# Significant-Presence Range Queries in Categorical Data

*Mark de Berg*

*Herman J. Haverkort*

institute of information and computing sciences, utrecht university

technical report UU-CS-2004-009

www.cs.uu.nl

# Significant-Presence Range Queries in Categorical Data

Mark de Berg* Herman J. Haverkort†

### Abstract

In traditional colored range-searching problems, one wants to store a set of $n$ objects with $m$ distinct colors for the following queries: report all colors such that there is at least one object of that color intersecting the query range. Such an object, however, could be an 'outlier' in its color class. Therefore we consider a variant of this problem where one has to report only those colors such that at least a fraction $\tau$ of the objects of that color intersects the query range, for some parameter $\tau$. Our main results are on an approximate version of this problem, where we are also allowed to report those colors for which a fraction $(1-\varepsilon)\tau$ intersects the query range, for some fixed $\varepsilon > 0$. We present efficient data structures for such queries with orthogonal query ranges in sets of colored points, and for point stabbing queries in sets of colored rectangles.

## 1 Introduction

**Motivation.** The range-searching problem is one of the most fundamental problems in computational geometry. In this problem we wish to construct a data structure on a set $S$ of objects in $\mathbb{R}^d$, such that we can quickly decide for a query range which of the input objects it intersects. The range-searching problem comes in many flavors, depending on the type of objects in the input set $S$, on the type of allowed query ranges, and on the required output (whether one wants to report all intersected objects, to count the number of intersected objects, etc.). The range-searching problem is not only interesting because it is such a fundamental problem, but also because it arises in numerous applications in areas like databases, computer graphics, geographic information systems, and virtual reality. Hence, it is not surprising that there is an enormous literature on the subject—see for instance the surveys by Agarwal [1], Agarwal and Erickson [2], and Nievergelt and Widmayer [9].

In this paper, we are interested in range searching in the context of databases. Here one typically wants to be able to answer questions like: given a database of customers, report all customers whose ages are between 20 and 30, and whose income is between \$50,000 and \$75,000. In this example, the customers can be represented as points in $\mathbb{R}^2$, and the query range is an axis-parallel rectangle.[1] This is called the (planar) *orthogonal range-searching problem*, and it has been studied extensively—see the surveys [1, 2, 9] mentioned earlier.

There are situations, however, where the data points are not all of the same type but fall into different categories. Suppose, for instance, that we have a database of stocks. Each stock

*Department of Computer Science, TU Eindhoven, P.O.Box 513, 5600 MB Eindhoven, the Netherlands. Email: m.t.d.berg@tue.nl

†Institute of Information and Computing Sciences, Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, the Netherlands, Email: herman@cs.uu.nl

[1] From now on, whenever we use terms like "rectangle" or "box" we implicitly assume these are axis-parallel.

falls into a certain category, namely the industry sector it belongs to—energy, banking, food, chemicals, etc. Then it can be interesting for an analyst to get answers to questions like: "In which sectors companies had a 10–20% increase in their stock values over the past year?" In this simple example, the input can be seen as points in 1D (namely for each stock its increase in value), and the query is a 1-dimensional range-searching query.

Now we are no longer interested in reporting all the points in the range, but in reporting only the categories that have points in the range. This means that we would like to have a data structure whose query time is not sensitive to the total number of points in the range, but to the total number of categories in the range. This can be achieved by building a suitable data structure for each category separately, but this is inefficient if the number of categories is large. This has led researchers to study so-called *colored range-searching problems*: store a given set of colored objects—the color of an object represents its category—such that one can efficiently report those colors that have at least one object intersecting a query range [3, 7, 10, 11].

We believe, however, that this is not always the correct abstracted version of the range-searching problem in categorical data. Consider for instance the stock example sketched earlier. The standard colored range-searching data structures would report all sectors that have *at least one* company whose increase in stock value lies in the query range. But this does not necessarily say anything about how the sector is performing: a given sector could be doing very badly in general, but contain a single 'outlier' whose performance has been good. It is much more natural to ask for all sectors for which *most* stocks, or at least a significant portion of them, had their values increase in a certain way. Therefore we propose a different version of the colored range-searching problem: given a fixed threshold parameter $\tau$, with $0 < \tau < 1$, we wish to report all colors such that at least a fraction $\tau$ of the objects of that color intersect the query range. We call this a *$\tau$-significant-presence query*, as opposed to the standard *presence query* that has been studied before.

**Problem statement and results.** We study significant-presence queries in categorical data in two settings: orthogonal range searching where the data is a set of colored points in $\mathbb{R}^d$ and the query is a box, and stabbing queries where the data is a set of colored boxes in $\mathbb{R}^d$ and the query is a point. We now discuss our results on these two problems in more detail.

Let $S = S_1 \cup \cdots \cup S_m$ be a set of $n$ points in $\mathbb{R}^d$, where $m$ is the number of different colors and $S_i$ is the subset of points of color class $i$. Let $\tau$ be a fixed parameter with $0 < \tau < 1$. We are interested in answering $\tau$-significant-presence queries on $S$: given a query box $Q$, report all colors $i$ such that $|Q \cap S_i| \geqslant \tau \cdot |S_i|$. For $d = 1$, we present a data structure that uses $O(n)$ storage, and that can answer significant-presence queries in $O(\log n + k)$ time, where $k$ is the number of reported colors. Unfortunately, the generalization of our approach to higher dimensions leads to a data structure using already cubic storage in the planar case. To show this fact, we obtain the following result which is of independent interest. Let $P$ be a set of $n$ points in $\mathbb{R}^d$, and $t$ a parameter with $1 \leqslant t \leqslant n/(2d)$. Then the maximum number of combinatorially distinct boxes containing exactly $t$ points from $P$ is $\Theta(n^d t^{d-1})$ in the worst case.

As a data structure with cubic storage is prohibitive in practice, we study an approximate version of the problem. More precisely, we study *$\varepsilon$-approximate significant-presence queries*: here we are required to report all colors $i$ with $|Q \cap S_i| \geqslant \tau \cdot |S_i|$, but we are also allowed to report colors with $|Q \cap S_i| \geqslant (1-\varepsilon)\tau \cdot |S_i|$, where $\varepsilon$ is a fixed positive constant. For such queries we develop a data structure that uses $O(M^{1+\delta})$ storage, for any $\delta > 0$, and that can answer such queries in $O(\log n+k)$ time, where $M = m/(\tau^{2d-2}\varepsilon^{2d-1})$ and $k$ is the number of reported colors. We obtain similar results for the case where $\tau$ is not fixed, but part of the query—see Theorem 2.2. Note that the amount of storage does not depend on $n$, the total number of points, but only on $m$, the number of colors. This should be compared to the results for the previously considered case of presence queries on colored points sets. Here the best known results are: $O(n)$ storage with $O(\log n+k)$ query time for $d = 1$ [11], $O(n \log^2 n)$ storage with $O(\log n + k)$ query time for $d = 2$ [11], $O(n \log^4 n)$ storage with $O(\log^2 n + k)$ query time for $d = 3$ [10], and $O(n^{1+\delta})$ storage with $O(\log n + k)$ query time for $d \geqslant 4$ [3]. These bounds all depend on $n$, the total number of points; this is of course to be expected, since these results are all on the exact problem, whereas we allow ourselves approximate answers.

In the point-stabbing problem we are given a parameter $\tau$ and a set $B = B_1 \cup \cdots \cup B_m$ of $n$ colored boxes in $\mathbb{R}^d$, and we wish, for a query point $q$, to report all colors $i$ such that the number of boxes in $B_i$ containing $q$ is at least $\tau \cdot |B_i|$. We study the $\varepsilon$-approximate version of this problem, where we are also allowed to report colors such that the number of boxes containing $q$ is at least $(1 - \varepsilon)\tau \cdot |B_i|$. Our data structure for this case uses $O(M^{1+\delta})$ storage, for any $\delta > 0$, and it has $O(\log n + k)$ query time, where $M = m/(\tau\varepsilon)^d$. The best results for standard colored stabbing queries, where one has to report all colors with at least one box containing the query point, are as follows. For $d = 2$, there is a structure using $O(n \log n)$ storage with $O(\log^2 n + k)$ query time [10], and for $d > 2$ there is a structure using $O(n^{1+\delta})$ storage with $O(\log n + k)$ query time [3].

## 2 Orthogonal range queries

Our global approach is to first reduce significant-presence queries to standard presence queries. We do this by introducing so-called *test sets.*

### 2.1 Test sets for orthogonal range queries

Let $P$ be a set of $n$ points in $\mathbb{R}^d$, and let $\tau$ be a fixed parameter with $0 < \tau < 1$. A set $T$ of boxes—that is, axis-parallel hyperrectangles—is called a *$\tau$-test set* for $P$ if:

1. any box from $T$ contains at least $\tau n$ points from $P$, and
2. any query box $Q$ that contains at least $\tau n$ points from $P$ fully contains at least one box from $T$.

We call the boxes in $T$ *test boxes.* We can answer a significant-presence query on $P$ by answering a presence query on $T$: a query box $Q$ contains at least $\tau n$ points from $P$ if and only if it contains at least one test box. This does not yet reduce the problem to a standard presence-query problem, because $T$ contains boxes instead of points. However, like Agarwal *et al.* [3], we can map the set $T$ of boxes in $\mathbb{R}^d$ to a set of points in $\mathbb{R}^{2d}$, and the query box $Q$ to a box in $\mathbb{R}^{2d}$, in such a way that a box $b \in T$ is fully contained in $Q$ if and only if its corresponding point in $\mathbb{R}^{2d}$ is contained in the transformed query box.[2] This means we can apply the results from the standard presence queries on colored point sets.

[2] In fact, the transformed query box is unbounded to one side along each coordinate-axis, so it is a $d$-dimensional 'octant'.

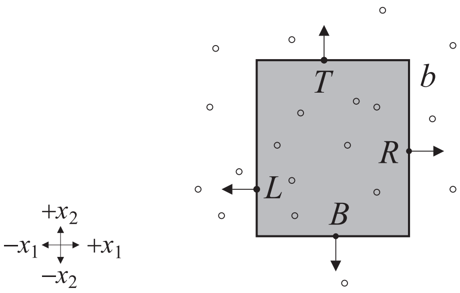

Figure 1: Peeling a $(\tau n)$-box $b$ in two dimensions ($\tau n = 12$). The black dots are the four points of $D(b)$. Initially, each point is extreme in only one direction, as indicated by the arrows. We can choose any of them, let us take $T$.

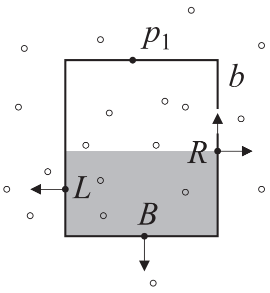

Figure 2: For $p_2$, we cannot take $R$, since it is extreme in two directions among the remaining points of $D(b)$. We have to take one of the others, for example $L$.

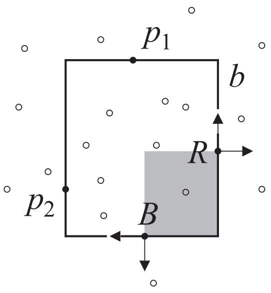

Figure 3: Now, all remaining points of $D(b)$ are extreme in 2 directions: we stop peeling here. $R$ and $B$ together form the basis $D^*(b)$ of $b$. We conclude that $b$ has a peeling sequence of type $+x_2, -x_1$.

It remains to find small test sets. As it turns out, this is not possible in general: below we show that there are point sets that do not admit test sets of near-linear size. Hence, after studying the case of exact test sets, we will turn our attention to approximate test sets.

**Exact test sets.** Let $t$ be a parameter with $1 \leqslant t \leqslant n$. Define a *t-box* to be a minimal box containing at least $t$ points from $P$, that is, a box $b$ containing at least $t$ points such that there is no strictly smaller box $b' \subset b$ that contains $t$ or more points. It is easy to see that any $(\tau n)$-box must be a test box, and that the collection of all $(\tau n)$-boxes forms a $\tau$-test set. Hence, the smallest possible test set consists exactly of these $(\tau n)$-boxes.

In the 1-dimensional case a box is a segment, and a minimal segment is uniquely defined by the point from $P$ that is its left endpoint. This means that any set of $n$ points on the real line has a test set that has size $(1-\tau)n + 1$. Unfortunately, the size of test sets increases rapidly with the dimension, as the next lemma shows.

**Lemma 2.1** *For any set $P$ of $n$ points in $\mathbb{R}^d$, there is a $\tau$-test set that has size $O(\tau^{d-1}n^{2d-1})$. Moreover, for some sets $P$, any $\tau$-test set has size $\Omega(\tau^{d-1}n^{2d-1})$.*

*Proof.* By the observation made before, bounding the size of a test set boils down to bounding the number of $(\tau n)$-boxes. In this proof, when we use the term direction we mean one of the $2d$ directions $+x_1, -x_1, ..., +x_d, -x_d$. Let $b$ be a $(\tau n)$-box, and let $D(b)$ be a set of points in $b$ such that there is at least one point of $D(b)$ on each facet of $b$. If there are more such sets, let $D(b)$ be a set with minimum cardinality.

The central concept in the proof is that of a peeling sequence, which is defined as follows: a *peeling sequence* for $D(b)$ is a sequence $p_1, p_2, ...$ of points from $D(b)$ with the following property: any $p_i$ in the sequence is extreme in exactly one direction among the points in $D(b) - \{p_1, ..., p_{i-1}\}$. Ties are broken arbitrarily, i.e. if multiple points are extreme in the same direction, we appoint one of them to be the extreme point in that direction. The *type* of a peeling sequence is the sequence $\vec{d_1}, \vec{d_2}, ...$ of directions such that $\vec{d_i}$ is the unique direction in which $p_i$ is extreme among $D(b) - \{p_1, ..., p_{i-1}\}$. Note that there are $(2d)!/(2d-\ell)! = O(1)$

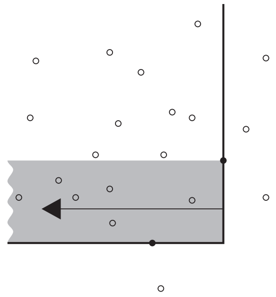

Figure 4: Constructing a $(\tau n)$-box with sequence type $+x_2, -x_1$ in two dimensions. First choose a basis of two points for the remaining directions (the black dots). Then follow the sequence type in reverse order. The extreme point for direction $-x_1$ must be one of the first $\tau n$ points found when traversing the shaded area in the direction of the arrow.

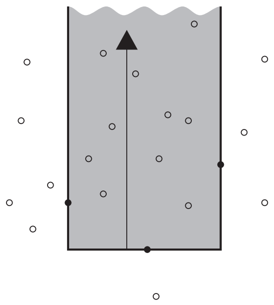

Figure 5: The extreme point for the first direction of the sequence, $+x_2$, must be the $(\tau n)$'th point in the shaded area.

different sequence types of a given length $\ell$, so we have $O(1)$ different sequence types of length between 0 and $d$.

It is easy to see that there must be a peeling sequence $\sigma(b)$ of length $q = \max(0, |D(b)|-d)$: consider an incremental construction of the sequence, peeling off points from $D(b)$ one at a time, as illustrated in Figs. 1–3. There are $2d$ directions, so as long as there are more than $d$ points left there must be a point that is extreme in only one direction, which we can peel off.

Call $D^*(b) := D(b) - \sigma(b)$ the *basis* of $b$. We charge the box $b$ to its basis $D^*(b)$, and we claim that each basis is charged $O((\tau n)^{d-1})$ times. Since there are $O(n^d)$ possible bases, this proves the theorem. To prove the claim, consider a basis $D^*$, and choose a sequence type. Any $(\tau n)$-box $b$ whose basis $D(b)$ is equal to $D^*$ and whose peeling sequence has the given type can be reconstructed incrementally as follows—see Figs. 4 and 5 for an illustration. Start with $D = D^*$. Now consider the last direction $\vec{d_q}$ of the sequence type. Since the last point $p_q$ of the peeling sequence is extreme only in direction $\vec{d_q}$, it must be contained in the semi-infinite box which is bounded in all other directions by planes through points in $D$. Hence, only the first $\tau n$ points in this semi-infinite box are candidates for $p_q$, otherwise the box would already contain too many points. A similar argument shows there are only $\tau n$ choices for $p_{q-1}, ..., p_2$. The first point $p_1$ from the sequence (which is the last point added in the reconstruction) is then fixed, as $b$ must contain exactly $\tau n$ points—see Figure 5.

To prove the lower bound, consider the following configuration (shown in Fig. 6 for the planar case). We pair the $2d$ directions $+x_1, -x_1, ..., +x_d, -x_d$ into $d$ pairs $(\vec{d}_{11}, \vec{d}_{12})$, $(\vec{d}_{21}, \vec{d}_{22})$,..., $(\vec{d}_{d1}, \vec{d}_{d2})$ so that no pair contains opposite directions, that is $\vec{d}_{i1} \neq -\vec{d}_{i2}$ for $1 \leqslant i \leqslant d$. Let $h_i$ be the 2-plane spanned by the directions $\vec{d}_{i1}$ and $\vec{d}_{i2}$ and containing the origin. On each 2-plane $h_i$, we place $n/d$ points $p_i(1), ..., p_i(n/d)$ such that all of them are in the positive quadrant with respect to the origin and both directions $\vec{d}_{i1}$ and $\vec{d}_{i2}$. We place these points along a staircase. More precisely, we require that for $1 < j \leqslant n/d$, the point $p_i(j)$ is closer to the origin than $p_i(j-1)$ with respect to direction $\vec{d}_{i1}$, and further from the origin with respect to direction $\vec{d}_{i2}$. Any box containing at least one point from each of these sets can now be specified by choosing two points $p_i(b_i)$ and $p_i(b'_i)$ in each 2-plane $h_i$; we define the box $b$ to be the minimum bounding box of the points chosen. By choosing $b'_i \leqslant b_i + (\tau n - 1)/(d-1) - 1$ for $1 \leqslant i < d$, and $b'_d = b_d - 1 + \sum_{i=1}^{d-1}(b'_i - b_i + 1)$, we get a box

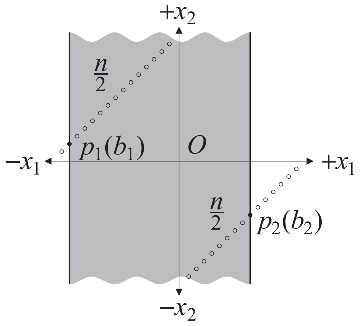

Figure 6: A lower bound on the number of $(\tau n)$-boxes in two dimensions. The four directions are grouped in two pairs $(-x_1, +x_2)$ and $(+x_1, -x_2)$. We place a staircase of $n/2$ points in the positive quadrant for each pair (in two dimensions, these quadrants are coplanar; in higher dimensions this is not necessarily the case). Choosing one defining point on each staircase fixes two sides of a box. We have $\Theta(n^2)$ ways to do so.

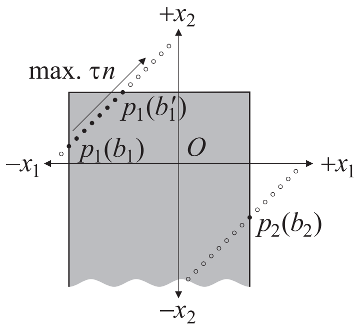

Figure 7: Choosing one additional point on one staircase fixes another side of the box. This additional point must be one of the first $\Theta(\tau n)$ points found when walking up the staircase from the first defining point on that staircase. On the remaining staircase, we will have no choice but to choose the point such that the box will contain exactly $\tau n$ points.

containing exactly $\tau n$ points. Having $\Theta(n)$ choices for each $b_i$ $(1 \leqslant i \leqslant d)$ and $\Theta(\tau n)$ choices for each $b_i'$ $(1 \leqslant i \leqslant d-1)$, we can construct $\Theta(\tau^{d-1} n^{2d-1})$ different $(\tau n)$-boxes. □

Note that already in the plane, the bound is cubic in $n$.

**Remark 2.1** A different way to state the result above is as follows. Let $P$ be a set of $n$ points in $\mathbb{R}^d$, and let $t$ be a parameter with $1 \leqslant t \leqslant n/(2d)$. Then the maximum number of combinatorially distinct boxes containing exactly $t$ points from $P$ is $\Theta(n^d t^{d-1})$. In other words, we have proved a tight bound on the number of $t$-sets for ranges that are boxes instead of hyperplanes. Since $t$-sets have been studied extensively—see e.g. [6] and [12]—we suspected that the case of box-ranges would have been considered as well, but we have only found a result on this for $t = 2$: Alon *et al.* [4] proved that the maximum number of 2-boxes is $(1 - \frac{1}{2^{2^{d-1}-1}})n^2/2 + o(n^2)$.

**Remark 2.2** The lower-bound example in the proof of Lemma 2.1 is quite contrived, and one may hope that much smaller test sets are possible if the points are distributed more regularly. This is not the case, however. As an example, consider the planar case with $\tau = 1/2$, and suppose the point set $P$ is distributed uniformly at random in the unit square. Then the number of $(n/2)$-rectangles is still $\Theta(n^3)$ with high probability. This can be seen as follows. Consider the partitioning of the unit square into nine regions, as in Fig. 2.2. Since the points are distributed uniformly, the expected number of points in a region of area $\alpha$ is $\alpha n$. Moreover, the number of points in the region is at least $(2/3)\alpha n$ with probability greater than $1 - \exp(-\alpha n/18)$, which follows from standard tail estimates on the binomial distribution. Hence, the following properties hold simultaneously with high probability:

(1) each of the three darkly shaded regions in Fig. 2.2 has $\Theta(n)$ points;

(2) the lightly shaded region has at least $n/2$ points, which also implies that the six bottommost regions together have at most $n/2$ points.

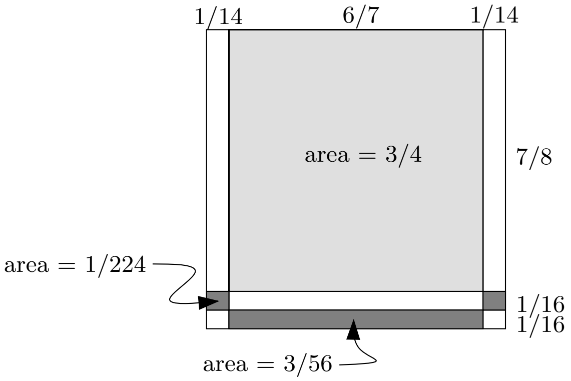

Figure 8: Partitioning of the unit square used in the argument in Remark 2.2.

It follows from (1) that there are $\Theta(n^3)$ triples of points such that each darkly shaded region contains one point from the triple, and it follows from (2) that for each such triple there is a rectangle with these points on the left, right, and bottom edge that contains exactly $n/2$ points.

**Approximate test sets.** The worst-case bound from Lemma 2.1 is quite disappointing. Therefore we now turn our attention to approximate test sets. A set $T$ of boxes is called an *$\varepsilon$-approximate $\tau$-test set* for a set $P$ of $n$ points if

1. any box from $T$ contains at least $(1-\varepsilon)\tau n$ points from $P$;
2. any query box $Q$ that contains at least $\tau n$ points from $P$ fully contains at least one box from $T$.

This means we can answer $\varepsilon$-approximate significant-presence queries on $P$ by answering a presence query on $T$.

**Lemma 2.2** *For any set $P$ of $n$ points in $\mathbb{R}^d$ ($d > 1$) and any $\varepsilon$ with $0 < \varepsilon < 1/2$, there is an $\varepsilon$-approximate $\tau$-test set of size $O(1/(\varepsilon^{2d-1}\tau^{2d-2}))$. Moreover, there are sets $P$ for which any $\varepsilon$-approximate $\tau$-test set has size $\Omega(1/(\varepsilon^{2d-1}\tau^d))$.*

*Proof.* To prove the upper bound, we proceed as follows. We will construct test sets recursively, starting with the full set $P$ as input. If the size of the current set $P$ is less than $\tau n_0$, where $n_0$ is the original number of points, there is nothing to do. Otherwise, we choose a hyperplane $h$ orthogonal to the $x_1$-axis, such that at most half of the points in $P$ lies on either side of $h$. Then we construct three test sets, one for queries on one side of $h$, one for queries on the other side, and one for queries intersecting $h$. The first two test sets are constructed by applying the procedure recursively. The latter set is constructed as follows.

Let $n$ be the number of points in the current set $P$. We construct a collection $H_2(P)$ of $n(2d-1)/(\varepsilon\tau n_0)$ hyperplanes orthogonal to the $x_2$-axis, such that there are $\varepsilon\tau n_0/(2d-1)$ points of $P$ between any pair of consecutive hyperplanes.[3] We do the same for the other axes, except the $x_1$-axis, obtaining sets $H_3(P), \ldots, H_d(P)$.

[3] If there are more points with the same $x_2$-coordinate, we choose the hyperplanes such that we have at most $\varepsilon\tau n_0/(2d-1)$ points strictly in between consecutive hyperplanes, and at least $\varepsilon\tau n_0/(2d-1)$ points in between or on consecutive hyperplanes.

From these collections of hyperplanes we construct our test set as follows. Take any possible subset $H^*$ of $2d-2$ hyperplanes from $H_2(P) \cup \cdots \cup H_d(P)$ such that $H_2(P)$ up to $H_d(P)$ each contribute exactly two hyperplanes to $H^*$. Let $P(H^*)$ be the set of points in $P$ that lie on or between the hyperplanes contributed by $H_i(P)$, for all $2 \leqslant i \leqslant d$. Construct a collection $H_1(H^*)$ of hyperplanes orthogonal to the $x_1$-axis, such that there are $\varepsilon\tau n_0/(2d-1)$ points of $P(H^*)$ between each pair of consecutive hyperplanes. For each such hyperplane $h' \in H_1(H^*)$, construct a test box $b$ with the following properties:

1. $b$ is bounded by $h'$, the hyperplanes from $H^*$, and one additional hyperplane parallel to $h'$ and through a point of $P(H^*)$;
2. $b$ is a $((1-\varepsilon)\tau n_0)$-box.

Of all the test boxes thus constructed, we discard those that do not intersect $h$. Hence we will only keep boxes for which $h'$ is relatively close to $h$: there cannot be more than $(1-\varepsilon)\tau n_0$ points from $P(H^*)$ between $h$ and $h'$.

This implies that the total number of test boxes we create in this step is bounded by $(1-\varepsilon)\tau n_0 \;/\; (\varepsilon\tau n_0/(2d-1)) \leqslant (2d-1)/\varepsilon$ for a fixed set $H^*$. Hence, we create at most $(n(2d-1)/(\varepsilon\tau n_0))^{2d-2} \cdot (2d-1)/\varepsilon$ boxes in total. The number $T(n)$ of boxes created in the entire recursive procedure therefore satisfies:

$$\begin{array}{ll} T(n) = 0 & \text{if } n < \tau n_0 \\ T(n) \leqslant 2T(n/2) + \left(\frac{2d-1}{\varepsilon\tau n_0}\right)^{2d-2} \cdot \frac{2d-1}{\varepsilon} \cdot n^{2d-2} & \text{otherwise.} \end{array}$$

This leads to $|T| = T(n_0) = O(1/(\varepsilon^{2d-1}\tau^{2d-2}))$.

We now argue that $T$ is an $\varepsilon$-approximate $\tau$-test set for $P$. By construction, every box in $T$ contains at least $(1-\varepsilon)\tau n_0$ points, so it remains to show that every box $Q$ that contains at least $\tau n_0$ points from $P$ fully contains at least one box $b$ from $T$. Let $h$ be the first hyperplane used in the recursive construction. If at least $\tau n_0$ points in $Q$ lie to the same side of $h$, we can assume that there is a test box contained in $Q$ by induction. If this is not the case, we will show that a test box $b$ inside $Q$ was created for queries intersecting $h$. To see that such a box must exist, observe that for any $i$ with $2 \leqslant i \leqslant d$, there must be a hyperplane $h_i \in H_i(P)$ that intersects $Q$ and has at most $\varepsilon\tau n_0/(2d-1)$ points from $Q \cap P$ below it. Similarly, there is a hyperplane $h'_i \in H_i(P)$ intersecting $Q$ with at most $\varepsilon\tau n_0/(2d-1)$ points from $Q \cap P$ above it. Note that $h_i \neq h'_i$. Let $H^*$ be the set $\{h_2, h'_2, h_3, h'_3, \ldots, h_d, h'_d\}$. Since each of these hyperplanes 'splits off' at most $\varepsilon\tau n_0/(2d-1)$ points from $Q$, they define, together with the facets of $Q$ orthogonal to the $x_1$-axis, a box contained in $Q$ and containing at least $(1-\varepsilon+\varepsilon/(2d-1))\tau n_0$ points. From this, it follows that our construction, when processing this particular $H^*$, must have produced a test box $b \subset Q$. The proof is illustrated in Fig. 9.

To prove the lower bound, recall the construction used in Lemma 2.1 for the lower bound for the exact case. There we used $d$ staircases of $n/d$ points each. We then picked two points from each staircase, with at most $(\tau n - 1)/(d-1)$ points between (and including) them, except for the last staircase, where we picked only one point. Each such combination of points defined a different $(\tau n)$-box, thus given $\Omega(\tau^{d-1}n^{2d-1})$ different $(\tau n)$-boxes. Now, for the approximate case, we consider a subset of $(n/d)/(\varepsilon\tau n+2)$ so-called *anchor points* along each staircase, such that two consecutive anchor points have $\varepsilon\tau n+1$ points in between. We now pick two anchor points from each staircase, except the last staircase, where we pick one. We make sure that in between two chosen anchor points from the same staircase, there

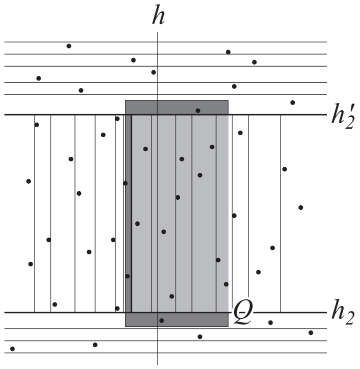

Figure 9: An example query range $Q$ (shaded area) that intersects $h$, showing also $h_2$, $h_2'$ and the grid $H_1(\{h_2, h_2'\})$. The three dark areas of $Q$ each contain at most $\varepsilon\tau n_0/3$ points. Hence, if $Q$ contains at least $\tau n_0$ points, the bright area of $Q$ contains at least $(1-\varepsilon)\tau n_0$ points, and a test box like the one shown above, bounded by $h_2$, $h_2'$ and a grid line from $H_1(\{h_2, h_2'\})$, must lie inside $Q$.

are at most $(\tau n - 1)/(d-1)$ points. We then pick a final point on the last staircase to obtain a $(\tau n)$-box. Each of these boxes must be captured by a different test box, because the intersection of two such boxes contains less than $(1-\varepsilon)\tau n$ points. The lower bound follows. □

**Putting it all together.** To summarize, the construction of our data structure for $\varepsilon$-approximate significant-presence queries on $S = S_1 \cup \cdots \cup S_m$ is as follows. We construct an $\varepsilon$-approximate $\tau$-test set $T_i$ for each color class $S_i$. This gives us a collection of $M = O(m/(\varepsilon^{2d-1}\tau^{2d-2}))$ boxes in $\mathbb{R}^d$. We map these boxes to a set $\hat{S}$ of colored points in $\mathbb{R}^{2d}$, and construct a data structure for the standard colored range-searching problem (that is, presence queries) on $P$, using the techniques of Agarwal *et al.* [3]. Their structure was designed for searching on a grid, but using the standard trick of normalization—replace every coordinate by its rank, and transform the query box to a box in this new search space in $O(\log n)$ time before running the query algorithm—we can employ their results in our setting.

The same technique works for exact queries, if we use exact test sets. This gives a good result for $d = 1$, if we use the results from Gupta *et al.* [10] on quadrant range searching.

**Theorem 2.1** *Let $S = S_1 \cup \cdots \cup S_m$ be a colored point set in $\mathbb{R}^d$, and $\tau$ a fixed constant with $0 < \tau < 1$. For $d = 1$, there is a data structure that uses $O(n)$ storage such that exact $\tau$-significant-presence queries can be answered in $O(\log n + k)$ time, where $k$ is the number of reported colors. For $d > 1$, there is, for any $\varepsilon$ with $0 < \varepsilon < 1/2$ and any $\delta > 0$, a data structure for $S$ that uses $O(M^{1+\delta})$ storage such that $\varepsilon$-approximate $\tau$-significant-presence queries on $S$ can be answered in $O(\log n + k)$ time, where $M = O(m/(\varepsilon^{2d-1}\tau^{2d-2}))$.*

**Remark 2.3** Observe that, since we only have constantly many points per color, we could also use standard range-searching techniques. But this would increase the term $k$ in the reporting time to $O(k/(\varepsilon^{2d-1}\tau^{2d-2}))$, which is undesirable.

**The case of variable $\tau$.** Now consider the case where the parameter $\tau$ is not given in advance, but is part of the query. We assume that we have a lower bound $\tau_0$ on the value of $\tau$ in any query. Then we can still answer queries efficiently, at only a small increase in storage. To do so, we build a collection of $O(T)$ substructures, where $T = \log(1/\tau_0)/\log(1+\varepsilon/2)$. More precisely, for integers $i$ with $0 \leqslant i \leqslant T$, we define $\tau_i := (1+\varepsilon/2)^i\tau_0$, and for each such $i$ we build a data structure for $(\varepsilon/2)$-approximate $\tau_i$-significant-presence queries on $S$. To answer a query with a query box $Q$ and query parameter $\tau$, we first find the largest $\tau_i$ smaller than or equal to $\tau$, and we query with $Q$ in the corresponding data structure. This leads to the following result.

**Theorem 2.2** *Let $S = S_1 \cup \cdots \cup S_m$ be a colored point set in $\mathbb{R}^d$, and $\tau_0$ a fixed constant with $0 < \tau_0 < 1$. For $d > 1$, any $0 < \varepsilon < 1/2$ and any $\delta > 0$, there is a data structure for $S$ that uses $O(M^{1+\delta}/\varepsilon)$ storage such that, for any $\tau \geqslant \tau_0$, one can answer $\varepsilon$-approximate $\tau$-significant-presence queries on $S$ in $O(\log n + k)$ time, where $M = O(m/(\varepsilon^{2d-1}\tau_0^{2d-2}))$ and $k$ is the number of reported colors.*

*Proof.* By Theorem 2.1, the size of substructure $i$ is $O(M^{1+\delta}(\tau_0/\tau_i)^D) = O(M^{1+\delta}/(1+\varepsilon/2)^{Di})$, where $M = O(m/(\varepsilon^{2d-1}\tau_0^{2d-2}))$ and $D = (2d-2)(1+\delta)$. The total size of all substructures is therefore $O(M^{1+\delta}\sum_{i=0}^{T}(1+\varepsilon/2)^{-Di}) = O(M^{1+\delta}/\varepsilon)$.

It remains to show that queries are answered correctly. Note that $\tau_i \leqslant \tau \leqslant (1+\varepsilon/2)\tau_i$. Now, any color $j$ with $|Q \cap S_j| \geqslant \tau_i|S_j|$ will be reported by our algorithm, so certainly any color with $|Q \cap S_j| \geqslant \tau|S_j|$ will be reported. Second, for any reported color $j$ we have:

$$\begin{aligned} |Q \cap S_j| &\geqslant (1-\varepsilon/2)\cdot\tau_i|S_j| \\ &\geqslant (1-\varepsilon/2)\cdot\tau/(1+\varepsilon/2)\cdot|S_j| \\ &\geqslant (1-\varepsilon)\tau\cdot|S_j|. \end{aligned}$$

This proves the correctness of the algorithm. □

## 3 Stabbing queries

Let $B = B_1 \cup \cdots \cup B_m$ be a set of $n$ colored boxes in $\mathbb{R}^d$, where $B_i$ denotes the subset of boxes of color $i$. Let $\tau$ be a constant with $0 < \tau < 1$. For a point $q$, we use $B_i(q)$ to denote the subset of boxes from $B_i$ that contain $q$. We want to preprocess $B$ for the following type of stabbing queries: given a query point $q$, report all colors $i$ such that $|B_i(q)| \geqslant \tau \cdot |B_i|$. As was the case for range queries, we are not able to obtain near-linear storage for exact queries for $d > 1$, so we focus on the $\varepsilon$-approximate variant, where we are also allowed to report a color if $|B_i(q)| \geqslant (1-\varepsilon)\tau \cdot |B_i|$.

Our approach is similar to our approach for range searching. Thus we define an *$\varepsilon$-approximate $\tau$-test set* for a set $B_i$ to be a set $T_i$ of test boxes such that

1. for any point $q$ with $|B_i(q)| \geqslant \tau \cdot |B_i|$, there is a test box $b$ with $q \in b$;

2. for any test box $b$ and any point $q \in b$, we have $|B_i(q)| \geqslant (1-\varepsilon)\tau \cdot |B_i|$.

This means we can answer a query by reporting all colors $i$ for which there is a test box $b \in T_i$ that contains $q$.

**Lemma 3.1** *For any set $B_i$ of boxes in $\mathbb{R}^d$, there is an $\varepsilon$-approximate $\tau$-test set $T_i$ consisting of $O(1/(\varepsilon\tau)^d)$ disjoint boxes. Moreover, for $\varepsilon < 1/(2d)$, there are sets of boxes in $\mathbb{R}^d$ for which any $\varepsilon$-approximate $\tau$-test set has size $\Omega(((1-\tau)/(\varepsilon\tau))^d)$.*

*Proof.* For each of the $d$ main axes, sort the facets of the input boxes orthogonal to that axis, and take a hyperplane through every $(\varepsilon\tau n_i/d)$-th facet, where $n_i := |B_i|$. This gives $d$

collections of $d/(\varepsilon\tau)$ parallel planes, which together define a grid with $O(1/(\varepsilon\tau)^d)$ cells. We let $T_i$ consist of all cells that are fully contained in at least $(1-\varepsilon)\tau \cdot |B_i|$ boxes from $B_i$. Clearly $T_i$ has the required number of boxes, and has property (2). (Note: using the fact that, coming from infinity, we must cross at least $d(1-\varepsilon)/\varepsilon \geqslant (1/\varepsilon) - 1$ hyperplanes before we can come to a cell from $T_i$, we can in fact obtain a slightly stronger bound on the size of $T_i$ for the case where $\tau$ is large.)

It remains to show that $T_i$ has property (1). Let $q$ be a point for which $|B_i(q)| \geqslant \tau \cdot |B_i|$, and let $C$ be the cell containing $q$. Since any cell is crossed by at most $\varepsilon\tau n_i$ facets, we must have $C \in T_i$.

The lower bound is proved as follows. For each of the main axes, take a collection of $(1-\tau)/(2d\varepsilon\tau)$ hyperplanes orthogonal to that axis. Slightly 'inflate' each hyperplane to obtain a very thin box. This way each intersection point of $d$ hyperplanes becomes a tiny hypercube. Next, each of these thin boxes is replaced by $2\varepsilon\tau n_i$ identical copies of itself. Note that each tiny hypercube is now covered by $2d\varepsilon\tau n_i$ boxes, and that there are $((1-\tau)/(2d\varepsilon\tau))^d$ such hypercubes. Add a collection of $(1-2d\varepsilon)\tau n_i$ big boxes, each containing all the tiny hypercubes. The tiny hypercubes are now covered by exactly $\tau n_i$ boxes, and the remaining space is covered by at most $(1-2\varepsilon)\tau n_i$ boxes. (Since we have used slightly less than $n_i$ boxes in total, we need to add some more boxes, at some arbitrary location disjoint from all other boxes.) Any test set must contain each of the hypercubes, and the result follows. □

To solve our problem, we construct a test set $T_i$ for each color class $B_i$ according to the lemma above. This gives us a collection of $M = O(m/(\varepsilon\tau)^d)$ colored boxes. Applying the results of Agarwal *et al.* [3] again, we get the following result.

**Theorem 3.1** *Let $B = B_1 \cup \cdots \cup B_m$ be a colored set of boxes in $\mathbb{R}^d$, and $\tau$ a fixed constant with $0 < \tau < 1$. For $d = 1$, there is a data structure that uses $O(n)$ storage such that exact $\tau$-significant-presence queries can be answered in $O(\log n + k)$ time, where $k$ is the number of reported colors. For $d > 1$, there is, for any $\varepsilon$ with $0 < \varepsilon < 1/2$ and any $\delta > 0$, a data structure for $B$ that uses $O(M^{1+\delta})$ storage such that $\varepsilon$-approximate $\tau$-significant-presence queries on $B$ can be answered in $O(\log n + k)$ time, where $M = O(m/(\varepsilon\tau)^d)$.*

**Remark 3.1** Note that, since the test boxes from any given color are disjoint, we can simply report the color of each box containing the query point $q$. Thus we do not have to use the structure of Agarwal *et al.*, but we can apply results from standard non-colored stabbing queries [5]. This way we can slightly reduce storage to $O(M \log^{d-2+\delta} M)$ at the cost of a slightly increased query time of $O(\log^{d-1} M + k)$. Also note that we can treat the case of variable $\tau$ in exactly the same way as for range queries.

## 4 Concluding remarks

Standard colored range searching problems ask to report all colors that have at least one object of that color intersecting the query range. We considered the variant where a color should only be reported if some constant pre-specified fraction of the objects intersects the range. We developed efficient data structures for an approximate version of this problem for orthogonal range searching queries and for stabbing queries. One obvious open problem is whether there exists a data structure for the exact problem with near-linear space. We have shown that this is impossible using our test-set approach, but perhaps a completely different approach is possible. Another open problem is to close the gap between our upper and lower bounds for the size of approximate test sets for orthogonal range searching. Finally, one can develop structures that can report the color that has the most points in the query range. Krizanc *et al.* [8] recently studied this problem for $d = 1$, and it would be interesting to generalize their results to $d \geqslant 2$.

**Acknowledgements.** We thank Joachim Gudmundsson and Jan Vahrenhold for inspiring discussions about the subject of this paper. Herman Haverkort's work is supported by the Netherlands' Organization for Scientific Research (NWO).

## References

[1] P.K. Agarwal. Range Searching. In: J. Goodman and J. O'Rourke (Eds.), *CRC Handbook of Computational Geometry*, CRC Press, pages 575–598, 1997.

[2] P.K. Agarwal and J. Erickson. Geometric range searching and its relatives. In: B. Chazelle, J. Goodman, and R. Pollack (Eds.), *Advances in Discrete and Computational Geometry*, Vol. 223 of *Contemporary Mathematics*, pages 1–56, American Mathematical Society, 1998.

[3] P.K. Agarwal, S. Govindarajan, and. S. Muthukrishnan. Range searching in categorical data: colored range searching on a grid. In *Proc. 10th Annu. European Sympos. Algorithms* (ESA 2002), pages 17–28, 2002.

[4] N. Alon, Z. Füredi, and M. Katchalski. Separating pairs of points by standard boxes. *European J. Combinatorics* 6:205–210 (1985).

[5] B. Chazelle. A functional approach to data structures and its use in multi-dimensional searching. *SIAM J. Comput.* 17: 427–462 (1988).

[6] T.K. Dey. Improved bounds for planar $k$-sets and related problems. *Discrete and Computational Geometry* 19(30):373–382 (1998).

[7] M. van Kreveld. New Results on Data Structures in Computational Geometry. PhD thesis, Utrecht University, 1992.

[8] D. Krizanc, P. Morin, and M. Smid. Range mode and range median queries on lists and trees. In *Proc. 14th Annu. Int. Sympos. on Algorithms and Computation (ISAAC 2003)*, 2003.

[9] J. Nievergelt and P. Widmayer. Spatial data structures: concepts and design choices. In: J.-R. Sack and J. Urrutia (Eds.) *Handbook of Computational Geometry*, pages 725–764, Elsevier Science Publishers, 2000.

[10] J. Gupta, R. Janardan, and M. Smid. Further results on generalized intersection searching problems: counting, reporting, and dynamization. *J. Algorithms* 19: 282–317 (1995).

[11] R. Janardan and M. Lopez. Generalized intersection searching problems. *Internat. J. Comput. Geom. Appl.* 3:39–70 (1993).

[12] M. Sharir, S. Smorodinsky, and G. Tardos. An Improved Bound for $k$-Sets in Three Dimensions. *Discrete and Computational Geometry* 26(2):195–204 (2001).