Analysis of half-space range search using the k-d search skip list

Mario A. Lopez* Bradford G. Nickerson†

1 Abstract

We analyse the average cost of half-space range reporting for the k-d search skip list. The k-d search skip list is a dynamic data structure requiring $\Theta(kn)$ space for a set of n k-dimensional points. Under the assumption of uniform random distribution of points in k-d space, and a random query hyperplane, our results show that for the 2-d case, the average running time for half-space range reporting using the k-d search skip list is $O(\sqrt{n}+t)$, where t is the number of reported points.

Keywords: multidimensional data, algorithms, half-space range reporting, random hyperplane

2 Introduction

Algorithms for orthogonal range search and half space range search are extensively covered in the literature (e.g. [1, 3, 5, 11, 13]). For a survey of extensive literature on the subject the reader may want to consult [6] or [2]. As mentioned in [14], algorithms for range search traditionally fall into two categories: best worst case search time achievable by a linear space data structure, and minimum storage space to enable polylog (worst case) search time. The algorithm we analyse here falls into the former category.

The distribution of multidimensional data is unlikely to occur in patterns leading to worst case range search cases. Expected case analysis of range search algorithms is often a good predictor of their performance (e.g. [4, 7, 10]). The k-d search skip list [9] is a linear space data structure (based on the 1-3 deterministic skip list of [8]) supporting k-d range search in $O(kn)$ time, with dynamic update time of $O(k \log n)$. Figure 1 illustrates the eight rectangles of the internal nodes of a 2-d search skip list containing 14 points. Figure 2 shows a simplified version of the 2-d search skip list arising from inserting the 14 points of Figure 1. Notice that each interior node contains the bounding box of all nodes in its down sublist.

Here we analyse the expected time for half-space k-d range reporting under the assumption that both the input points and the query hyperplane are randomly and uniformly distributed. In addition, we change the structure of the k-d skip list in two ways, as follows: (1) we use a randomized skip list instead of the deterministic one used in [9], and (2) points are inserted into the list, not by x-coordinate, but by their position along a space-filling curve. These changes result in a simpler analysis and a better average performance. Other than these two changes, the data structure analysed here is identical to that of [9].

3 Random Hyperplane

In $\Re^k$, a hyperplane h is defined as the set of points X satisfying

$$X : \boldsymbol{a} \cdot \boldsymbol{x} = c$$

where $\boldsymbol{a} = (a_1, a_2, \ldots, a_k)$ are the hyperplane coefficients, $\boldsymbol{x} \in X$, and $c \in \Re$ is a constant. We say $\boldsymbol{x} = (x_1, x_2, \ldots, x_k)$ belongs to positive half-space h^*, i.e. $\boldsymbol{x} \in h^*$, if $\boldsymbol{a} \cdot \boldsymbol{x} - c > 0$.

A *random* hyperplane is equally likely to be in any orientation or position. Assume, without loss of generality, that a random hyperplane intersects the unit hypercube $[0,1]^k$. Vector $\boldsymbol{a}$ is a unit normal to the hyperplance, and $\boldsymbol{a} = (a_1, a_2, \ldots, a_k)$ are direction cosines, each in $[-1, 1]$. the direction cosines of the unit vector perpendicular to the hyperplane, and $c \in [0, 1]$ is a random variable that fixes the hyperplane such that it intersects the unit hypercube, then we have defined a random hyperplane. Considering $k-1$ of the angles θ defining the direction cosines as independent random variables, then along with c, we have k random variables defining a random hyperplane.

4 Rectangle Intersection

Internal nodes of the k-d search skip list store the k-d bounding box of all points in the down sublist. In 2-d, the bounding box is a rectangle, and we have the situation as shown in Figure 3. Coordinates of the lower left and upper right corners of a rectangle are (L, B) and (R, T), respectively. We consider

*Department of Computer Science, University of Denver, Denver, Colorado, U.S.A., mlopez@cs.du.edu

†Faculty of Computer Science, University of New Brunswick, Fredericton, N.B. Canada, bgn@unb.ca Research supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

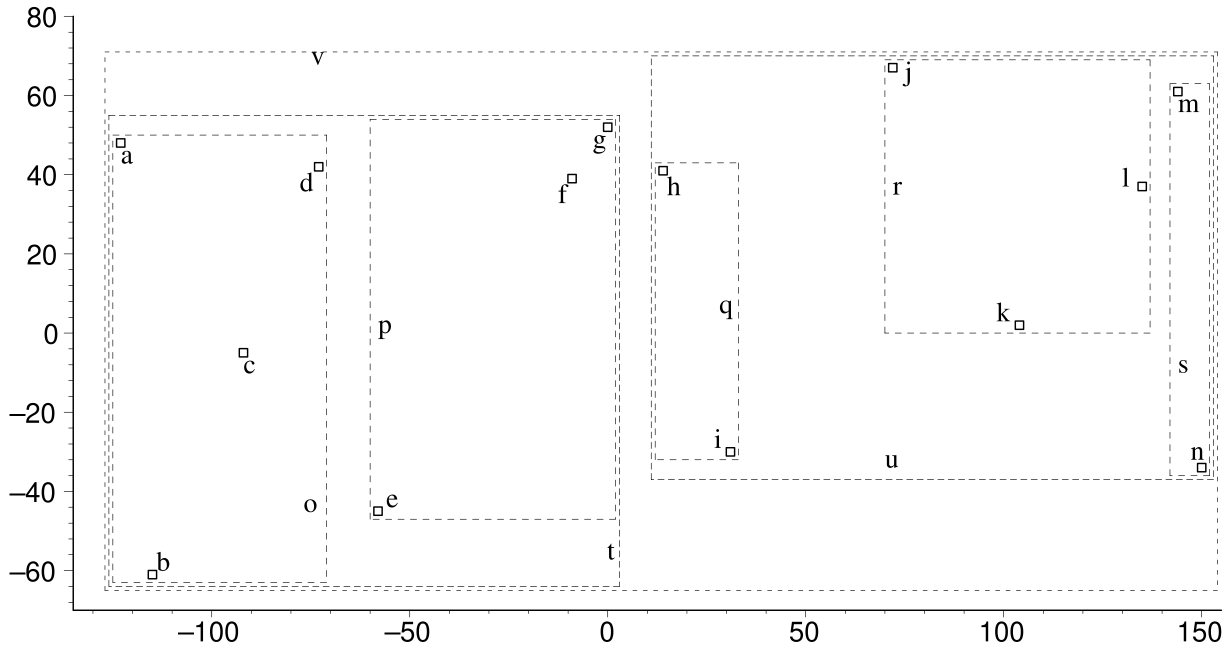

Figure 1: Rectangles induced by 14 points (labelled a through n) stored in a 2-d search skip list.

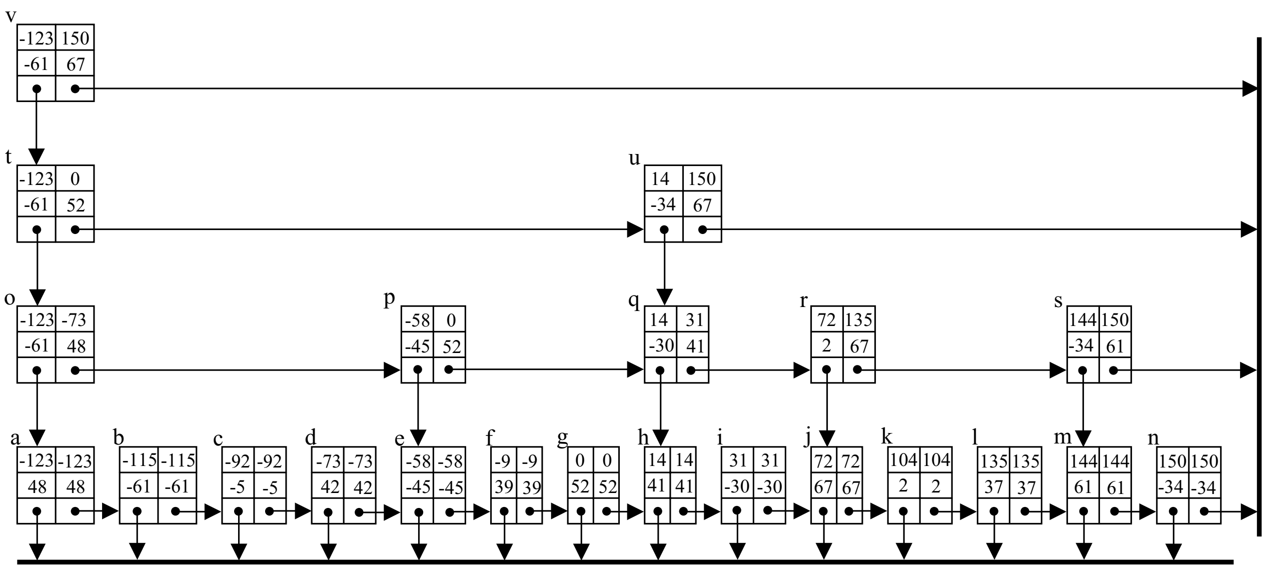

Figure 2: Simplified representation of a 2-d search skip list for the points shown in Figure 1.

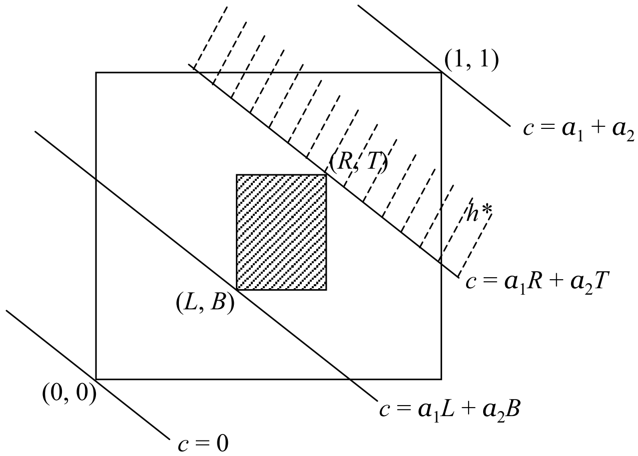

Figure 3: A rectangle with random half-space in orientation I.

four possible orientations of a query plane depending on the sign of a_1 and a_2. Orientation I refers to the unit normal direction cosines $a_i \in [0, 1]$, as shown in Figure 3.

Theorem 1 *In 2-d, the probability P of a random half-space intersecting a rectangle in the unit square $[0, 1]^2$ is $\frac{1}{2} + \frac{(T-B)+(R-L)}{4}$.*

Proof Halfspace h^*, in orientation I, intersects the rectangle iff (R, T) is above line h. Since c is distributed uniformly in the range $[0, a_1 + a_2]$, the probability that h^* intersects the rectangle is

$$\frac{Ra_1 + Ta_2}{a_1 + a_2} = \frac{R\cos\theta + T\sin\theta}{\cos\theta + \sin\theta}$$

Accordingly, the probability of the positive half space h^* in orientation I intersecting the rectangle is

$$P_I = \frac{2}{\pi}\int_0^{\frac{\pi}{2}} \frac{R\cos\theta + T\sin\theta}{\cos\theta + \sin\theta} d\theta \qquad (1)$$

for all possible orientations $\theta \in \left[0, \frac{\pi}{2}\right]$ of the unit normal to the random line defining h^*, as well as the fraction of the unit square where h^* intersects the rectangle. Evaluating equation 1 gives $P_I = \frac{R+T}{2}$.

The three other possible orientations of the line defining h^* are now considered. They are

Orientation II: $a_1 \in [-1, 0]$, $a_2 \in [0, 1]$,

Orientation III: $(a_1, a_2) \in [-1, 0]$, and

Orientation IV: $a_1 \in [0, 1]$, $a_2 \in [-1, 0]$.

These three orientations are shown in Figure 4. The probability of the positive half space h^* intersecting the rectangle in orientations II, III and IV are

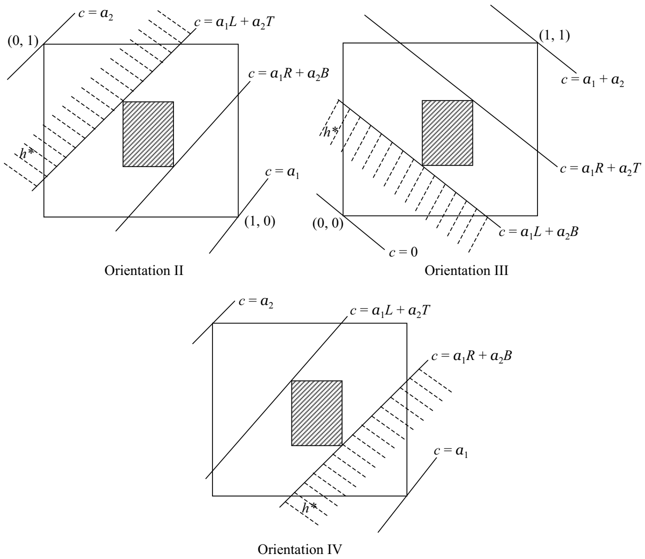

Figure 4: A rectangle with random half-space in the three other orientations.

$$\begin{aligned} P_{II} &= \frac{2}{\pi}\int_{\frac{\pi}{2}}^{\pi} \frac{L\cos\theta + T\sin\theta - \cos\theta}{\sin\theta - \cos\theta} d\theta \\ &= \frac{1 + T - L}{2} \end{aligned} \qquad (2)$$

$$\begin{aligned} P_{III} &= \frac{2}{\pi}\int_{-\pi}^{\frac{-\pi}{2}} \frac{(1-L)\cos\theta + (1-B)\sin\theta}{\sin\theta + \cos\theta} d\theta \\ &= 1 - \frac{L+B}{2} \end{aligned} \qquad (3)$$

$$
\begin{aligned}
P_{IV} &= \frac{2}{\pi}\int_{\frac{-\pi}{2}}^{0} \frac{R\cos\theta + B\sin\theta - \sin\theta}{\cos\theta - \sin\theta} d\theta \\
&= \frac{1+R-B}{2} \qquad (4)
\end{aligned}
$$

Equations 2, 3 and 4 account for all possible orientations $\theta \in \left[\frac{\pi}{2}, \pi\right]$, $\theta \in \left[-\pi, \frac{-\pi}{2}\right]$, and $\theta \in \left[\frac{-\pi}{2}, 0\right]$, respectively, of the unit normal to the random line defining h^*, as well as the fraction of the unit square where h^* intersects the rectangle. Adding all four components of the probability together, and dividing by four to account for the fact that each component contributes $\frac{1}{4}$ of the probability gives

$$P = \frac{P_I + P_{II} + P_{III} + P_{IV}}{4} = \frac{1}{2} + \frac{R-L}{4} + \frac{T-B}{4}. \quad \blacksquare$$

Thus, the probability of a random half-space intersecting a rectangle in the unit square is $\geq \frac{1}{2}$. Note, in particular, that this probability depends exclusively on the size of the rectangle and not on its location.

For k-d, we offer the following conjecture:

Conjecture 1 *For k-d,* $P = \frac{1}{2} + \frac{1}{2k}\left(\sum_{i=1}^{k} \Delta x_i\right)$.

In conjecture 1, Δx_i is the side length of the rectangle in dimension i.

5 2-d Search Skip List

For Q_n = time complexity of half-space range search in a data structure containing n points, we have

$$\mathbf{E}\{Q_n\} = \sum_{i=1}^{no. of interior nodes} P\{R_i \cap h \neq 0\}, \qquad (5)$$

where $\mathbf{E}$ is the expected value, P is the probability, and R_i refers to a rectangle representing the bounding box of one internal node of the search skip list. The algorithm for half-space range search in a k-d search skip list [9] prunes a node and it's down sublist if all points in R_i are not in h^*; i.e. R_i's down sublist is pruned if $\boldsymbol{x} \notin h^*$, $\forall \boldsymbol{x} \in R_i$. Symmetrically, the algorithm reports all points within a node's down sublist as in range if all points in R_i are in h^*; i.e. report all points in node R_i's down sublist if $\boldsymbol{x} \in h^*$, $\forall \boldsymbol{x} \in R_i$. Without counting the cost of nodes traversed solely for reporting purposes (similar to Willard's [12] worst-case locate runtime), the cost of half-space range search arises from those nodes whose bounding box intersects the hyperplane h.

Lemma 2 *In 2-d, the probability P of a random half-plane h intersecting a rectangle in the unit square $[0,1]^2$ is* $\frac{(T-B)+(R-L)}{2}$.

Proof The proof follows the same technique used for proving Theorem 1, but accounting for the probability of a random half-plane rather than a half-space intersecting a rectangle. Equations (1), (2), (3) and (4) become

$$P_I = \frac{2}{\pi}\int_{0}^{\frac{\pi}{2}} \frac{(R-L)\cos\theta + (T-B)\sin\theta}{\cos\theta + \sin\theta} d\theta \quad (6)$$

$$P_{II} = \frac{2}{\pi}\int_{\frac{\pi}{2}}^{\pi} \frac{(L-R)\cos\theta + (T-B)\sin\theta}{\sin\theta - \cos\theta} d\theta \quad (7)$$

$$P_{III} = \frac{2}{\pi}\int_{-\pi}^{\frac{-\pi}{2}} \frac{(L-R)\cos\theta + (B-T)\sin\theta}{-\sin\theta - \cos\theta} d\theta \quad (8)$$

$$P_{IV} = \frac{2}{\pi}\int_{\frac{-\pi}{2}}^{0} \frac{(R-L)\cos\theta + (B-T)\sin\theta}{\cos\theta - \sin\theta} d\theta \quad (9)$$

Evaluating these equations gives

$$P_I = P_{II} = P_{III} = P_{IV} = \frac{(T-B)+(R-L)}{2} \quad (10)$$

Adding all four components of the probability together, and dividing by four to account for the fact that each component contributes $\frac{1}{4}$ of the probability gives the result. $\blacksquare$

With Lemma 2 in place, we can now proceed to the main result.

Theorem 3 *Consider a 2-d search skip list S of n points drawn at random from a uniform distribution and inserted into S by Hilbert order. The expected time $\mathbf{E}\{Q_n\}$ of half-space range searching with a random query plane is $O(\sqrt{n}+t)$, where t is the number of reported points.*

Proof Let $w+1$ be the number of levels of S, labeled $0, \ldots, w$, and S', the set of internal nodes of S, i.e., the set of nodes at levels $0 \ldots, w-1$. With each node v of S' we define an indicator variable I_v associated with the event $h \cap R \neq \emptyset$, where R is the bounding rectangle stored at v. Thus, v is further explored by going into its down sublist iff $I_v = 1$.

Let v be an arbitrary node of S' with bounding rectangle R of size $w \times h$. Let v_i be the children of v with bounding rectangles R_i. By Lemma 2, $\mathbf{E}\{I_v\} = (w+h)/2$. Since the expected length of a side of R_i is no more than 1/2 the length of the corresponding side of R, we have $\mathbf{E}\{I_{v_i}\} \leq (w+h)/4$. Coupled with the fact that $\mathbf{E}\{I_r\} \leq 1$ for the root r, we have that $\mathbf{E}\{I_v\} \leq \frac{1}{2}(2^{1-j}) = 2^{-j}$, where j is the level of node v (and $j=0$ at the root). Since $\mathbf{E}\{w\} = \lceil \log_4 n \rceil$ and the expected number of nodes of S at level j is 4^j then the expected number of nodes explored in S is at most

$$\sum_{v \in L'} \mathbf{E}\{I_v\} \leq \sum_{j=0}^{w-1} 4^j 2^{-j} = \sum_{j=0}^{w-1} 2^j = 2^w - 1 = O(\sqrt{n})$$

Other nodes visited correspond to points inside h^* and add $O(t)$ to the complexity, establishing the claim. ∎

The above proof assumes a randomized skip list is used, where the expected number of nodes in each down sublist is four.

6 Conclusions

We have shown that the expected cost of half-space range search using the k-d search skip list containing n points is $O(\sqrt{n}+t)$ in the 2-d case. This result is competitive with the best known algorithms for half-space range search requiring linear space. It remains to expand the analysis to 3-d and to k-d. If our conjecture is correct, we anticipate good upper bounds for the expected case of half-space range search in higher dimensions. The approach we have chosen for defining a random hyperplane will be useful in other expected case analyses involving half-space range search. Can our approach be used to analyse the expected orthogonal range search time in an R-tree and other data structures whose internal nodes contain bounding boxes encompassing their children's bounding boxes?

References

[1] P.K. Agarwal and J. Matoušek. Dynamic half-space range reporting and its applications. *Algorithmica*, 13:325–345, 1995.

[2] M. J. Atallah, editor. *Algorithms and Theory of Computation Handbook*, chapter 20. CRC Press, Boca Raton, 1998.

[3] J.L. Bentley and H.A. Maurer. Efficient worst-case data structures for range searching. *Acta Informatica*, 13(2):155–168, 1980.

[4] P. Chanzy, L. Devroye, and C. Zamora-Cura. Analysis of range search for random k-d trees. *Acta Informatica*, 37(Fasc.4/5):355–383, 2001.

[5] B. Chazelle. Lower bounds for orthogonal range searching: I. The reporting case. *Journal of the Association for Computing Machinery*, 37(2):200–212, 1990.

[6] J.E. Goodman and J. O'Rourke, editors. *Handbook of Discrete and Computational Geometry*, chapter 31. CRC Press, Boca Raton, 1997.

[7] S.T. Leutenegger and M.A. Lopez. The effect of buffering on the performance of r-trees. *IEEE Trans. on Knowledge and Data Engineering*, 12(1):33–44, Jan/Feb 2000.

[8] J. Ian Munro, T. Papadakis, and R. Sedgewick. Deterministic skip lists. In *Proc. of the Third Annual Symposium on Discrete Algorithms*, pages 367–375, Orlando, Florida, January 27-29 1992.

[9] B.G. Nickerson and Y. Pan. On dynamic k-d halfspace range reporting. In *Proc. of the 12th Canadian Conference on Computational Geometry*, pages 187–194, Fredericton, N.B., Canada, Aug. 16-19 2000. Faculty of Computer Science.

[10] R. Seidel and C.R. Aragon. Randomized search trees. *Algorithmica*, 16:464–497, 1996.

[11] P.M. Vaidya. Space-time trade-offs for orthogonal range queries. *SIAM Journal of Computing*, 18(4):748–758, 1989.

[12] D.E. Willard. Polygon retrieval. *SIAM Journal of Computing*, 11(1):149–165, 1982.

[13] A.C. Yao. Space-time tradeoff for answering range queries. In *Proc. of the 14th Annual ACM Symposium on the Theory of Computing*, pages 128–136, San Francisco, CA, May 5-7 1982.

[14] F.F. Yao. *Handbook of Theoretical Computer Science*, volume A: Algorithms and Complexity, chapter 7, pages 370–374. Elsevier, Amsterdam, 1990.