, 2005.

[HPS05] S. Har-Peled and B. Sadri. How fast is the k-means method? *Algorithmica*, 41(3):185–202, 2005.

[HS85] D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10:180–184, 1985.

[HS00] G. R. Hjaltason and H. Samet. Contractive embedding methods for similarity searching in metric spaces. Technical Report 4102, University of Maryland, 2000.

[HS03] G. R. Hjaltason and H. Samet. Index-driven similarity search in metric spaces. *ACM Trans. Database Syst.*, 28(4):517–580, 2003.

[HS04] D. P. Hardin and E. B. Saff. Discretizing manifolds via minimum energy points. *Notices of the AMS*, 51(10):1186–1194, 2004.

[IM04] P. Indyk and J. Matoušek. Low distortion embeddings of finite metric spaces. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*. CRC Press, second edition, 2004.

[Ind04] P. Indyk. Nearest neighbors in high dimensional spaces. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*. CRC Press, second edition, 2004.

[Jac01] P. Jaccard. Etude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bull. Soc. Vaudoise Sci. Nat*, 37:547–579, 1901.

[K0́3] B. Kégl. Intrinsic dimension estimation using packing numbers. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 681–688. MIT Press, Cambridge, MA, 2003.

[Kah91] S. Kahan. A model for data in motion. In *STOC '91: Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, pages 265–277. ACM Press, 1991.

[KL04] R. Krauthgamer and J. R. Lee. Navigating nets: simple algorithms for proximity search. In *SODA '04: Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 798–807. Society for Industrial and Applied Mathematics, 2004.

[KLS] G. Kozma, Z. Lotker, and G. Stupp. The minimal spanning tree and the upper box dimension. arXiv:math.CO/0311481.

[KM83] I. Kalantari and G. McDonald. A data structure and an algorithm for the nearest point problem. *IEEE Transactions on Software Engineering*, 9(5), 1983.

[KM00] F. Korn and S. Muthukrishnan. Influence sets based on reverse nearest neighbor queries. In *SIGMOD 2000: Proceedings of the ACM International Conference on Management of Data*, pages 201–212. ACM Press, 2000.

[Knu98] Donald Ervin Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley, Boston, 1998.

[KR02] D. R. Karger and M. Ruhl. Finding nearest neighbors in growth-restricted metrics. In *STOC '02: Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 741–750. ACM Press, 2002.

[KT61] A. Kolmogorov and V. Tikhomirov. ϵ-entropy and ϵ-capacity of sets of functions. *Translations of the AMS*, 17:277–364, 1961.

[KTAD79] K.W.Pettis, T.A.Bailey, A.K.Jain, and R.C. Dubes. An intrinsic dimensionality estimator from near-neighbor information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:25–37, 1979.

[Lar67] D.G. Larman. A new theory of dimension. *Proc. London Math. Soc. (3rd Series)*, pages 178–192, 1967.

[LB01] E. Levina and P.J. Bickel. The earth mover's distance is the Mallows distance: Some insights from statistics. In *Proc. ICCV '01*, pages 251–256, 2001.

[LB05] E. Levina and P. J. Bickel. Maximum likelihood estimation of intrinsic dimension. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 777–784. MIT Press, Cambridge, MA, 2005.

[LBG80] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28:84–95, 1980.

[LMG04] T. Liu, A. W. Moore, and A. Gray. Efficient exact k-NN and nonparametric classification in high dimensions. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[Mat95] P. Mattila. *Geometry of Sets and Measures in Euclidean Spaces*. Cambridge University Press, Cambridge, 1995.

[MNSW00] D. M. Mount, N. Netanyahu, R. Silverman, and A. Y. Wu. Chromatic nearest neighbor searching: A query sensitive approach. *Computational Geometry: Theory and Applications*, 17:97–119, 2000.

[MOV94] M. L. Micó, J. Oncina, and E. Vidal. A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing time and memory requirements. *Pattern Recognition Letters*, 15(1):9–17, 1994.

[MS58] F. Marczewski and H. Steinhaus. On a certain distance of sets and the corresponding distance of functions. *Colloquium Mathematicum*, 6:319–327, 1958.

[Nav02] G. Navarro. Searching in metric spaces by spatial approximation. *VLDB Journal*, 11(1):28–46, 2002.

[Omo89] S. Omohundro. Five balltree construction algorithms. Technical report, International Computer Science Institute, 1989.

[Orc91] M. T. Orchard. A fast nearest-neighbor search algorithm. In *ICASSP '91: 1991 International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 2297–3000, 1991.

[Pes97] Y. B. Pesin. *Dimension Theory in Dynamical Systems. Contemporary views and applications.* University of Chicago Press, 1997.

[PM05] A. N. Papadopoulos and Y. Manolopoulos. *Nearest Neighbor Search: A Database Perspective*. Springer-Verlag, New York, 2005.

[Pug90] W. Pugh. Skip lists: A probabilistic alternative to balanced trees. *Comm. ACM*, 33(6):668–676, 1990.

[R̀70] A. Rènyi. *Probability Theory.* North-Holland, New York, 1970.

[RT60] D. Rogers and T. Tanimoto. A computer program for classifying plants. *Science*, 132(3434):1115–1118, 1960.

[SFcT03] A. Singh, H. Ferhatosmanoglu, and A. Ş. Tosun. High dimensional reverse nearest neighbor queries. In *CIKM '03: Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pages 91–98. ACM Press, 2003.

[Smi00] M. Smid. Closest-point problems in computational geometry. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry.* North-Holland, 2000.

[Spa80] H. Spaeth. *Cluster Analysis Algorithms for Data Reduction and Classification of Objects.* J. Wiley & Sons, Hoboken, NJ, 1980.

[SR03] L. K. Saul and S. T. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *J. Mach. Learn. Res.*, 4:119–155, 2003.

[Sul84] D. Sullivan. Entropy, Hausdorff measures old and new, and limit sets of geometrically finite Kleinian groups. *Acta Mathematica*, 153:259–277, 1984.

[SW90] D. Shasha and T.-L. Wang. New techniques for best-match retrieval. *ACM Trans. Inf. Syst.*, 8(2):140–158, 1990.

[TFP03] Y. Tao, C. Faloutsos, and D. Papadias. The power-method: a comprehensive estimation technique for multi-dimensional queries. In *CIKM '03: Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pages 83–90. ACM Press, 2003.

[Tri82] C. Tricot. Two definitions of fractional dimension. *Math. Proceedings of the Cambridge Philosophical Society*, 91:57–74, 1982.

[Uhl91] J. K. Uhlmann. Satisfying general proximity/similarity queries with metric trees. *Inform. Proc. Letters*, 40:175–179, 1991.

[VD95] P. Verveer and R. Duin. An evaluation of intrinsic dimensionality estimators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:81–86, 1995.

[vdWS88] W. van de Water and P. Schram. Generalized dimensions from near-neighbor information. *Phys. Rev. A*, 37:3118–3125, 1988.

[Vid86] E. Vidal. An algorithm for finding nearest neighbours in (approximately) constant average time. *Pattern Recognition Letters*, 4:145–157, 1986.

[Vid94] E. Vidal. New formulation and improvements of the nearest neighbours in approximating and eliminating search algorithm (AESA). *Pattern Recognition Letters*, 15(1), 1994.

[Wil96] P. Willett. Molecular diversity techniques for chemical databases. *Information Research*, 2(3), 1996.

[Woj03] A. Wojna. Center-based indexing in vector and metric spaces. *Fundam. Inf.*, 56(3):285–310, 2003.

[XA03] H. Xu and D. K. Agrafiotis. Nearest neighbor search in general metric spaces using a tree data structure with a simple heuristic. *J. Chem. Inf. Comput. Sci.*, 43:1933–1941, 2003.

[Yia93] P. N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *SODA '93: Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 311–321. Society for Industrial and Applied Mathematics, 1993.

[Yuk98] J. E. Yukich. *Probability theory of classical Euclidean optimization*, volume 1675 of *Lecture Notes in Mathematics*. Springer-Verlag, New York, 1998.

[ZJL02] K. Zatloukal, M. H. Johnson, and R. Ladner. Nearest neighbor search for data compression. In M. Goldwasser, D. Johnson, and C. McGeoch, editors, *Data Structures, Nearest Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges*. AMS, 2002.