Closest Pair and the Post Office Problem for Stochastic Points

Pegah Kamousi[1], Timothy M. Chan[2], and Subhash Suri[1]

[1] Computer Science, UC Santa Barbara, CA 93106
[2] Computer Science, University of Waterloo, Ontario N2L3G1

Abstract. Given a (master) set M of n points in d-dimensional Euclidean space, consider drawing a random subset that includes each point $m_i \in M$ with an independent probability p_i. How difficult is it to compute elementary statistics about the closest pair of points in such a subset? For instance, what is the *probability* that the distance between the closest pair of points in the random subset is no more than ℓ, for a given value ℓ? Or, can we preprocess the master set M such that given a query point q, we can efficiently estimate the *expected* distance from q to its nearest neighbor in the random subset? We obtain hardness results and approximation algorithms for stochastic problems of this kind.

1 Introduction

Many years ago, Knuth [12] posed the now classic *post-office* problem, namely, given a set of points in the plane, find the one closest to a query point q. The problem, which is fundamental and arises as a basic building block of numerous computational geometry algorithms and data structures [7], is reasonably well-understood in small dimensions. In this paper, we consider a *stochastic* version of the problem in which each post office may be *closed* with certain probability. In other words, a given set of points M in d dimensions includes the locations of all the post offices but on a typical day each post office $m_i \in M$ is only open with an independent probability p_i. The set of points M together with their probabilities form a probability distribution where each point m_i is included in a random subset of points with probability p_i. Thus, given a query points q, we now must talk about the *expected* distance from q to its closest neighbor in M. Similarly, instead of simply computing the closest pair of points in a set, we must ask: how *likely* is it that the closest pair of points are no more than ℓ apart?

In this paper, we study the complexity of such elementary proximity problems and establish both upper and lower bounds. In particular, we have a finite set of points M in a d-dimensional Euclidean space, which constitutes our *master* set of points, and hence the mnemonic M. Each member m_i of M has probability p_i of being present and probability $1 - p_i$ of being absent. (Following the post-office analogy, the ith post office is open with probability p_i and closed otherwise.) These probabilities are independent, but otherwise arbitrary, and lead to a sample space of 2^n possible subsets, where a sample subset includes the ith point with independent probability p_i. We achieve the following results.

1. It is NP-Hard to compute the probability that the closest pair of points have distance at most a value ℓ, even for dimension 2 under the L_∞ norm.
2. In the *linearly-separable* and *bichromatic* planar case, the closest pair probability can be computed in polynomial time under the L_∞ norm.
3. Without the linear separability, even the bichromatic case is NP-Hard under the L_2 or L_∞ norm.
4. Even with linear separability and L_∞ norm, the bichromatic case becomes NP-Hard in dimension $d \geq 3$.
5. We give a linear-space data structure with $O(\log n)$ query time to compute the expected distance of a given query point to its $(1+\varepsilon)$-approximate nearest neighbor when the dimension d is a constant.

Related Work

A number of researchers have recently begun to explore geometric computing over *probabilistic* data [1, 2, 14, 19]. These studies are fundamentally different from the classical geometric probability that deals with properties of random point sets drawn from some infinite sets, such as points in unit square [11]. Instead, the recent work in computational geometry is concerned with worst-case set of objects and worst-case probabilities or behavior. In particular, the recent work of Agarwal [1, 2] deals with the database problem of skyline computation using a multiple-universe model. The work of van Kreveld and Löffler [14, 19] deals with objects whose locations are randomly distributed in some *uncertainty* regions. Unlike these results, our focus is not on the locational uncertainty but rather on the discrete probabilities which which each point may appear.

2 The Stochastic Closest Pair Problem

We begin with the complexity of the stochastic closest pair problem, which asks for the probability that the closest pair has distance at most a given bound ℓ. We will show that this basic problem is intractable, via reduction from the problem of *counting vertex covers* in planar unit disk graphs (UDGs). In order to show that even the *bichromatic* closest pair problem is hard, we also prove that a corresponding vertex cover counting problem is hard for a bichromatic version of the unit disk graphs.

2.1 Counting Vertex Covers in Unit Disk Graphs

The problem of counting the vertex covers in a graph [13, 16, 18] is the following. Given a graph $G = (V, E)$, how many subsets $S \subseteq V$ constitute a vertex cover, where S is a vertex cover of G if for each $uv \in E$, either $u \in S$ or $v \in S$. (We note that we are counting vertex covers and *not* minimum vertex covers.) This problem is known to be #P-hard even for planar bipartite graphs with maximum degree 4 [16]. Our reduction will use *unit disk graphs* (UDG), which are the

intersection graphs of n equal-sized circles in the plane: each node corresponds to a circle, and there is an edge between two nodes if the corresponding circles intersect. We will first prove that the minimum vertex cover problem is hard for planar unit disk graphs of maximum degree 3 (3-planar UDG), using which we then prove that counting the vertex covers is also hard for 3-planar UDGs. We will then extend this result to a special class of planar unit disk graphs, which we call the *rectilinear unit disk graphs* (to be defined later). The first step in the proof is the following well-known lemma of Valiant [17].

Lemma 1. *A planar graph $G = (V, E)$ with maximum degree 4 can be embedded in the plane using $O(|V|)$ area in such a way that its nodes are at integer coordinates and its edges are drawn so that they are made up of line segments of the form $x = i$ or $y = j$, for integers i and j.*

Throughout this section, we assume that the disks defining unit disk graphs have radius 1. We use the notation $d(P, Q)$ for the L_2 distance between two sets P and Q. (When using the L_∞ or L_1 norms, we will use $d_\infty(P, Q)$ and $d_1(P, Q)$.)

Lemma 2. *The minimum vertex cover problem is* NP-hard *for planar unit disk graphs of maximum degree* 3*.*

Proof. The reduction is from minimum vertex cover for planar graphs with maximum degree three [8]. Let $G = (V, E)$ be an instance of such graphs. We embed G in the plane according to Lemma 1. Let $G_r = (V_r, E_r)$ be the graph obtained from this embedding by replacing each edge $uv \in E$ by a path consisting of even number $2k_{uv}$ of intermediate nodes, each at distance ≤ 1 from the previous one, in such a way that the resulting graph is a unit disk graph. The value k_{uv} depends on the L_1 distance between u and v in G_r. (To avoid unwanted edges between the intermediate nodes on different edges, some scaling may be required, but it is easy to see that this is always possible.)

It is not hard to see that G has a vertex cover of size $\leq k$ if and only if G_r has a vertex cover of size $\leq k + \sum_{uv \in E} k_{uv}$. But G_r is a 3-planar UDG, which shows that the 3-planar UDG vertex cover is NP-hard. □

The graph G_r above is a unit disk graph with maximum degree 3, which can be embedded in the plane such that the length of each edge is $\geq 2/3$, and the edges lie on the integer grid lines. This is possible by placing the $2k_{uv}$ intermediate nodes on integer grid points if $d_1(u, v)$ is an odd number, or uniformly distributing $d_1(u, v)$ intermediate points on uv in case $d_1(u, v)$ is even (in the extreme case where $d_1(u, v) = 2$, we place 2 nodes on uv and obtain 3 edges of length 2/3). Let us call a unit disk graph that admits such embedding a *rectilinear unit disk graph*. We have the following corollary of Lemma 2.

Corollary 1. *The minimum vertex cover problem is* NP-hard *for rectilinear unit disk graphs.*

Theorem 1. *It is* NP-hard *to count the vertex covers in a rectilinear unit disk graph. Moreover, the number of vertex covers cannot be approximated to any multiplicative factor in polynomial time assuming* P $\neq$ NP*.*

Proof. We will prove the inapproximability, which shows the hardness as well. Let $G = (V, E)$ be a rectilinear UDG. Suppose we have an α-approximation algorithm for counting the vertex covers in G, i.e., if $c(G)$ is the number of vertex covers, the algorithm outputs a value $\tilde{c}$ such that $(1/\alpha)c(G) \leq \tilde{c} \leq \alpha c(G)$.

Let G_p be the stochastic graph obtained from G by assigning the probability $p = 1/(2^n\alpha^2)$ of being present to each of the nodes in G. Since this probability is the same for all the nodes, an α-approximation algorithm for counting the vertex covers in G readily provides an α-approximation algorithm for computing the probability that a random subset of nodes in G_p is a vertex cover. Let $\Pr(G_p)$ denote this probability, and $\tilde{r}$ be an α-approximation to $\Pr(G_p)$.

The key observation is that $\tilde{r} \geq p^k$ if and only if G has a vertex cover of size k or less. To see this, suppose G has a vertex cover C of size k or less. Then the probability that a random subset of nodes of G_p is a vertex cover is at least p^k, i.e., the probability that all the nodes in C are present. In this case, $\tilde{r} \geq p^k/\alpha$. Otherwise, at least $k+1$ nodes must be present to constitute a vertex cover, which happens with probability at most $2^{|V|}p^{k+1} < p^k/\alpha^2$. In this case $\tilde{r} < p^k/\alpha$. Corollary 1, however, shows that the minimum vertex cover problem is hard for G, and therefore $\Pr(G_p)$ cannot be approximated to any factor α in polynomial time assuming P $\neq$ NP. This completes the proof. □

2.2 Bichromatic Unit Disk Graphs

We introduce the class of *bichromatic unit disk graphs* as the graphs defined over a set of points in the plane, each colored as blue or red, with an edge between a red and a blue pair if and only if their distance is ≤ 1. (We do not put edges between nodes of the same color regardless of the distance between them.) We will show that counting the vertex covers is NP-hard for bichromatic UDGs. We will need the following lemma from [15]. Remember that a Vandermonde matrix M is in the form $\mathbf{M_{ij}} = (\mu_i^j, 0 \leq i, j \leq n)$ (or its transpose) for a given sequence of numbers $\mu_0, \ldots, \mu_n$. (See [9, §5.1].)

Lemma 3. *Suppose we have v_i and b_i, $i = 0, \ldots, n$, related by the equation $v_i = \sum_{j=0}^{n} a_{ij}b_i, j = 0, \ldots, n$. Further suppose that the matrix of the coefficients (a_{ij}) is Vandermonde, with parameters $\mu_0, \ldots, \mu_n$ which are distinct. Then given the values $v_0, \ldots, v_n$, we can obtain the values $b_0, \ldots, b_n$ in time polynomial in n.*

Now consider the gadget $\mathcal{H}$ in Fig. 1 (a), which consists of l paths between u and v, for a given l. Let $G = (V, E)$ be an instance of a rectilinear UDG. Let $G' = (V', E')$ be the graph obtained from G by replacing each edge $uv \in E$ with the graph $\mathcal{H}$. We color u, v and the b_i's red, and the remaining nodes blue.

Lemma 4. *The graph G' is a bichromatic unit disk graph.*

Proof. First we embed the original graph G in the plane with edges of length $\geq 2/3$ lying on integer grid lines (this is possible by definition of G). We will then scale the grid by a factor of 3.5, so that for all $uv \in E$, $2 < d_2(u, v) \leq 3.5$. Next we embed the graph $\mathcal{H}$ on each edge $uv \in E$ such that $d(a_i, b_i) = d(b_i, c_i) = 1$,

while $d(u, a_i) = d(v, c_i) \leq 1$, for $i = 1, \ldots, l$. This is always possible since $2 < d_2(u, v) \leq 3.5$.

It is easy to see that the distance between two nodes of different colors from two different rows is always greater than 1, and therefore there won't be any edges between two different rows. Moreover, there should not be any connections between two different gadgets placed on two edges. Given that we can scale the initial embedding as desired, the only case to worry about is for two orthogonal edges. Considering Fig. 1 (b), it is easy to see that each two nodes of different colors in two different gadgets are at distance > 1. This completes the proof. □

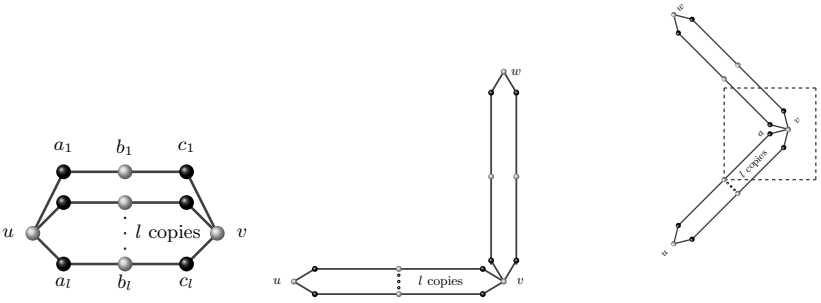

Fig. 1. (a) The gadget $\mathcal{H}$ (b) Two orthogonal gadgets (c) Two rotated gadgets.

Finally we arrive at the following theorem.

Theorem 2. *It is* NP-hard *to count the number of vertex covers in a bichromatic unit disk graph even if the distances are measured in the L_∞ metric.*

Proof. The reduction is from counting the vertex covers in rectilinear unit disk graphs. It is inspired by a technique in [15]. Let $G = (V, E)$ be an instance of a rectilinear UDG. Let $G'(l) = (V'(l), E'(l))$ be the graph obtained from G by replacing each edge $uv \in E$ with the gadget $\mathcal{H}$ in Fig. 1 (a). By Lemma 4, $G'(l)$ is a bipartite unit disk graph (note that when $l = 0$, the graph $G'(l)$ has no edges at all). Let $N = \binom{m+2}{2}$, where $m = |E|$. We will show that by counting the vertex covers in $G'(l), l = 0, \ldots, N-1$, we can effectively count the vertex covers in G.

In the graph $\mathcal{H}$, the number of vertex covers containing neither u nor v is 2^l since any vertex cover necessarily contains all the a_i's and c_i's ($i = 1, \ldots, l$), while b_i's may or may not be included. Moreover, the number of vertex covers containing one of u and v is 3^l, and the number of vertex covers containing both u and v is 5^l. (To see this, notice that when u is included and v is not, all the c_i's are necessarily included, and to cover the remaining edges on the i-th path, we need either a_i, b_i, or both. On the other hand, when both u and v are included, we would have 5 choices to cover the remaining two edges on each path.)

Let A_{ijk} be the number of subsets of nodes of the original graph, G, by which exactly i edges from G have none of their endpoints covered, j edges have exactly

one endpoint, and k edges have both of their endpoint covered. Then if $c'(l)$ is the number of vertex covers of $G'(l)$, we have

$$c'(l) = \sum_{i+j+k=m,\ i,j,k\geq 0} A_{ijk}(2^l)^i(3^l)^j(5^l)^k = \sum_{i+j+k=m,\ i,j,k\geq 0} A_{ijk}(2^i3^j5^k)^l. \quad (1)$$

The value we are interested in is $\sum_{j+k=m} A_{0jk}$, which is the number of vertex covers of G. Let $\mathbf{B} = (b_{ql})$ be the $N \times N$ matrix defined as

$$b_{ql} = (2^{i_q}3^{j_q}5^{k_q})^l \quad q = 1, \ldots, N, \quad l = 0, \ldots, N-1,$$

where (i_q, j_q, k_q) are all the triples summing up to m. Then $\mathbf{B}$ is a Vandermonde matrix and the values μ_q are distinct for $q = 1, \cdots, N$ since $\mu_q = 2^{i_q}, 3^{j_q}, 5^{k_q} = 2^{i_r}, 3^{j_r}, 5^{k_r} = \mu_r$ if and only if $i_q = i_r, j_q = j_r$ and $k_q = k_r$. By Lemma 3, we can solve (1) to obtain $\{A_{ijk}\}_{i+j+k=m,\ i,j,k\geq 0}$, and therefore also $\sum_{j+k=m} A_{0jk}$. We conclude that by counting the vertex covers in bichromatic unit disk graphs, we can count the vertex covers in rectilinear unit disk graphs. But Theorem 1 shows that this problem is hard.

Finally, we will show that the problem remains hard when we consider the distances under the L_∞ norm. Consider Fig. 1 (c), which illustrates a simple rotation of $G'(l)$ by the angle $\pi/2$. Consider the L_∞ ball around the point a. This ball does not include any point of a different color which is not connected to a in $G'(l)$. It is easy to see that the same applies to all other points in the graph, the connectivity structure of the graph remains unchanged, and so does the number of vertex covers. This completes the proof. □

2.3 Complexity of the Stochastic Closest Pair Problem

We are now ready to prove the following result.

Theorem 3. *Given a set M of points in the plane, where each point $m_i \in M$ is present with probability p_i, it is* NP-hard *to compute the probability that the L_2 or L_∞ distance between the closest pair is $\leq \ell$ for a given value ℓ.*

Proof. In a unit disk graph $G = (V, E)$ defined over the full set M of points, a random subset S of nodes is a vertex cover if and only if in the complement of that subset, no two nodes are at distance ≤ 1. (In other words, all the edges are covered by S.) Therefore, computing the probability that a random subset of nodes is a vertex cover in G amounts to computing the probability that the closest pair of present points in a random subset S are at distance > 1. But as discussed in Theorem 1, counting the vertex covers in a unit disk graph is NP-hard. The fact that Theorem 1 applies to rectilinear unit disk graphs readily proves that the problem remains hard for the L_∞ metric.

The next theorem, which considers the bichromatic version of this problem, is based on the same argument as above along with Theorem 2.

Theorem 4. *Given a set R of red and a set B of blue points in the plane, where each point $r_i \in R$ is only present with an independent, rational probability p_i, and each point $b_i \in B$ is present with probability q_i, it is* NP-hard *to compute the probability that the closest L_2 or L_∞ distance between a bichromatic pair of present points is less that a given value ℓ.*

3 Linearly Separable Point Sets under the L_∞ Norm

In the following, we show that when the red points are linearly separable from the blue points by a vertical or a horizontal line, the stochastic bichromatic closest pair problem under L_∞ distances can be solved in polynomial time. We only describe the algorithm for the points separable by a vertical line, noting that all the arguments can be adapted to the case of a horizontal line.

Let $U = \{u_1, \ldots, u_n\}$ be the set of red points on one side, and $V = \{v_1, \ldots, v_m\}$ the set of blue points on the other side of a line. Each point $u_i \in U$ is present with probability p_i, while each point $v_j \in V$ is present with probability q_j. We sort the red points by x-coordinate (from right to left), and the blue points by y-coordinate (top-down). Let $R[i, j, k]$ be the region defined by the intersection of the halfplanes $x \leq 0, x \geq x(u_i) - 1, y \geq y(v_j)$ and $y \leq y(v_k)$, for $y(v_j) < y(v_k)$ (Fig. 2 (a), where $x(u_i)$ and $y(v_j)$ denote the x-coordinate of the i-th red point and the y-coordinate of the j-th blue point, respectively. By abuse of notation, we will also use $R[i, j, k]$ to refer to the set of (blue) points inside this region.

Let $P[i, j, k]$ denote the probability that the subset $U_i = \{u_1, u_2, \ldots, u_i\}$ of red points does **not** have a neighbor within distance ≤ 1 in $R[i, j, k]$. The value we are interested in is $P[n, m, 1]$, which is the probability that the closest pair distance is > 1. We fill up the table for $P[i, j, k]$ values using dynamic programming, row by row starting from u_1 (the rightmost red point).

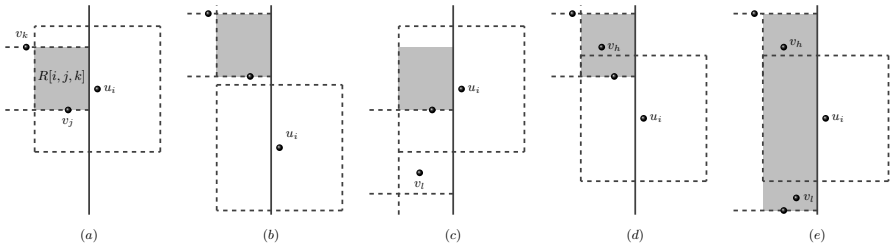

Fig. 2. Different configurations of $R[i, j, k]$ and $B(u_i)$.

Let $B(u_i)$ be the L_∞ ball of radius 1 around u_i. In the computation of the entry $P[i, j, k]$, there are 4 cases:

1. $B(u_i)$ contains the region $R[i, j, k]$ (Fig. 2 (a)). In this case

$$P[i, j, k] = p_i \prod_{v_t \in B(u_i)} (1 - q_t) + (1 - p_i) \cdot P[i - 1, j, k].$$

2. $B(u_i)$ does not intersect with $R[i,j,k]$ (Fig. 2 (b)). In this case, $P[i,j,k] = P[i-1,j,k]$.
3. $B(u_i)$ partially intersects with $R[i,j,k]$. If $y(u_i) - 1 < y(v_k)$ (Fig. 2 (c)),

$$P[i,j,k] = p_i \prod_{v_t \in B(u_i) \cap R[i,j,k]} (1-q_t) \cdot P[i-1,j,l] + (1-p_i) \cdot P[i-1,j,k],$$

where v_l is the highest blue point in $R[i,j,k]$ but outside $B(u_i)$.
If $y(u_i) + 1 < y(v_k)$) (Fig. 2 (d)), then

$$P[i,j,k] = p_i \prod_{v_t \in B(u_i) \cap R[i,j,k]} (1-q_t) \cdot P[i-1,h,k] + (1-p_i) \cdot P[i-1,j,k],$$

where v_h is the lowest blue point in $R[i,j,k]$ but outside $B(u_i)$.
4. $B(u_i)$ is contained in $R[i,j,k]$ (Fig. 2 (e)). In this case

$$P[i,j,k] = (1-p_i) \cdot P[i-1,j,k] + p_i \prod_{v_t \in B(u_i) \cap R[i,j,k]} (1-q_t) \cdot P[i-1,j,l] \cdot P[i-1,h,k],$$

where v_l and v_h are defined as before. The subtlety is that the two events of U_{i-1} having no close neighbor in $R[i-1,j,l]$, and U_{i-1} having no close neighbor in $R[i-1,h,k]$ are independent. Therefore we can multiply the corresponding probabilities. The reason is that all the points in U_{i-1} that potentially have a close neighbor in $R[i-1,j,l]$ must necessarily lie below the line $y = y(u_i)$, while those potentially close to a point in $R[i-1,h,k]$ must lie above that line. The two sets are therefore disjoint.

The base case ($R[1,j,k], j,k \in \{1,\ldots,m\}$) can be easily computed. The size of the table is $O(n^3)$. The values $\prod_{v_t \in B(u_i) \cap R[i,j,k]} (1-q_t)$ can be precomputed in $O(n^2)$ for each point v_i (by a sweep-line approach for example). This brings the computation time of each table entry down to a constant, and the running time of the whole algorithm to $O(n^3)$. This assumes a nonstandard RAM model of computation where each arithmetic operation on large numbers takes unit time.

Next Theorem considers the problem for $d > 2$.

Theorem 5. *Given a set R of red and a set B of blue points in a Euclidean space of dimension $d > 2$, each being present with an independent, rational probability, it is* NP-hard *to compute the probability that the L_∞ distance between the closest pair of bichromatic points is less than a given value r, even when the two sets are linearly separable by a hyperplane orthogonal to some axis.*

Proof. Let $d_\infty(R,B)$ be the minimum L_∞ distance between all the bichromatic pairs. It is always possible to make the two sets linearly separable in $d = 3$ by lifting all the blue (or red) points from the plane by a small value $\epsilon < d_\infty(R,B)$. This does not change the L_∞ distance of any pair of points. Therefore, an algorithm for solving the problem for linearly separable sets in $d > 2$, is essentially an algorithm for the stochastic bichromatic closest pair problem, which is NP-hard by Theorem 4. This completes the proof. □

4 Stochastic Approximate Nearest Neighbor Queries

Given a stochastic set M of points in a d-dimensional Euclidean space, and a query point q, what is the expected (L_2) distance of q to the closest present point of M? In this section we target this problem, and design a data structure for approximating the expected value of $d(S,q) = \min_{p\in S} d(p,q)$ with respect to a random subset S of M, assuming that d is a constant. (Technically, at least one point needs to be assigned probability 1 to ensure that the expected value is finite; alternatively, we can consider the expectation conditioned on the event that $d(S,q)$ is upper-bounded by a fixed value.) We obtain a linear-space data structure with $O(\log n)$ query time. Although our method is based on known techniques for approximate nearest neighbor search (namely, balanced quadtrees and shifting [3–5]), a nontrivial adaptation of these techniques is required to solve the stochastic version of the problem.

4.1 Approximation via a modified distance function $\widetilde{\ell}$

As before, we are given a set M of points in a d-dimensional Euclidean space and each point is only *present* with an independent probability. Assume that the points lie in the universe $\{0,\ldots,2^w-1\}^d$. Fix an odd integer $k = \Theta(1/\varepsilon)$. Shift all points in M by the vector $(j2^w/k, j2^w/k, \ldots, j2^w/k)$ for a randomly chosen $j \in \{0,\ldots,k-1\}$.

A *quadtree box* is a box of the form $[i_1 2^\ell, (i_1+1)2^\ell) \times \cdots \times [i_d 2^\ell, (i_d+1)2^\ell)$ for natural numbers $\ell, i_1, \ldots, i_d$. Given points p and q, let $\mathcal{D}(p,q)$ be the side length of the smallest quadtree box containing p and q. Let $B_s(p)$ be the quadtree box of side length $\lfloor\!\lfloor s \rfloor\!\rfloor$ containing p, where $\lfloor\!\lfloor s \rfloor\!\rfloor$ denotes the largest power of 2 smaller than s. Let $c_s(p)$ denote the center of $B_s(p)$. Let $[X]$ be 1 if X is true, and 0 otherwise.

Definition 1. (a) *Define* $\ell(p,q) = d(B_s(p), B_s(q)) + 2\sqrt{d}s$ *with* $s = \varepsilon^2\mathcal{D}(p,q)$. *Let* $\ell(S,q) = \min_{p\in S}\ell(p,q)$.

(b) r *is said to be* q-good *if the ball centered at* $c_{\varepsilon^2 r}(q)$ *of radius* $2r$ *is contained in* $B_{12kr}(q)$.

(c) *Define* $\widetilde{\ell}(S,q) = [\ell(S,q) \text{ is } q\text{-good}] \cdot \ell(S,q)$.

Lemma 5. (a) $\ell(S,q) \geq d(S,q)$. *Furthermore, if* $\ell(S,q)$ *is* q*-good, then* $\ell(S,q) \leq (1+O(\varepsilon))d(S,q)$.

(b) $\ell(S,q)$ *is* q*-good for all but at most* d *choices of the random index* j.

(c) $\widetilde{\ell}(S,q) \leq (1+O(\varepsilon))d(S,q)$ *always, and* $\mathbb{E}_j[\widetilde{\ell}(S,q)] \geq (1-O(\varepsilon))d(S,q)$.

Proof. Let $p^*, p \in S$ satisfy $d(S,q) = d(p^*,q) = r^*$ and $\ell(S,q) = \ell(p,q) = r$.

The first part of (a) follows since $\ell(p,q) \geq d(p,q)$. For the second part of (a), suppose that r is q-good. Since $d(p^*,q) \leq d(p,q) \leq \ell(p,q) = r$, we have $d(p^*, c_{\varepsilon^2 r}(q)) < 2r$, implying $\mathcal{D}(p^*,q) \leq 12kr$. Then $r = \ell(p,q) \leq \ell(p^*,q) \leq d(p^*,q) + O(\varepsilon^2\mathcal{D}(p^*,q)) \leq r^* + O(\varepsilon r)$, and so $r \leq (1+O(\varepsilon))r^*$.

For (b), we use [6, Lemma 2.2], which shows that the following property holds for all but at most d choices of j: the ball centered at q with radius $3r^*$ is

contained in a quadtree box with side length at most $12kr^*$. By this property, $\mathcal{D}(p^*, q) \leq 12kr^*$, and so $r = \ell(p, q) \leq \ell(p^*, q) \leq d(p^*, q) + O(\varepsilon^2 \mathcal{D}(p^*, q)) = (1 + O(\varepsilon))r^*$. Then the ball centered at $c_{\varepsilon^2 r}(q)$ of radius $2r$ is contained in the ball centered at q of radius $(2+O(\varepsilon^2))r < 3r^*$, and is thus contained in $B_{12kr^*}(q)$.

(c) follows from (a) and (b), since $1 - d/k \geq 1 - O(\varepsilon)$ (and $d(S, q)$ does not depend on j). □

By (c), $\mathbb{E}_j[\mathbb{E}_S[\widetilde{\ell}(S, q)]]$ approximates $\mathbb{E}_S[d(S, q)]$ to within factor $1 \pm O(\varepsilon)$. It suffices to give an exact algorithm for computing $\mathbb{E}_S[\widetilde{\ell}(S, q)]$ for a query point q for a fixed j; we can then return the average, over all k choices of j.

4.2 The data structure: a BBD tree

We use a version of Arya et al.'s balanced box decomposition (BBD) tree [3]. We form a binary tree T of height $O(\log n)$, where each node stores a cell, the root's cell is the entire universe, a node's cell is equal to the disjoint union of the two children's cells, and each leaf's cell contains $\Theta(1)$ points of M. Every cell B is a difference of a quadtree box (the *outer box*) and a union of $O(1)$ quadtree boxes (the *holes*). Such a tree can be constructed by forming the compressed quadtree and repeatedly taking centroids, as described by Arya et al. (in the original BBD tree, each cell has at most 1 hole and may not be perfect hypercubes). We will store $O(1/\varepsilon^{O(1)})$ amount of extra information (various expectation and probability values) at each node. The total space is $O(n/\varepsilon^{O(1)})$.

4.3 An exact query algorithm for $\widetilde{\ell}$

In this section, we describe the algorithm for estimating $\mathbb{E}_S[\widetilde{\ell}(S, q)]$, given a query point q. First we extend the definition of $\widetilde{\ell}$ slightly: let $\widetilde{\ell}(S, q, r_0) = [\ell(S, q) \leq r_0] \cdot [\ell(S, q) \text{ is } q\text{-good}] \cdot \ell(S, q)$.

Consider a cell B of T and a query point $q \in B$. Let $R(B^c, q)$ denote the set of all possible values for $\ell(p, q)$ over points p in B^c, the complement of B. We solve the following extension of the query problem (all probabilities and expectations are with respect to the random subset S):

Problem 1. For every $r_0 \in R(B^c, q)$, compute the values $\Pr[\ell(S \cap B, q) > r_0]$ and $\mathbb{E}[\widetilde{\ell}(S \cap B, q, r_0)]$.

It suffices to compute these values for $r_0 \leq \sqrt{d}|B|$, where $|B|$ denotes the maximum side length of B, since they don't change as r_0 increases beyond $\sqrt{d}|B|$.

Lemma 6. *The number of elements in $R(B^c, q)$ that are below $\sqrt{d}|B|$ is $O(1/\varepsilon^{2d})$.*

Proof. If p is inside a hole H of B, then $\mathcal{D}(p, q) \geq |H|$, so we can consider a grid of side length $\Theta(\varepsilon^2|H|)$ and round p to one of the $O(1/\varepsilon^{2d})$ grid points without affecting the value of $\ell(p, q)$.

If p is outside the outer box of B, then $\mathcal{D}(p, q) \geq |B|$, so we can round p using a grid of side length $\Theta(\varepsilon^2|B|)$. In this case the number of grid points for $d(p, q) \leq \ell(p, q) \leq \sqrt{d}|B|$ is $O(1/\varepsilon^{2d})$ as well. □

We now describe the query algorithm. The base case when B is a leaf is trivial. Let B_1 and B_2 be the children cells of B. Without loss of generality, assume that $q \in B_2$ (i.e., $q \notin B_1$). We apply the following formulas, based on the fact that $\ell(S \cap B, q) = \min\{\ell(S \cap B_1, q), \ell(S \cap B_2, q)\}$ and that $S \cap B_1$ and $S \cap B_2$ are independent:

$$\Pr[\ell(S \cap B, q) > r_0] \;=\; \Pr[\ell(S \cap B_1, q) > r_0] \cdot \Pr[\ell(S \cap B_2, q) > r_0]; \tag{2}$$

$$\begin{aligned}
&\mathbb{E}[\widetilde{\ell}(S \cap B, q, r_0)] \\
&= \sum_{r \le \sqrt{d}|B_2|} \Pr[\ell(S \cap B_1, q) = r] \cdot \mathbb{E}[\widetilde{\ell}(S \cap B_2, q, \min\{r, r_0\})] \;+ && (3)\\
&\quad \sum_{r \le \sqrt{d}|B_2|} \Pr[\ell(S \cap B_1, q) = r] \cdot \Pr[\ell(S \cap B_2, q) > r] \cdot [r < r_0] \cdot [r \text{ is } q\text{-good}] \cdot r && (4)\\
&\quad + \Pr[\ell(S \cap B_1, q) > \sqrt{d}|B_2|] \cdot \mathbb{E}[\widetilde{\ell}(S \cap B_2, q, r_0)] && (5)\\
&\quad + \mathbb{E}\left[[\ell(S \cap B_1, q) > \sqrt{d}|B_2|] \cdot \widetilde{\ell}(S \cap B_1, q, r_0)\right] \cdot \Pr[S \cap B_2 = \emptyset]. && (6)
\end{aligned}$$

(3) and (5) cover the case when $\ell(S \cap B_2, q) \le \ell(S \cap B_1, q)$, and (4) and (6) cover the case when $\ell(S \cap B_1, q) < \ell(S \cap B_2, q)$. For (5), note that $\ell(S \cap B_2, q) \le r_0$ already implies $S \cap B_2 \neq \emptyset$ and $\ell(S \cap B_2, q) \le \sqrt{d}|B_2|$.

By recursively querying B_2, we can compute all probability and expectation expressions concerning $S \cap B_2$ in (2)–(6). Note that $r_0 \in R(B^c, q) \subseteq R(B_2^c, q)$, and in the sums (3) and (4), it suffices to consider $r \in R(B_2^c, q)$ since $S \cap B_1 \subset B_2^c$. In particular, the number of terms with $r \le \sqrt{d}|B_2|$ is $O(1/\varepsilon^{2d})$, as already explained. For the probability and expectation expressions concerning $S \cap B_1$, we examine two cases:

- Suppose that q is inside a hole H of B_1. For all $p \in B_1$, $\mathcal{D}(p, q) \ge |H|$ and $\ell(p, q) \ge \Omega(\varepsilon^2|H|)$, so we can consider a grid of side length $\Theta(\varepsilon^4|H|)$ and round q to one of the $O(1/\varepsilon^{4d})$ grid points without affecting the value of $\ell(p, q)$, nor affecting whether $\ell(p, q)$ is q-good. Thus, all expressions concerning $S \cap B_1$ remain unchanged after rounding q. We can precompute these $O(1/\varepsilon^{O(1)})$ values for all grid points q (in $O(n/\varepsilon^{O(1)})$ time) and store them in the tree T.
- Suppose that q is outside the outer box of B_1. For all $p \in B_1$, $\mathcal{D}(p, q) \ge |B_1|$, so we can consider a grid of side length $\Theta(\varepsilon^2|B_1|)$ and round each point $p \in M \cap B_1$ to one of the $O(1/\varepsilon^{2d})$ grid points without affecting the value of $\ell(p, q)$. Duplicate points can be condensed to a single point by combining their probabilities; we can precompute these $O(1/\varepsilon^{2d})$ probability values (in $O(n)$ time) and store them in the tree T. We can then evaluate all expressions concerning $S \cap B_1$ for any given q by brute force in $O(1/\varepsilon^{O(1)})$ time.

Since the height of T is $O(\log n)$, this recursive query algorithm runs in time $O((1/\varepsilon^{O(1)}) \log n)$. Therefore we arrive at the main result of this section.

Theorem 6. *Given a stochastic set of n points in a constant dimension d, we can build an $O(n/\varepsilon^{O(1)})$-space data structure in $O((1/\varepsilon^{O(1)})n \log n)$ time, so*

that for any query point, we can compute a $(1+\varepsilon)$-factor approximation to the expected nearest neighbor distance in $O((1/\varepsilon^{O(1)})\log n)$ time.

5 Acknowledgment

The work of the first and the third author was supported in part by National Science Foundation grants CCF-0514738 and CNS-1035917.

References

1. P. Afshani, P. K. Agarwal, L. Arge, K. G. Larsen, and J. M. Phillips. (Approximate) uncertain skylines. In *ICDT*, pages 186–196, 2011.
2. P. K. Agarwal, S.-W. Cheng, Y. Tao, and K. Yi. Indexing uncertain data. In *PODS*, pages 137–146, 2009.
3. S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45:891–923, 1998.
4. T. M. Chan. Approximate Nearest Neighbor Queries Revisited. *Discrete and Computational Geometry*, 20:359–373, 1998.
5. T. M. Chan. Closest-point problems simplified on the RAM. In *Proc. SODA*, pages 472–473, 2002.
6. T. M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *J. Algorithms*, 46:178–189, 2003.
7. M. De Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational geometry: algorithms and applications*. Springer, 2008.
8. M. R. Garey and D. S. Johnson. The Rectilinear Steiner Tree Problem is NP-Complete. *SIAM Journal on Applied Mathematics*, 32(4):pp. 826–834, 1977.
9. G. H. Hardy, G. Polya, and J. E. Littlewood. *Inequalities*. Cambridge Press, 1952.
10. P. Kamousi, T. Chan, and S. Suri. Stochastic Minimum Spanning Trees in Euclidean Spaces. In *Proc. SoCG*, 2011. (To appear).
11. D. A. Klain and G. Rota. *Introduction to Geometric Probability*. Cambridge, 1997.
12. D. E. Knuth. *The Art of Computer Programming, Volume III: Sorting and Searching*. Addison-Wesley, 1973.
13. M.-S. Lin and Y.-J. Chen. Counting the number of vertex covers in a trapezoid graph. *Inf. Process. Lett.*, 109:1187–1192, 2009.
14. M. Löffler and M. J. van Kreveld. Largest and Smallest Convex Hulls for Imprecise Points. *Algorithmica*, 56(2):235–269, 2010.
15. J. S. Provan and M. O. Ball. The Complexity of Counting Cuts and of Computing the Probability that a Graph is Connected. *SIAM J. Comput.*, 12(4):777–788, 1983.
16. S. P. Vadhan. The Complexity of Counting in Sparse, Regular, and Planar Graphs. *SIAM Journal on Computing*, 31:398–427, 1997.
17. L. Valiant. Universality Considerations in VLSI Circuits. *IEEE Trans. Computers*, 30:135–140, 1981.
18. L. G. Valiant. The Complexity of Enumeration and Reliability Problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.
19. M. J. van Kreveld, M. Löffler, and J. S. B. Mitchell. Preprocessing Imprecise Points and Splitting Triangulations. *SIAM J. Comput.*, 39(7):2990–3000, 2010.