Approximate Range Searching in Higher Dimension *

Bernard Chazelle† Ding Liu† Avner Magen‡

Abstract

Applying standard dimensionality reduction techniques, we show how to perform approximate range searching in higher dimension while avoiding the curse of dimensionality. Given n points in a unit ball in $\mathbb{R}^d$, an approximate halfspace range query counts (or reports) the points in a query halfspace; the qualifier "approximate" indicates that points within distance ε of the boundary of the halfspace might be misclassified. Allowing errors near the boundary has a dramatic effect on the complexity of the problem. We give a solution with $\tilde{O}(d/\varepsilon)^2$ query time and $dn^{O(\varepsilon^{-2})}$ storage. For an exact solution with comparable query time, one needs roughly $\Omega(n^d)$ storage. In other words, an approximate answer to a range query lowers the storage requirement from exponential to polynomial. We generalize our solution to polytope/ball range searching.

1 Introduction

A staple of computational geometry [1, 2], range searching is the problem of preprocessing a set P of n points in $\mathbb{R}^d$ so that, given a region R (the range) chosen from a predetermined class (eg, all d-dimensional boxes, simplices, or halfspaces), the points of $P \cap R$ can be counted or reported quickly. The case of halfspaces is noteworthy because many range searching problems with "algebraic" ranges can be reduced to it through linearization-via-lifting. The counting version can be solved in $O(\log n)$ query time and $O(n^d / \log^d n)$ storage, while the reporting case can be handled in $O(\log n + k)$ query time and $O(n^{\lfloor d/2 \rfloor}\text{polylog}(n))$ storage, where k is the number of points to be reported [1]. In both cases, the exponential dependency on d—the so-called curse of dimensionality—is a show-stopper for large d. Lower bound work in a variety of highly reasonable models suggests that the curse of dimensionality is inevitable [4, 5].

Inspired by recent work on approximate nearest neighbor searching [8, 9, 7], we seek the mildest relaxation of the problem that will break the curse of dimensionality. Without loss of generality we assume that all the points of P lie in a unit ball of ℓ_2^d. Let S_h be a halfspace, with h denoting its bounding hyperplane. Given $\varepsilon > 0$, the *fuzzy boundary* of S_h is the slab formed by all points within distance ε of h (Fig. 1). Approximate halfspace range searching refers to counting (or reporting) the points of $P \cap S_h$, making allowance for errors regarding the points in the fuzzy boundary; in other words, the output should be the size of a set whose symmetric difference with $P \cap S_h$ lies entirely in the fuzzy boundary.

Approximate range searching is relevant in situations where the data is inherently imprecise and points near the boundary cannot be classified as being inside or outside with any certainty. In the case of reporting, of course, one can always move the boundary by ε to ensure that the output contains *every* point of $P \cap S_h$, which then allows us to retrieve the right points by filtering out the outsiders.

Theorem 1 *Approximate halfspace range searching can be solved using $dn^{O(\varepsilon^{-2})}$ storage and $\tilde{O}(d/\varepsilon)^2$ query time.*[1] *Any given query is answered correctly with arbitrarily high probability.*

Our algorithm beats the lower bound for the exact version of the problem. Indeed, it is known that in the arithmetic model $\Omega(n^{1-O(1/d\varepsilon^2)})$ query time is required, if only $dn^{O(\varepsilon^{-2})}$ storage is available [4]. Our algorithm generalizes to ranges formed by polytopes bounded by a fixed number of hyperplanes and to (Euclidean) ball range searching.

We also propose an alternative algorithm for approximate halfspace range searching with a query time of $\tilde{O}(d^2\varepsilon^{-2} + dn^{1/(1+\varepsilon)}\varepsilon^{-2})$ and storage $\tilde{O}(dn\varepsilon^{-2} + n^{1+1/(1+\varepsilon)})$—and a slightly different definition of approximation. Again, the query time is better than the solution for the exact problem, since by [4] $\Omega(n^{1-O(1)/d})$ query time is necessary when we have close to $O(n^2)$ space.

Approximate range searching does not originate with this paper. Arya and Mount [3] gave an algorithm for the problem that uses optimal $O(dn)$ storage but provides a query time of $O(\log n + \varepsilon^{-d})$, which is exponential in d. There are other differences as well. For example, range queries are assumed to be bounded regions, which rules out halfspaces. The underlying technique is based on space partitions, which is quite orthogonal to our dimension reduction approach.

2 Approximate Halfspace Range Searching

We show how to reduce approximate halfspace range searching to an approximate variant of ball range searching in the

*This work was supported in part by NSF grants CCR-998817, 0306283, ARO Grant DAAH04-96-1-0181.

†Department of Computer Science, Princeton University, {chazelle, dingliu}@cs.princeton.edu

‡Department of Computer Science, University of Toronto, avner@cs.toronto.edu

[1] The notation $\tilde{O}(f)$ stands for $O(f \text{ polylog}(f))$.

Hamming cube. Initially, we make the "homogeneous" assumption that the hyperplanes bounding the query halfspaces pass through the origin and that all of the n points lie in the Euclidean ball $\|x\|_2 \leq 1$. We relax the homogeneous condition later by lifting to one dimension higher.

2.1 The Homogeneous Case

Let v_h be the unit vector normal to h pointing inside S_h. Any point p_1 in S_h outside the fuzzy boundary is at a distance from h at least ε (Fig. 1). It follows that the angle between Op_1 and v_h is less than $\pi/2 - \varepsilon$. (We assume throughout this paper that ε is small enough.) Similarly, for a point p_2 not in S_h and outside the fuzzy boundary, the angle between Op_2 and v_h is greater than $\pi/2 + \varepsilon$. This provides a separation criterion to distinguish between points we must include and those we must not.

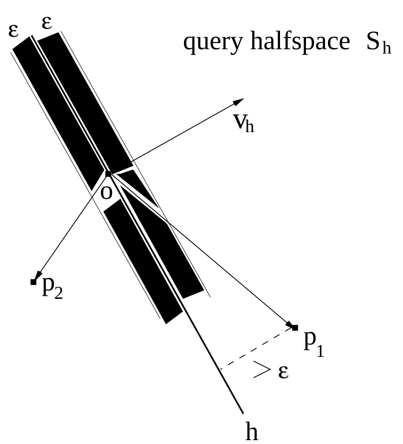

Figure 1: Approximate halfspace range searching.

Let S^{d-1} denote the unit $(d-1)$-sphere in $\mathbb{R}^d$ and let $sign(t)$ be 1 if $t \geq 0$ and -1 otherwise. Let x and y be two vectors in $\mathbb{R}^d$ and let $0 \leq \theta_{x,y} \leq \pi$ be the angle between them. We use $\mathcal{E}_{x,y}$ to denote the event: $sign(x \cdot u) = sign(y \cdot u)$. If u is uniformly distributed over S^{d-1}, then it is well known that $\mathrm{Prob}[\mathcal{E}_{x,y}] = 1 - \theta_{x,y}/\pi$. It follows that $\mathrm{Prob}[\mathcal{E}_{Op_1,v_h}] > 1/2 + \varepsilon/\pi$ and $\mathrm{Prob}[\mathcal{E}_{Op_2,v_h}] < 1/2 - \varepsilon/\pi$.

Following Kleinberg's approach [8] to nearest neighbor searching, we invoke VC-dimension theory [5, 10] to show the existence of a small number of unit vectors that can be used to distinguish between p_1 and p_2. Let $\mathcal{W}_{x,y}$ denote the subset of S^{d-1} for which $\mathcal{E}_{x,y}$ happens. Let $\mathcal{R}$ be the collection of $\mathcal{W}_{x,y}$, for all $x, y \in \mathbb{R}^d$. We consider the range space $(S^{d-1}, \mathcal{R})$. Each range $\mathcal{W}_{x,y}$ is a Boolean combination of four halfspaces; therefore the exponent of its (primal) shatter function is $2d+2$. A finite subset A of S^{d-1} is said to be a γ-approximation for the range space $(S^{d-1}, \mathcal{R})$ if, for all $R \in \mathcal{R}$, $||R \cap A|/|A| - \mu(R)| \leq \gamma$. It follows from VC dimension theory [5] that the range space $(S^{d-1}, \mathcal{R})$ admits of an $(\varepsilon/(2\pi))$-approximation A of size $O(d\varepsilon^{-2} \log(d\varepsilon^{-1}))$. Moreover, a randomly chosen set A of that size is good with high probability.

Thus, $|\mathcal{W}_{v_h,op_1} \cap A|/|A| \geq \mu(\mathcal{W}_{v_h,op_1}) - \varepsilon/(2\pi) > 1/2 + \varepsilon/(2\pi)$. Similarly, $|\mathcal{W}_{v_h,op_2} \cap A|/|A| < 1/2 - \varepsilon/(2\pi)$. For any vector x let $\widehat{x} \in \{+1,-1\}^{|A|}$ be defined as follows: the i-th coordinate of $\widehat{x}$ is $sign(x \cdot u_i)$, where u_i is the i-th vector in A according to a fixed ordering. Recall that $|\mathcal{W}_{x,y} \cap A|$ is the number of vectors $u \in A$ such that $sign(x \cdot u) = sign(y \cdot u)$. So $|\mathcal{W}_{x,y} \cap A| = |A| - d_H(\widehat{x}, \widehat{y})$ where $d_H(\cdot,\cdot)$ is the Hamming distance. We thus have $d_H(\widehat{v_h}, \widehat{op_1}) < (1/2 - \varepsilon/(2\pi))|A|$ and $d_H(\widehat{v_h}, \widehat{op_2}) > (1/2 + \varepsilon/(2\pi))|A|$.

It immediately follows that approximate halfspace range searching (under the homogeneous condition) reduces to *approximate ball range searching in the Hamming cube*: Preprocess n points in $\{+1,-1\}^{|A|}$ so that, given any $\widehat{v_h}$, the points in the Hamming ball centered at $\widehat{v_h}$ with diameter $|A|/2$ can be approximately counted (or reported) quickly. The term "approximately" means that all points within distance $(1/2 - \varepsilon/(2\pi))|A|$ must be included while all points further than $(1/2 + \varepsilon/(2\pi))|A|$ must be excluded.

2.2 The General Case

To remove the homogeneous condition, we lift the problem into $\mathbb{R}^{d+1}$. Map each point $p = (p_1, \ldots, p_d)$ to $p' = (p_1, \ldots, p_d, 1) \in \mathbb{R}^{d+1}$. Note that the new point set in $\mathbb{R}^{d+1}$ lies in a ball of radius $\sqrt{2}$. Given a query halfspace: $q_1x_1 + \cdots + q_dx_d \geq q_{d+1}$, first we compute the distance from O to its bounding hyperplane. If it exceeds $\sqrt{2}$, then all of the n points are on one side of the hyperplane and we return the exact answer (either 0 or n). Otherwise, $q_{d+1}^2 \leq 2\sum_i q_i^2$. We map the query to a new halfspace in $\mathbb{R}^{d+1}$: $q_1x_1 + \cdots + q_dx_d - q_{d+1}x_{d+1} \geq 0$. Note that the new query passes through the origin. Moreover, (i) all point-halfspace incidence relations are preserved by the map; and (ii) point-hyperplane distances are preserved to within a factor of $\sqrt{2}$ because of the upper bound on q_{d+1}^2. The problem is now reduced to the homogeneous case after suitable rescaling.

3 Approximate Ball Range Searching in the Hamming Cube

We give two solutions for approximate ball range searching in the k-dimensional Hamming cube H^k, where $k = |A| = O(d\varepsilon^{-2}\log(d\varepsilon^{-1}))$. Recall that the problem is to preprocess a set S of n points so that, given any $q \in H^k$, we can quickly count (or report) approximately the points of S within distance $k/2$ to q.

3.1 A High-Storage Solution

We adapt to the problem at hand Kushilevitz et al.'s solution to approximate nearest neighbor searching in the Hamming cube [9]. Fix two parameters m and t to be determined later. The search structure $\mathcal{S}$ consists of m substructures $\mathcal{T}_1, \ldots, \mathcal{T}_m$, all of them constructed in the same way but independently of one another. Fix $i \in \{1, \ldots, m\}$. The substructure $\mathcal{T}_i$ is built by picking t coordinates of H^k at random (out of k). Project each point $x \in H^k$ onto the subspace spanned by these t coordinates. The resulting vector $t_i(x) \in \{0,1\}^t$ is called the *trace* of x. Each $\mathcal{T}_i$ consists of a table of 2^t entries, one for each trace. Each entry stores a number (for range counting) or a pointer to a list of points (for range reporting), to be specified below. The intuition is that, as long as t is large enough, say $t = \Theta(\varepsilon^{-2} \log n)$, by a discrete analogue of Johnson and Lindenstrauss's theorem, the random projections preserve inter-point distances in appropriate range within a relative error of ε.

We say that a substructure $\mathcal{T}_i$ *fails* at query $q \in H^k$ if there exists $p \in S$ such that either of the following holds:

- $d_H(p,q) < (1/2 - \varepsilon/(2\pi))k$ but $d_H(t(p), t(q)) > (1/2 - \varepsilon/(3\pi))t$;
- $d_H(p,q) > (1/2 + \varepsilon/(2\pi))k$ but $d_H(t(p), t(q)) < (1/2 + \varepsilon/(3\pi))t$.

Lemma 2 *The probability that $\mathcal{T}_i$ fails at q is at most $ne^{-\Omega(\varepsilon^2 t)}$.*

Let $0 < c < 1$ be a constant to be specified later. We say that the structure $\mathcal{S}$ fails at q if more than cm sub-structures $\mathcal{T}_i$ fail at q.

Lemma 3 *For any $\gamma > 0$, if we set $m = (k + \log \gamma^{-1})/c$ and $t = O(\varepsilon^{-2} \ln(2en/c))$, then $\mathcal{S}$ fails nowhere with probability at least $1 - \gamma$.*

The proofs of Lemma 2 and Lemma 3 follow from standard applications of the Chernoff bounds and can be found in [9]. We thus omit them here.

Lemma 3 implies that, with high probability, for any query $q \in H^k$ and any $p \in S$, there are at least $(1-c)m$ substructures $\mathcal{T}_i$ that provide the following guarantees: (i) if $d_H(p,q) < (1/2 - \varepsilon/(2\pi))k$ then $d_H(t(p), t(q)) \le (1/2 - \varepsilon/(3\pi))t$; (ii) if $d_H(p,q) > (1/2 + \varepsilon/(2\pi))k$ then $d_H(t(p), t(q)) \ge (1/2 + \varepsilon/(3\pi))t$. In the preprocessing stage, for each entry $t_i(x)$ in the table associated with $\mathcal{T}_i$, we store the number of points $p \in S$ such that $d_H(t_i(x), t_i(p)) \le (1/2 - \varepsilon/(3\pi))t$ (for range reporting, we store a pointer to a list of such points). To answer a query q, we pick one substructure $\mathcal{T}_i \in \mathcal{S}$ uniformly at random. We compute $t_i(q)$ and use it to index the table of $\mathcal{T}_i$. We output the answer stored at that entry. By Lemma 3, with probability at least $1 - c$, the substructure $\mathcal{T}_i$ does not fail at q, and so we get a correct answer for approximate ball range queries. It is easy to see that the storage requirement is $\tilde{O}(nk + m2^t)$ (for reporting, the last term is $m2^t n$), and the query time $\tilde{O}(k)$ (+ output size for reporting). In view of Lemma 3 and the reduction shown in the last section, this proves Theorem 1.

We remark that the above algorithm, after some suitable modification, also works when each query is the intersection of a set of halfspaces. The definition for fuzzy boundary is then generalized in the obvious way. As long as the number of halfspaces is constant, the time and space bounds of Theorem 1 remain the same.

Another problem we can solve is approximate ball range searching in Euclidean space. Given a ball $B(q,r)$ in $\mathbb{R}^d$ with center q and radius r, approximate ball range searching includes all points inside the smaller ball $B(q, r - s\varepsilon)$ while excludes all points outside the larger ball $B(q, r + s\varepsilon)$, for some parameter $s = s(r)$. Points in the annulus $B(q, r+s\varepsilon) \backslash B(q, r - s\varepsilon)$ may be misclassified. In the Hamming cube, the technique described in this section solves approximate ball range searching for $s = \Theta(r)$. On the other hand, in such a solution the width of the annulus (the fuzzy region) grows with r. When r is large, it might be too big to provide an estimation of the true answer. We give another solution in which s is bounded even when r is large. Moreover, it works in Euclidean space. The idea is to reduce approximate ball range searching in $\mathbb{R}^d$ to approximate halfspace range searching in $\mathbb{R}^{d+1}$ via linearization. In this solution $s = (r+1)/r$. Thus $s = O(1)$ when $r = \Omega(1)$. The time and space bounds are essentially the same as those of approximate halfspace range searching, and in particular, the bounds of Theorem 1 apply to approximate ball range searching as well. We omit the details in this version.

3.2 A Low-Storage Solution

The storage achieved in the previous section is polynomial in n but with an exponent of $O(\varepsilon^{-2})$. We proposed another solution that uses roughly quadratic space and still provides sublinear query time. For this purpose, however, we need to relax the meaning of approximation further. If $N_r(q)$ denotes the number of points of S in the Hamming ball centered at q of radius r, then we output a number N such that $(1-\alpha)N_{(1-O(\varepsilon))k/2}(q) \le N \le (1+\alpha)N_{(1+O(\varepsilon))k/2}(q)$, for any fixed $\alpha > 0$. In section 3.6 of [6], it is shown that computing such a number N can be reduced to the $(1+\varepsilon)$-PLEB problem (stands for "Point Location in Equal Balls") with a multiplicative overhead of $\alpha^{-3} \log^2 n$ in both query time and storage. The $(1+\varepsilon)$-PLEB problem is defined as follows [6, 7]: given a set P of n points in the Hamming cube H^k and a fixed $r \le k$, preprocess P such that, given any query $q \in H^k$,

- if there exists a point $p \in P$ such that $d_H(p,q) \le r$, then answer "yes" and return a point $p' \in P$ such that $d_H(p', q) \le (1+\varepsilon)r$.
- if $d_H(p,q) > (1+\varepsilon)r$ for any $p \in P$ then answer "no".
- otherwise answer anything (either "yes" or "no").

It is shown in [7] that $(1+\varepsilon)$-PLEB in the Hamming cube H^k can be solved with query time $O(kn^{1/(1+\varepsilon)})$ and storage $(kn + n^{1+1/(1+\varepsilon)})$. Therefore approximate ball range searching can be solved with query time $\tilde{O}(dn^{1/(1+\varepsilon)}\varepsilon^{-2})$

and storage $\tilde{O}(dn\varepsilon^{-2}+n^{1+1/(1+\varepsilon)})$, following the above reduction and $k = O(d\varepsilon^{-2}\log(d\varepsilon^{-1}))$. This leads to an algorithm for approximate halfspace range searching with query time $\tilde{O}(d^2\varepsilon^{-2} + dn^{1/(1+\varepsilon)}\varepsilon^{-2})$ and storage $\tilde{O}(dn\varepsilon^{-2} + n^{1+1/(1+\varepsilon)})$, as claimed in the introduction.

References

[1] Agarwal, P.K., Erickson, J. *Geometric range searching and its relatives*, Advances in Discrete and Computational Geometry (B. Chazelle, J.E.Goodman, and R. Pollack, eds.), 1–56, AMS Press, 1999.

[2] Agarwal, P.K. *Range searching*, in “Handbook of Discrete and Computational Geometry,” eds. J.E. Goodman and J. O’Rourke, 2n ed., Chapman and Hall, CRC, 2004.

[3] Arya, S., Mount, D.M. *Approximate range searching*, Computational Geometry: Theory and Applications, 17 (2000), 135-163.

[4] Brönnimann, H., Chazelle, B., Pach, J. *How hard is halfspace range searching*, Discrete Comput. Geom. 10 (1993), 143–155.

[5] Chazelle, B. *The Discrepancy Method: Randomness and Complexity*, Cambridge University Press, 2000; paperback version 2001.

[6] Indyk, P. *High-dimensional Computational Geometry*, Ph.D. Thesis, Stanford University, 2000. http://theory.lcs.mit.edu/˜indyk/thesis.ps

[7] Indyk, P., Motwani, R. *Approximate nearest neighbors: Towards removing the curse of dimensionality*, Proc. ACM STOC (1998), 604–613.

[8] Kleinberg, J.M. *Two algorithms for nearest-neighbor search in high dimensions*, Proc. ACM STOC (1997), 599–608.

[9] Kushilevitz, E., Ostrovsky, R., Rabani, Y. *Efficient search for approximate nearest neighbor in high dimensional spaces*, SIAM J. COMPUT. 30 (2000), 457–474.

[10] Vapnik, V.N., Chervonenkis, A.Y. *On the uniform convergence of relative frequencies of events to their probabilities*, Theory of Prob. App. 16 (1971), 264–280.