Boosting Spatial Pruning: On Optimal Pruning of MBRs

Tobias Emrich, Hans-Peter Kriegel, Peer Kröger, Matthias Renz, Andreas Züfle
Institute for Informatics, Ludwig-Maximilians-Universität München
Oettingenstr. 67, D-80538 München, Germany
{emrich,kriegel,kroeger,renz,zuefle}@dbs.ifi.lmu.de

ABSTRACT

Fast query processing of complex objects, e.g. spatial or uncertain objects, depends on efficient spatial pruning of the objects' approximations, which are typically minimum bounding rectangles (MBRs). In this paper, we propose a novel effective and efficient criterion to determine the spatial topology between multi-dimensional rectangles. Given three rectangles R, A, and B in a multi-dimensional space, the task is to determine whether A is definitely closer to R than B. This *domination* relation is used in many applications to perform spatial pruning. Traditional techniques apply spatial pruning based on minimal and maximal distance. These techniques however show significant deficiencies in terms of effectivity. We prove that our decision criterion is correct, complete, and efficient to compute even for high dimensional databases. In addition, we tackle the problem of computing the number of objects dominating an object o. The challenge here is to incorporate objects that only partially dominate o. In this work we will show how to detect such partial domination topology by using a modified version of our decision criterion. We propose strategies for conservatively and progressively estimating the total number of objects dominating an object. Our experiments show that the new pruning criterion, albeit very general and widely applicable, significantly outperforms current state-of-the-art pruning criteria.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval

General Terms

Performance

1. INTRODUCTION

Speeding-up queries using minimal bounding rectangles (MBRs) as object approximations is a common technique used in many different ways. For example, rectangles are used for data sets with spatially extended objects such as polygons or CAD models because operations on the exact object representation are usually much more expensive than on the object approximations. Furthermore, MBRs are used as spatial key for spatial access methods, e.g. the most prominent ones including the R-Tree [13], R*-Tree [2], X-Tree [3] as well as specialized adaptations like the TPR tree [23] and the U-Tree [20] among many others. In the last decade, MBR approximations have also become very popular for uncertain databases [4, 6, 7, 8, 18] in order to approximate all possible locations of an uncertain vector object such as a GPS signal.

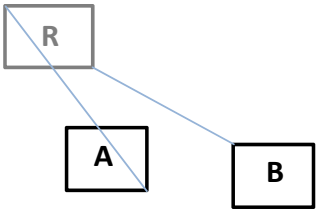


Figure 1: Spatial pruning on MBRs.

Rectangular approximations are commonly integrated into spatial pruning methods in order to speed-up spatial queries such as distance-range (ε-range) queries and k-nearest neighbor queries. Generally, current spatial pruning methods utilize the boundaries of regions, in particular of axis-aligned rectangles, in order to facilitate the pruning, i.e. to filter out true drops that do not match the query predicate. In this context, spatial pruning techniques are used for numerous application fields including searching in multi-dimensional vector spaces [12, 16, 19], similarity search in time series databases [17], query processing on spatio-temporal data [14, 22] and probabilistic query processing on uncertain data [4, 9, 20].

Most types of spatial/similarity queries used for the above mentioned applications, including k-nearest neighbor (kNN) queries, reverse k-nearest neighbor (RkNN) queries, and ranking queries, commonly require the following information. Given three point sets A, B, and R in a multi-dimensional space $\mathbb{R}^d$, e.g. representing MBRs, the task is to determine whether object A is definitely closer to R than B w.r.t. a distance function defined on the objects in $\mathbb{R}^d$. If this is the case, we say A *dominates* B w.r.t. R. An example of such a situation is depicted in Figure 1. In fact, we will focus on point sets that represent rectangles, e.g. minimum bounding rectangles (MBRs), because rectangles are the most preva-

SIGMOD'10, June 6–11, 2010, Indianapolis, Indiana, USA.

lent form of approximations for sets of points representing more complex objects as mentioned above. However, it should be noted that the concepts presented here can be easily extended to general point sets representing e.g. pixels of pictures, multi-represented objects, spatial objects [5], etc. The concept of domination is a central problem for most types of similarity queries including the ones mentioned above in order to identify true hits and true drops (pruning). For example, in case of a 1NN query around R, we can prune B if it is dominated by A w.r.t. R and for an R1NN query around R, we can prune B if A dominates R w.r.t. B.

The domination problem is trivial for point objects. However, applied to rectangles, the domination problem is much more difficult to solve. The problem is that the distance between two objects approximated by rectangles is no longer a single value but is represented by an interval. If two such distance intervals overlap, we cannot definitely detect whether one distance is smaller than the other. Traditionally, the minimal distance and maximal distance between rectangles are used to decide which object is closer to another object. For example, in a nearest-neighbor query we can prune all objects whose minimal distance to the query object exceeds the maximal distance between at least one other object and the query object. In fact, the traditional distance approximations based on minimal and maximal distance are not always suitable to determine the distance relationship between objects. An example is depicted in Figure 1 showing three objects A, B and R each approximated by rectangles. In order to decide whether object A is closer to R than object B, we cannot apply the minimal/maximal distances because the minimal distance between B and R is smaller than the maximal distance between A and R. Here, the problem is that when comparing the maximal distance between A and R with the minimal distance between B and R we take two different positions of the object R into account. For the maximal distance between A and R, we assume that the object R is located at the upper left corner of its rectangle approximation. For the minimum distance between B and R we assume that the object R is located at the lower right corner of its rectangle approximation. However, since an object approximated by a rectangle cannot be located at different positions at the same time, the two distances between A and R and between B and R depend on each other. To the best of our knowledge, none of the existing work, except approaches for reverse k-NN queries [21, 11], take this dependency into account. In our example, in fact, it can be detected that object A is closer to R than to object B when taking the above mentioned conditions into account.

For many spatial query applications, it even does not suffice to determine whether an object is dominated by another object. For example, in order to detect whether an object A approximated by a rectangle belongs to the kNNs of a query object Q ($k > 1$), we have to determine the number of objects that dominate A w.r.t. Q. In such a case, we have to tackle the general problem of computing the number of objects dominating a given object A w.r.t. a given object R which we call *domination count* of A w.r.t. R. The challenge here is to incorporate also sets of objects where each of the elements does not dominate A if considered separately, but the entire set dominates A if considered as a compound object. We say that the single objects of such a set dominate A only *partially*, while the set dominates A in the same sense discussed above for single objects.

In this paper, we propose a new decision criterion for the domination problem that can be used in all of the above sketched applications and in any algorithm designed for the above mentioned query types. In addition, it is the basis for our novel method to determine conservative and progressive bounds for the domination count of an object efficiently. In particular, we claim the following contributions.

- We discuss current state-of-the-art decision criteria for the domination problem among rectangles focussing on their correctness, completeness, and efficiency.
- We propose a novel decision criterion for the domination problem among rectangles that is correct, complete, and can be efficiently computed.
- We propose a number of heuristics that can be used to estimate the domination count in consideration of partial domination.
- We show how our domination decision criterion and our heuristics to determine the domination count can be used to improve spatial pruning strategies for a variety of spatial query processing methods.
- We present extensive experiments to evaluate our new pruning criteria in comparison to state-of-the-art approaches.

The remainder of this paper is organized as follows: Section 2 introduces our novel domination decision criterion. An effective estimation of the domination count is proposed in Section 3. The applicability of our concepts for spatial query problems are discussed in Section 4. Section 5 presents experimental results and Section 6 finally concludes the paper.

2. DETERMINING DOMINATION

2.1 The Problem of Domination Decision

Let $\mathcal{D} \subseteq \mathbb{R}^d$ be a database of d-dimensional points and *dist* be a distance function on objects in $\mathbb{R}^d$. In this paper we will focus on the L_p norms as the most commonly used family of distance functions in the area of similarity search. Intuitively, our problem is the following. Given the point sets $A, B, R \subseteq \mathcal{D}$, we want to decide if A "is definitely closer to" R than B to R w.r.t. the distance function *dist*. If this is the case, we say A *dominates* B w.r.t. R. In fact, we will focus on points sets that represent rectangles, e.g. minimum bounding rectangles (MBRs) because rectangles are the most prevalent form of approximations for sets of points representing more complex objects like page regions of directory nodes in spatial index structures, polygons, time series, uncertain objects, etc. (see above).

DEFINITION 1 (DOMINATION). *Let $A, B, R \subseteq \mathbb{R}^d$ be rectangles. The rectangle A dominates B w.r.t. R iff for all points $r \in R$ it holds that every point $a \in A$ is closer to r than any point $b \in B$, i.e.*

$$Dom(A, B, R) \Leftrightarrow \forall r \in R, \forall a \in A, \forall b \in B : dist(a, r) < dist(b, r) \quad (1)$$

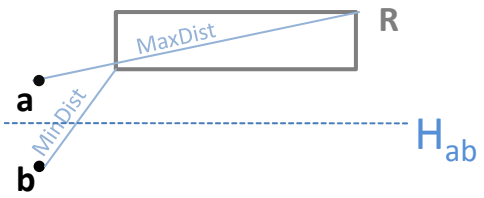


Figure 2: MBR pruning example

To determine $Dom(A, B, R)$, Equation 1 is not very helpful because a rectangle contains an infinite number of points in $\mathbb{R}^d$ and it is simply not computable to test all triples $a \in A$, $b \in B$ and $r \in R$. Rather, a domination decision criterion $DDC(A, B, R)$ for the single domination relation is required, which should fulfill the following properties:

- **Correctness:** if $DDC(A, B, R)$ returns *true* then A dominates B w.r.t. R, i.e.
$$DDC(A, B, R) \Rightarrow Dom(A, B, R).$$
- **Completeness:** if $DDC(A, B, R)$ returns *false* then A does not dominate B w.r.t. R, i.e.
$$\neg DDC(A, B, R) \Rightarrow \neg Dom(A, B, R).$$
- **Efficiency:** $DDC(A, B, R)$ can be evaluated efficiently.

2.2 Existing Domination Decision Criteria

In the following, $X_i = [X_i^{min}, X_i^{max}]$ represents the interval of the rectangle X in dimension i, $X_i^{mid} = 1/2 \cdot (X_i^{min} + X_i^{max})$ is the mean of interval X_i, and x_i denotes the value of point x in dimension i $(1 \leq i \leq d)$.

The Min-/MaxDist decision criterion. Probably the most well-known decision criterion for the domination problem among rectangles used in many database applications is based on two well known metrics defined on rectangles [19]. The minimum distance $MinDist(A, B)$ between two rectangles A and B always underestimates the distance of point pairs $(a, b) \in A \times B$ and is defined as

$$MinDist(A, B) = \sqrt[p]{\sum_{i=1}^{d} \begin{cases} |A_i^{min} - B_i^{max}|^p, \text{ if } A_i^{min} > B_i^{max} \\ |B_i^{min} - A_i^{max}|^p, \text{ if } B_i^{min} > A_i^{max} \\ 0 \qquad\qquad\quad , \text{ else} \end{cases}} \tag{2}$$

The maximum distance $MaxDist(A, B)$ between two rectangles A and B always overestimates the distances of all point pairs $(a, b) \in A \times B$ and is defined as:

$$MaxDist(A, B) = \sqrt[p]{\sum_{i=1}^{d} \begin{cases} |A_i^{max} - B_i^{min}|^p, \text{ if} A_i^{mid} \geq B_i^{mid} \\ |B_i^{max} - A_i^{min}|^p, \text{ if} B_i^{mid} > A_i^{mid} \end{cases}} \tag{3}$$

Definition 2 (Min-/MaxDist criterion). *Let $A, B, R \in \mathbb{R}^d$ be rectangles. The* Min-/MaxDist domination decision criterion *is defined as*

$$DDC_{MinMax}(A, B, R) \Leftrightarrow MaxDist(A, R) < MinDist(B, R).$$

Lemma 1. *The Min-/MaxDist decision criterion is correct, i.e. $DDC_{MinMax}(A, B, R) \Rightarrow Dom(A, B, R)$.*

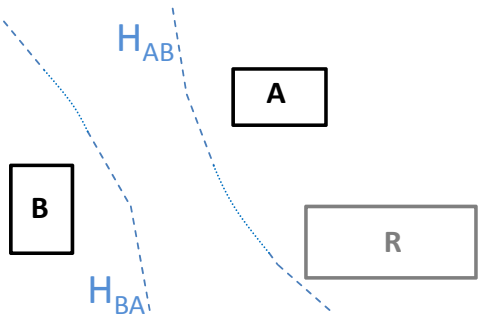


Figure 3: Voronoi-based decision criterion on MBRs

Proof. The following holds due to the conservative properties of *MinDist* and *MaxDist*:
$DDC_{MinMax}(A, B, R) \Leftrightarrow MaxDist(A, R) < MinDist(B, R)$
$\Rightarrow \forall a \in A, \forall r \in R, \forall b \in B : dist(a, r) \leq MaxDist(A, R) < MinDist(B, R) \leq dist(b, r) \Leftrightarrow Dom(A, B, R)$. □

Lemma 2. *The Min-/MaxDist decision criterion is not complete, i.e. $\neg DDC_{MinMax}(A, B, R) \not\Rightarrow \neg Dom(A, B, R)$.*

Proof. Figure 2 shows an example for the 2D space where $DDC_{MinMax}(A, B, R)$ is false although $Dom(A, B, R)$ holds. In the examples, $A = a$ and $B = b$ are rectangles with zero extension, i.e. points. Clearly, $MaxDist(a, R) < MinDist(b, R)$ is not satisfied, i.e. $DDC_{MinMax}(a, b, R)$ is false. The Voronoi line H_{ab} between a and b, i.e. the line containing all points that have equal distance to a and b, which is the dashed line in Figure 2 divides the 2D space into two half spaces. It is obvious that all points above that line (located in the half space containing a) have a distance to a that is smaller than the distance to b. Thus, according to Definition 2.1, a dominates b w.r.t. all objects which lie completely above H_{ab}. As a consequence, $Dom(a, b, R)$ holds. □

Let us note that the Min-/MaxDist domination decision criterion is complete for two arbitrary rectangles A and B if R is a point, i.e. R has no extension in all dimensions. In addition, the Min-/MaxDist domination decision criterion can be computed efficiently in $O(d)$ time since the calculation of *MinDist* and *MaxDist* is linear in d.

Voronoi-based decision criterion. The Voronoi plane H_{ab} between two points a and b that has been used in the proof of Lemma 2 is used in [21] as a different decision criterion for points. In a d-dimensional space $H_{ab} = \{x \in \mathbb{R}^d \mid dist(a, x) = dist(b, x)\}$ is a $(d-1)$-dimensional hyperplane containing all points having equal distance to a and to b. It divides the space into two half-spaces $H_{ab}(a)$ containing a and $H_{ab}(b)$ containing b. If a rectangle R lies completely within one of these half-spaces, then R is closer to the respective point in the same half-space. In the example of Figure 2, R is in the half-space $H_{ab}(a)$, thus all $r \in R$ are closer to a than to b. A Voronoi hyperplane between a point and a rectangle has been proposed in [11]. For the general case of two rectangles, we need to construct the Voronoi plane H_{AB} between two rectangles A and B which is the intersection of all Voronoi half-spaces between all pairs of points of the corresponding rectangles and can be defined as $H_{AB} = \{x \in \mathbb{R}^d \mid MinDist(x, B) = MaxDist(x, A)\}$, see [10]. An example of a Voronoi plane between two rectangles A and B is H_{AB}, depicted in Figure 3. This Voronoi

plane is piecewise linear and curvilinear (cf. [10] for more details on the Voronoi plane between two rectangles). If a rectangle lies completely within the half-space $H_{AB}(A)$, then R is definitely closer to A. However, to determine the half-space containing all points that are definitely closer to B than to A, $H_{AB}(B)$ cannot be used and the Voronoi plane H_{BA} has to be computed. The reason is that unlike in the case of points, there exist points p for which neither $Dom(A, p, R)$ nor $Dom(B, p, R)$ is true. Intuitively, the Voronoi-based domination decision criterion states that A dominates B w.r.t. R if R is completely contained in the half-space $H_{AB}(A)$.

DEFINITION 3 (VORONOI-BASED CRITERION). *Let $A, B, R \in \mathbb{R}^d$ be rectangles. The Voronoi-based decision criterion is defined as*

$$DDC_{\mathbf{Voronoi}}(A, B, R) \Leftrightarrow R \subseteq H_{AB}(A).$$

LEMMA 3. *The Voronoi-based decision criterion is correct and complete, i.e. $DDC_{\mathbf{Voronoi}}(A, B, R) \Leftrightarrow Dom(A, B, R)$.*

PROOF. By definition of H_{AB} the statement holds: $DDC_{\mathbf{Voronoi}}(A, B, R) \Leftrightarrow R \subseteq H_{AB}(A) \Leftrightarrow \forall a \in A, b \in B$: $R \subseteq H_{ab}(a). \Leftrightarrow \forall a \in A, \forall b \in B, \forall r \in R : dist(a, r) < dist(b, r) \Leftrightarrow Dom(A, B, R)$. □

Computing any Voronoi plane between any $a \in A$ and $b \in B$ to obtain the curvilinear plane as depicted in Figure 3 is rather complex. To the best of our knowledge, there exists no efficient solution for this problem. However, it is clear that any such algorithm must scale exponentially in the dimensions, since even for the simple case where b is a point, the number of different pieces of the plane is equal to the number of corners of A which is in $O(2^d)$ (cf. [11] for a discussion on the computation of such Voronoi planes).

Corner-based decision criterion. The corner-based decision has recently been proposed as a pruning criterion for RkNN search of spatial objects in R^2 [10]. This approach exploits the property that the side $H_{AB}(A)$ of H_{AB} that is responsible for pruning is convex for RkNN queries. Thus, if a rectangle R is not fully contained in $H_{AB}(A)$ (i.e. R cannot be pruned), then at least one corner of R must be contained in $H_{AB}(B)$. Therefore, it is sufficient to consider only corners of MBRs. The Min-/MaxDist decision criterion, that is correct and complete in the case where only points are considered, is then applied to the corners. For more details on this decision criterion, refer to [10]. However, since this criterion requires to consider all 2^d corners of MBRs, the complexity must scale in $O(2^d)$.

Summary. Table 1 summarizes the discussion of existing decision criteria for the domination problem. It can be observed, that none of these approaches meets all the desired properties, i.e. either is not complete or suffers from exponential runtime. The fourth approach in Table 1 called "Optimal" is our new decision criterion which is described in the next section.

2.3 A Correct, Complete, and Linear-Time Domination Decision Criterion

We will derive a new decision criterion that is correct, complete, and can be computed in $O(d)$ time. Our novel domination decision criterion can be derived from the original definition of domination in Definition 1 by applying the following six equivalences.

Table 1: Overview decision criteria

Criterion	Correct	Complete	Efficient
$DDC_{\mathbf{MinMax}}$	YES	NO	YES: $O(d)$
$DDC_{\mathbf{Voronoi}}$	YES	YES	NO: $O(2^d)$
$DDC_{\mathbf{Corner}}$	YES	YES	NO: $O(2^d)$
$DDC_{Optimal}$	YES	YES	YES: $O(d)$

EQUIVALENCE 1.
$\forall a \in A, b \in B, r \in R : dist(a, r) < dist(b, r) \Leftrightarrow$
$\forall r \in R : MaxDist(A, r) < MinDist(B, r)$

PROOF.
(1) "⇒"
If the left-hand side holds for each $r \in R$ then it also holds for that $a \in A$ and $b \in B$ that maximize and minimize the distance to r, respectively. These points $a \in A$ and $b \in B$ obviously determine the values of *MaxDist* and *MinDist*, respectively.
(2) "⇐"
If the right-hand side holds for each $r \in R$ as well as for that $a \in A$ and $b \in B$ that maximizes and minimizes the distance to r, i.e. determines the value of *MaxDist* and *MinDist*, respectively, then it also holds for any $a \in A$ and any $b \in B$. □

EQUIVALENCE 2.
$\forall r \in R : MaxDist(A, r) < MinDist(B, r) \Leftrightarrow$
$\forall r \in R : \sqrt[p]{\sum_{i=1}^{d} MaxDist(A_i, r_i)^p} < \sqrt[p]{\sum_{i=1}^{d} MinDist(B_i, r_i)^p}$

PROOF. Follows directly from the definition of *MaxDist* and *MinDist* for L_p norms (see above). □

EQUIVALENCE 3.
$\forall r \in R : \sqrt[p]{\sum_{i=1}^{d} MaxDist(A_i, r_i)^p} < \sqrt[p]{\sum_{i=1}^{d} MinDist(B_i, r_i)^p}$
$\Leftrightarrow \forall r \in R : \sum_{i=1}^{d} (MaxDist(A_i, r_i)^p - MinDist(B_i, r_i)^p) < 0$

PROOF.
$\forall r \in R : \sqrt[p]{\sum_{i=1}^{d} MaxDist(A_i, r_i)^p} < \sqrt[p]{\sum_{i=1}^{d} MinDist(B_i, r_i)^p}$
$\Leftrightarrow \forall r \in R : \sum_{i=1}^{d} MaxDist(A_i, r_i)^p < \sum_{i=1}^{d} MinDist(B_i, r_i)^p$
$\Leftrightarrow \forall r \in R : \sum_{i=1}^{d} MaxDist(A_i, r_i)^p - \sum_{i=1}^{d} MinDist(B_i, r_i)^p < 0$
$\Leftrightarrow \forall r \in R : \sum_{i=1}^{d} (MaxDist(A_i, r_i)^p - MinDist(B_i, r_i)^p) < 0$ □

EQUIVALENCE 4.
$\forall r \in R : \sum_{i=1}^{d} (MaxDist(A_i, r_i)^p - MinDist(B_i, r_i)^p) < 0 \Leftrightarrow$
$\max_{r \in R}(\sum_{i=1}^{d} (MaxDist(A_i, r_i)^p - MinDist(B_i, r_i)^p)) < 0$

PROOF. Instead of considering all possible $r \in R$, it is sufficient to consider only that point $r' \in R$ which maximizes the left-hand side of the inequality. If the inequality holds for this point r', then it obviously holds for all possible $r \in R$ and vice versa. □

The next equivalence requires the following lemma:

LEMMA 4. *Let $F : \mathbb{R}^d \to \mathbb{R}$ be a function that is summed by treating each dimension independently, i.e. there exists a function $f : \mathbb{R} \to \mathbb{R}$ such that*

$$F(o) = \sum_{i=1}^{d} f(o_i)$$

Also, let $A \subseteq \mathbb{R}^d$ be a rectangle and

$$\sigma := \text{argmax}_{a \in A}(F(a))$$

be the object in A that maximizes F. Then, the following holds:

$$\max_{a\in A}(\sum_{i=1}^{d} f(a_i)) = \sum_{i=1}^{d} \max_{a_i\in A_i}(f(a_i))$$

PROOF.

$$\max_{a\in A}(\sum_{i=1}^{d} f(a_i)) \overset{\text{Def } F(a)}{=} \max_{a\in A}(F(a)) \overset{\text{Def } \sigma}{=} F(\sigma)$$

$$\overset{\text{Def } F(a)}{=} \sum_{i=1}^{d} f(\sigma_i) \overset{\text{Def } \sigma}{=} \sum_{i=1}^{d} \max_{a_i\in A_i}(f(a_i))$$

□

EQUIVALENCE 5.
$\max_{r\in R}(\sum_{i=1}^{d} MaxDist(A_i,r_i)^p - MinDist(B_i,r_i)^p) < 0 \Leftrightarrow$
$\sum_{i=1}^{d} \max_{r_i\in R_i}(MaxDist(A_i,r_i)^p - MinDist(B_i,r_i)^p) < 0$

PROOF. This follows from lemma 4 by substituting

$$F(r) = MaxDist(A,r) - MinDist(B,r)$$

□

The final equivalence (equivalence 6) makes the equation computable. It is based on the assumption that for finding the maximum r_i in dimension i, it is sufficient to consider the boundary points (R_i^{min} and R_i^{max}) of the interval R_i. This assumption is proven in the following two lemmas.

LEMMA 5. *Let A and B be intervals. The function $f : \mathbb{R} \to \mathbb{R}$ defined as $f(x) = MaxDist(A,x)^p - MinDist(B,x)^p$ has no local maximum.*

PROOF. A formal proof for this lemma can be found in the appendix. □

LEMMA 6. *Let $f : \mathbb{R} \to \mathbb{R}$ be a function that has no local maximum and $I = [I_{min}, I_{max}] \subset \mathbb{R}$ be an arbitrary finite interval. The value that maximizes f in the interval I must be either I_{min} or I_{max}, i.e.*

$$\underset{i\in I}{argmax}(f(i)) \in \{I_{min}, I_{max}\}$$

PROOF. Let $p \in [I_{start}, I_{end}]$ be the value that maximizes f in I, i.e. $p = \text{argmax}_{i\in I}(f(a))$. Then, $\forall i \in I : f(i) \leq f(p)$, in particular, $f(I_{min}) \leq f(p)$ and $f(I_{max}) \leq f(p)$. Note that $f(I_{min}) < p$ and $f(I_{max}) < p$ cannot both be true, because this would be a contradiction to the assumption that $f(x)$ has no local maximum. Thus it must either hold that $f(I_{start}) = f(p)$ or $f(I_{end}) = f(p)$, i.e. $I_{min} = \text{argmax}_{i\in I}(f(x))$ or $I_{max} = \text{argmax}_{i\in I}(f(x))$. □

Now we can derive the final equivalence.

EQUIVALENCE 6.
$\sum_{i=1}^{d} \max_{r_i\in R_i}(MaxDist(A_i,r_i)^p - MinDist(B_i,r_i)^p) < 0 \Leftrightarrow$
$\sum_{i=1}^{d} \max_{r_i\in\{R_i^{min},R_i^{max}\}}(MaxDist(A_i,r_i)^p - MinDist(B_i,r_i)^p) < 0$

PROOF. Follows from lemma 5 and 6. □

DEFINITION 4 (OPTIMAL DECISION CRITERION). *Our novel optimal domination decision criterion is defined as*

$DDC_{Optimal}(A,B,R) \Leftrightarrow$

$$\sum_{i=1}^{d} \max_{r_i\in\{R_i^{min},R_i^{max}\}}(MaxDist(A_i,r_i)^p - MinDist(B_i,r_i)^p) < 0$$

LEMMA 7. *The novel optimal domination decision criterion is correct and complete.*

PROOF. Correctness and completeness follow directly from equivalences 1 to 6. □

Obviously, the novel optimal domination decision criterion can be computed in $O(d)$ time and, thus, fulfills all three desired properties mentioned in Section 2.1.

3. DOMINATION COUNT COMPUTING

In most applications, testing the single domination relation of only two rectangles (w.r.t. a reference rectangle) is too basic. Rather, in the context of a set of rectangles $\mathcal{O} \subseteq \mathbb{R}^d$, the number of rectangles $A_i \in \mathcal{O}$ that dominate a given rectangle B w.r.t. R (referred to as *domination count*) is required. For example, a kNN query algorithm can use the information that at least k rectangles of $\mathcal{O}$ dominate rectangle $B \in \mathcal{O}$ w.r.t. a query rectangle R to identify B as true drop that can be pruned. The number of rectangles that dominate a given rectangle can analogously be used e.g. for RkNN queries and inverse ranking queries.

DEFINITION 5 (DOMINATION COUNT). *Let $B, R \subseteq \mathbb{R}^d$ be rectangles and $\mathcal{O}$ be a set of rectangles. The* domination count *of B w.r.t. R is defined by:*

$$DC(\mathcal{O},B,R) = \min_{r\in R}\{|A_i \in \mathcal{O} : MaxDist(A_i,r) < MinDist(B,r)|\}$$

Intuitively, if the domination count of B w.r.t. R is k, then for each point $r \in R$ there exist at least k rectangles $A_i \in \mathcal{O}$ which are closer to r than B.

Let us note that the domination count of B w.r.t. R cannot be computed by simply counting the number of rectangles that dominate B w.r.t. R by means of Definition 1 because this does not involve groups of rectangles that dominate R collectively, but not individually. An example of such a group of rectangles is shown in Figure 4. Neither rectangle A_1 nor rectangle A_2 dominates B w.r.t. R. However, B is dominated *partially* by A_1 and partially by A_2, respectively, i.e. it is dominated by A_1 and A_2 w.r.t. specific subregions of R.

However, when considering any point $r \in R$, rectangle B is dominated by at least one of the two rectangles A_1, A_2 w.r.t. r and, thus, B is dominated by the group $\mathcal{A} = \{A_1, A_2\}$ according to Definition 1.

In general, the problem of finding the subregion with the minimal domination count is hard. First, the computation of the intersection of a half-space and a hyper-polyhedron becomes increasingly complex [21] for increasing dimensionality. Secondly, the number of subregions grows very fast. To give a brief intuition of the possible number of subregions generated by a total of n objects, consider the case of axis parallel pruning regions. If $n \leq d$, then each object may split R in a different dimension, resulting in a total of 2^n subregions. For $n > d$, balanced splitting of dimensions results in at least $(1 + \lfloor \frac{n}{d} \rfloor^d)$ subregions. If d is assumed

to be constant, then $(1 + \lfloor \frac{n}{d} \rfloor^d) \in O(n^d)$. Thirdly, the resulting subregions can be complex d-dimensional polygons, particularly the subregions could have not only straight sides but also parabolic sides which makes computations involving these polygons very complex.

Though we are not able to compute the exact domination count of a given rectangle efficiently, we can try to find efficient solutions for approximating the domination count of a rectangle. In principal, in order to determine the domination count of B w.r.t. R we need to take the two constituting types of dominations into account: The first part is to count all objects A for which $Dom(A, B, R)$ holds. This number is called *basic domination count*. This can be done using e.g. $DDC_{Optimal}$. The second and more challenging part is to detect all minimal groups $\mathcal{A}$ that dominate B as a group but do not contain an element that already dominates B separately, i.e. each $A_i \in \mathcal{A}$ only partially dominate B. The consideration of this type of domination requires the concept of *partial domination* which will be introduced later on.

A simple lower bound of the domination count can be achieved by computing the basic domination count. Intuitively, the basic domination count simply counts the number of rectangles that (completely) dominate the rectangle B w.r.t. rectangle R, i.e. neglects groups of rectangles that only partially dominate B separately but completely dominate B as a group.

Definition 6 (Basic Domination Count). *Let $\mathcal{O} = \{A_1, ..., A_N\}$ be a set of d-dimensional rectangles and let $B, R \subseteq \mathbb{R}^d$ be two rectangles. The* basic domination count *of B w.r.t. R is the number of objects in $\mathcal{O}$ that dominate B w.r.t. R, formally:*

$$DC_{basic}(\mathcal{O}, B, R) = |\{A_i \in \mathcal{O} \mid Dom(A_i, B, R)\}|.$$

Using our novel domination decision criterion $DDC_{Optimal}$, the basic domination count DC_{basic} can be computed in $O(N \cdot d)$. This is worth noting since existing decision criteria only allow either to compute the exact DC_{basic} value in exponential time or to compute an approximation of the DC_{basic} value in linear time. In the latter case, we would obtain a lower bound of DC_{basic} which makes the lower bounding estimation of the domination count even more loose.

As discussed above, the domination count also takes into account all sets of rectangles that increase the domination count of a rectangle as a group and that do not contain any element that does so separately. Therefore, we need the concept of *partial domination*. In the remainder of this section, we will first formalize the concept of partial domination. In particular we will discuss how our novel domination decision criterion $DDC_{Optimal}$ can be used for (i) detecting partial domination and (ii) deriving a conservative approximation of the domination count.

3.1 Partial Domination

The concept of *partial domination* (cf. Figure 4) was first introduced in [11] (the authors used the term "partial pruning") for boosting $RkNN$ queries in the 2D space. It can be applied to any other similarity query type analogously.

Definition 7 (partial domination). *Let $A, B, R \subseteq \mathbb{R}^d$ be rectangles. A* dominates *B* partially *w.r.t. R, denoted by $PDom(A, B, R)$ if A dominates B for some, but not all $r \in R$, i.e.*

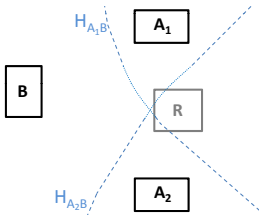


Figure 4: Partial Domination example for an RNN-query

$$PDom(A, B, R) \Leftrightarrow$$

$$\neg(\forall a \in A, b \in B, r \in R : dist(b, r) > dist(a, r)) \quad (4)$$

$$\bigwedge$$

$$\exists r \in R : \forall a \in A, b \in B : dist(b, r) > dist(a, r) \quad (5)$$

Inequality 4 holds if A does not dominate B w.r.t. all points $r \in R$. Note that Inequality 4 is simply the negation of $Dom(A, B, R)$ and can also be computed in $O(d)$ using our novel decision criterion $DCC_{Optimal}$. Inequality 5 is only satisfied if there exists an $r \in R$ for which B is dominated by A.

Obviously, the sets of objects that dominate B as a group can only contain rectangles A_i that partially dominate B, i.e. for which $PDom(A_i, B, R)$ holds. In other words, for the computation of the second part of the domination count of a rectangle B, we could use the detection of partial domination as a first step because only those rectangles A_i for which $PDom(A_i, B, R)$ holds could be the elements of those set of rectangles that dominate B as a group.

Partial domination can efficiently be detected by applying the following six equivalences analogously to Section 2.3. We start with inequality 5.

Equivalence 7.

$$\exists r \in R : \forall a \in A, b \in B \quad : \quad dist(b, r) > dist(a, r)$$
$$\Leftrightarrow \exists r \in R \quad : \quad MaxDist(A, r) < MinDist(B, r)$$

Proof. This proof is analogous to the proof of Equivalence 1, i.e. it exploits that the $DDC_{\mathbf{MinMax}}$, decision criterion is optimal in the case where R is a point. □

Equivalence 8.

$$\exists r \in R : MaxDist(A, r) < MinDist(B, r) \Leftrightarrow$$
$$\exists r \in R : \sqrt[p]{\sum_{i=1}^{d} MaxDist(A_i, r_i)^p} < \sqrt[p]{\sum_{i=1}^{d} MinDist(B_i, r_i)^p}$$

Proof. Follows directly from the definition of $MaxDist$ and $MinDist$ for L_p norms. □

Equivalence 9.

$$\exists r \in R : \sqrt[p]{\sum_{i=1}^{d} MaxDist(A_i, r_i)^p} < \sqrt[p]{\sum_{i=1}^{d} MinDist(B_i, r_i)^p}$$
$$\Leftrightarrow \exists r \in R : \sum_{i=1}^{d} MaxDist(A_i, r_i)^p - MinDist(B_i, r_i)^p < 0$$

Proof.

$$\exists r \in R : \sqrt[p]{\sum_{i=1}^{d} MaxDist(A_i, r_i)^p} < \sqrt[p]{\sum_{i=1}^{d} MinDist(B_i, r_i)^p}$$
$$\Leftrightarrow \exists r \in R : \sum_{i=1}^{d} MaxDist(A_i, r_i)^p < \sum_{i=1}^{d} MinDist(B_i, r_i)^p$$
$$\Leftrightarrow \exists r \in R : \sum_{i=1}^{d} MaxDist(A_i, r_i)^p - \sum_{i=1}^{d} MinDist(B_i, r_i)^p < 0$$
$$\Leftrightarrow \exists r \in R : \sum_{i=1}^{d} MaxDist(A_i, r_i)^p - MinDist(B_i, r_i)^p < 0 \quad \square$$

EQUIVALENCE 10.
$\exists r \in R : \sum_{i=1}^{d} MaxDist(A_i, r_i)^p - MinDist(B_i, r_i)^p < 0 \Leftrightarrow$
$MIN_{r \in R}(\sum_{i=1}^{d} MaxDist(A_i, r_i)^p - MinDist(B_i, r_i)^p) < 0$

PROOF. The rationale for equivalence 10 is that if there exists an $r \in R$ for which the left-hand side returns less than 0, then this also holds for the r which minimizes the term on the right-hand side and vice versa. □

EQUIVALENCE 11.
$\min_{r \in R}(\sum_{i=1}^{d} MaxDist(A_i, r_i)^p - MinDist(B_i, r_i)^p) < 0 \Leftrightarrow$
$\sum_{i=1}^{d} \min_{r_i \in R_i}(MaxDist(A_i, r_i)^p - MinDist(B_i, r_i)^p) < 0$

PROOF. This proof is analogous to the proof of Equation 5 using minimization instead of maximization. □

Analogously to Equivalence 6, the last equivalence below makes the equation computable. Again, we need two lemmas.

LEMMA 8. *Let D be a one dimensional vector database using L_p-Norm. Let A and B be intervals. The function $f : \mathbb{R} \to \mathbb{R}$:*

$$f(x) = maxDist(A, x)^p - minDist(B, x)^p$$

may have a local minimum only at A.mean.

PROOF. This proof is contained in the formal proof of lemma 5 in the appendix. □

LEMMA 9. *Let $f : \mathbb{R} \to \mathbb{R}$ be a function that has at most one local minimum at x. For any finite interval $I \subset \mathbb{R} = [I_{start}, I_{end}]$ the following holds:*

$$\underset{i \in I}{argmin}(f(i)) \in \{I_{start}, I_{end}, x\}$$

That is, the point of the interval I that minimizes $f(x)$ must be either the lower or the upper bound of I, or the local minimum x.

PROOF. The proof is similar to the proof of Lemma 6 and thus omitted here. □

In consideration of the above lemmas we now derive the final equivalence:

EQUIVALENCE 12.
$\sum_{i=1}^{d} \min_{r_i \in R_i}(MaxDist(A_i, r_i)^p - MinDist(B_i, r_i)^p) < 0 \Leftrightarrow$
$\sum_{i=1}^{d} min_{r_i}(MaxDist(A_i, r_i)^p - MinDist(B_i, r_i)^p) < 0,$
$\quad where\ r_i \in \{R_i^{min}, R_i^{max}, A_i^{mid}\}$

PROOF. Directly follows from Lemma 8 and Lemma 9. □

Thus, using the formula in Equivalence 12 we can efficiently detect all partial dominations. However, as indicated above, this is only the first step towards computing the domination count. In fact, we need to determine that subregion of the reference rectangle R, for which the domination count is minimal. Since we cannot test all possible points $r \in R$ (see also the discussion above), we propose three heuristics to conservatively approximate the domination count of a rectangle.

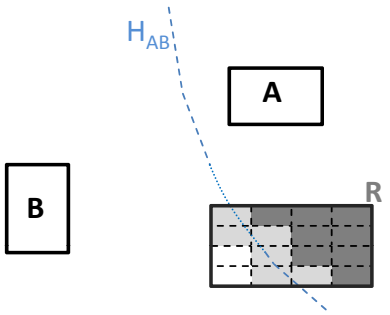


Figure 5: Partial domination using grid partitioning

3.2 Domination Count Estimation

Using the techniques proposed in Sections 2 and 3.1 we can check if an MBR A dominates B completely or partially w.r.t. R. These tests are generally applicable as long as the involved objects are MBRs. For calculating the domination count of B it is therefore possible to split R into smaller MBRs and then calculate the domination count for each cell individually. The following three heuristics use different approaches for splitting R to estimate the domination count.

3.2.1 *Domination Count Estimation based on grid partitioning*

A straight forward approach for splitting R is performed by using a grid with a fixed number m of partitions in each dimension. Considering the example in Figure 5, we can (using the decision criteria for domination and partial domination) assert that A dominates B w.r.t. all dark gray cells and partially dominates B w.r.t. all light gray cells of R. For the rest of the cells (white) A does not dominate B. Using this grid partitioning, the domination count ($DC(\mathcal{O}, B, R)$) can be estimated by the minimum domination count of all cells $c_i \in R$, that is:

$$DC_{grid}(\mathcal{O}, B, R) = min_i(DC_{basic}(\mathcal{O}, B, c_i))$$

This estimation is valid as we know that B is dominated by at least this amount of $A_i \subset \mathcal{O}$ w.r.t. each cell $c_i \in R$.

An example for the grid based partial pruning is given in Figure 6. Here an MBR R is partitioned into 16 cells. In addition two Voronoi hyperplanes H_{A_1B} and H_{A_2B} are shown. The objects $\mathcal{O} = \{A_1, A_2\}$ and B generating the hyperplanes are ommited here. For the area on the right-hand side of H_{A_1B}, object B is dominated by object A_1 and for the left-hand side of H_{A_2B}, B is dominated by A_2. It is clear that neither A_1 nor A_2 (fully-) dominate B with respect to the whole MBR R. For each cell the conservative domination count $DC_{basic}(\mathcal{O}, B, c_i)$ is shown. With respect to dark cells, A_1 and A_2 dominate B and thus the cells have a value of 2. With respect to light cells, only one of the two objects dominates B, therefore they get marked with a value of 1. By taking the minimum value of all cells $c_i \in R$ we obtain $DC_{grid}(\mathcal{O}, B, R) = 1$. The advantage of this approach is, that it returns a very accurate estimation of the domination count while avoiding expensive materialization of the Voronoi hyperplanes. The accuracy can be boosted by increasing the number of splits per dimension. In return increasing m will highly increase the runtime of the algorithm, as the number of cells $c_i \in R$ is m^d. This implies that this approach is not applicable for high dimensions. For

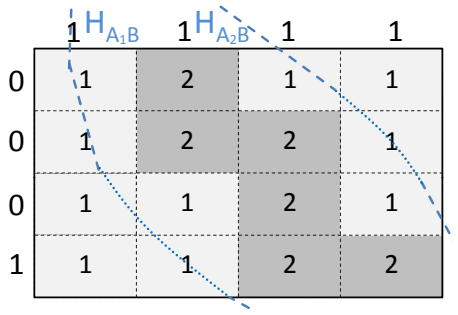


Figure 6: Domination Count estimation using grid partitioning.

each cell c_i, $DC_{basic}(\mathcal{O}, B, c_i)$ can be computed in a single scan of the objects for which $PDom(A_i, B, R)$ holds using the $DDC_{Optimal}$ (c.f. Definition 4). Thus the total time complexity is in $O(d \cdot |\mathcal{O}| \cdot m^d)$.

3.2.2 Domination Count Estimation based on slices

In order to reduce the runtime of the domination count estimation, we propose a second algorithm, which is not based on a grid partitioning. Instead of cells, this approach considers *slices*. Therefore an MBR R is split into m *slices* s_i^{dim} in each of the d-dimensions ($1 \leq dim \leq d$). This results in $d \cdot m$ overlapping *slices*. The domination count $DC(\mathcal{O}, B, R)$ can then be approximated by computing, for each dimension, the minimal domination count of all slices and using the result of the dimension maximizing this estimation.

$$DC_{slice}(\mathcal{O}, B, R) = max_{dim}(min_i(DC_{Basic}(\mathcal{O}, B, s_i^{dim})))$$

For example, the domination count $DC(\mathcal{O}, B, s_i)$ for each slice s_i (i.e. each row and each column) and each cell c_i is shown in Figure 6 for a 2 dimensional MBR R. The minimal domination count considering all rows is 0, while the minimal domination count w.r.t. all columns is 1. Thus $DC_{slice}(\mathcal{O}, B, s_i) = 1$ in this example. The complexity of this algorithm is in $O(m \cdot d)$. However, this approach yields much worse results than the grid-based approach for an identical m parameter. Details can be found in our experiments (Section 5).

3.2.3 Domination Count Estimation based on bisections

We next propose a bisection based approach that yields much better efficacy, while still being linear in d. This approach works iteratively. During each iteration, one section of R is chosen to be split evenly (mean split) in one dimension. After m splits, this results in $m + 1$ sections $s_0 \cup s_1 \cup \ldots \cup s_m = R$ and it holds that:

$$DC_{bisect}(\mathcal{O}, B, R) = min_i(DC_{basic}(\mathcal{O}, B, s_i))$$

The challenge here is to wisely choose the split section of R and the dimension to split in each iteration.

We propose to split the section $s \subset R$ with the lowest domination count estimation. This decision is optimal, because the estimation of $DC(\mathcal{O}, B, R)$ is determined by the section which results in the lowest domination count. Thus, in order to increase the domination count approximation, s must be split. If the decision for s is ambiguous, then one of the candidates of s is chosen arbitrarily. To determine the

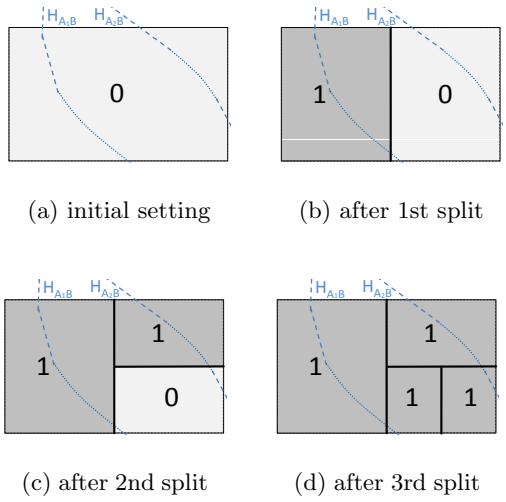


Figure 7: Example for computing DC_{bisect}

split axis, the heuristic tests each dimension, and greedily uses the dimension that yields the highest domination count $DC_{bisect}(\mathcal{O}, B, R)$ considering the two resulting bisections of s. In the case of ties the axis is chosen which maximizes the sum $\sum_{i=0}^{m} DC_{basic}(\mathcal{O}, B, s_i)$. An example is shown in Figure 7. Considering Figure 7(a) it is clear that none of the two objects $A_1, A_2 \in \mathcal{O}$ that are responsible for the Voronoi hyperplanes H_{A_1B} and H_{A_2B} dominates B w.r.t. R. Beginning with the y-axis as split axis would result in two equi-sized MBRs both of which result in a domination count $DC_{bisect}(\mathcal{O}, B, R)$ of 0 and therefore the approximation of $DC(\mathcal{O}, B, R)$ does not increase. Choosing the x-axis as split axis would result in two equi-sized MBRs shown in Figure 7(b) yielding the same domination count approximation but a higher sum ($\sum_{i=0}^{m} DC_{basic}(\mathcal{O}, B, s_i) = 1$). In the next iteration, the right MBR is chosen to be split, since it is responsible for the lowest domination count approximation. Both possible split axes are equal according to our heuristic. In the example, the y-axis is chosen arbitrarily (c.f. Figure 7(c)). The third (see Figure 7(d)) split of the lower-right MBR increases $DC_{bisect}(\mathcal{O}, B, R)$ to 1.

The bisection-based Domination Count Estimation algorithm uses m iterations. In each iteration i there exist exactly i sections of which the section with the lowest conservative domination count has to be found. This yields a complexity of $O(m^2)$ but can be reduced to $O(m \cdot log(m)))$ by using a Priority Queue to find the section with the lowest conservative domination count. For the greedy heuristic, in each iteration, each dimension has to be tested to determine the best split axis in $O(m \cdot d)$. Thus we get a total complexity of $O(m \cdot log(m) + m \cdot d) = O(m \cdot \max(log(m), d))$, where m is the number of iterations.

4. BOOSTING SIMILARITY QUERIES

In this section, we will show how the concepts of domination and domination count can be used to boost the pruning power of similarity search algorithms.

Nearest-Neighbor Search. For a kNN query with query object Q, any object $O \in \mathcal{D}$ can be pruned if $DC(\mathcal{D}, O, Q) \geq k$. Note, that for a kNN query, the query object corresponds

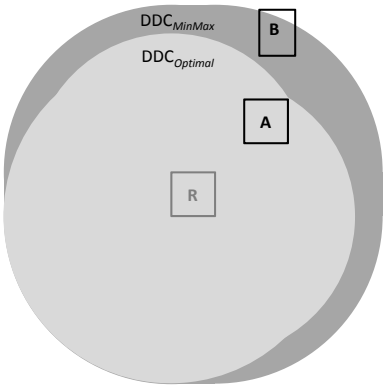


Figure 8: Refinement areas for fixed R and A

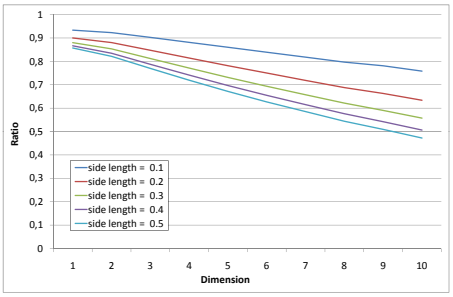


Figure 9: Ratio of the refinement areas of $DDC_{Optimal}$ and $DDC_{\mathbf{MinMax}}$ w.r.t. dimension and size of MBRs

to the reference object R in Definition 4. Thus, $DDC_{Optimal}$ has an advantage over $DDC_{\mathbf{MinMax}}$ in the general case but is equivalent in the special case where Q is a point, because then $DDC_{\mathbf{MinMax}}$ is optimal. However, as discussed above, there are many applications in which the query object is a rectangle.

Reverse Nearest Neighbor Search. For a general RkNN query with query object Q, any object $O \in DB$ can be certainly pruned if $DC(\mathcal{D}, Q, O) \geq k$. For R$k$NN queries, the query object corresponds to the object B in Definition 4. Thus $DDC_{Optimal}$ is superior to $DDC_{\mathbf{MinMax}}$ also in the special case where the query object is given as a point.

True hit detection. Our decision criterion $DDC_{Optimal}$ can be used to prune potential result candidates by being able to decide that they must not be part of the result set. A problem very similar to pruning is the detection of true hits, i.e. to quickly decide that a potential result candidate must be part of the result set. For example, in the case of kNN queries, an object B is a true hit, if there may be at most k objects that can be closer to R than B. In other words, B is a true hit, if it dominates at least $|\mathcal{D}|-k$ objects. Thus, for a kNN query, an object B is a true hit if $|\{A \in \mathcal{D}|dom(B, A, Q)\}| > |\mathcal{D}| - k$. For a R$k$NN query, an object B is a true hit if $|\{A \in \mathcal{D}|dom(Q, A, B)\}| \geq |\mathcal{D}| - k$. The concept of partial domination can be applied to true hit detection as well.

Inverse Similarity Ranking. The problem of inverse ranking is to determine for a given query object Q the number of objects that are closer to a given reference object R. Such queries are useful e.g. to determine the financial standing of bank customers in relation to existing customers. In this scenario, the attributes of customers are often uncertain (e.g. income of $40k-50$k) and thus modeled by uncertain regions, i.e. rectangles. Lower and upper bounds for the rank of Q are $DC(\mathcal{D}, Q, R) + 1$ and $|\mathcal{D}| - |\{A \in \mathcal{D}|dom(B, A, Q)\}| + 1$, respectively.

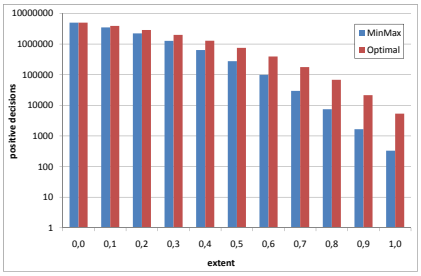
5. EXPERIMENTAL EVALUATION

This section evaluates the effectiveness and efficiency of our novel domination decision criterion in comparison to the prevalent $DDC_{\mathbf{MinMax}}$ decision criterion. After that we evaluate the performance of our domination-count-detection approach which is based on the concept of partial domination. Finally, we exemplarily will show how our new methods influences the performance of existing similarity search methods designed for kNN and RkNN queries. For all experiments the underlying distance function is the euclidian norm.

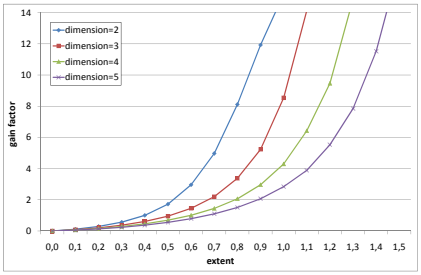
5.1 Single Object Domination

We first evaluate the effectiveness gain of $DDC_{Optimal}$ compared to $DDC_{\mathbf{MinMax}}$ in consideration of the decision power. In order to measure the decision power, we take for a given pair of rectangles R and A the region into account containing all points that cannot be detected to be dominated by A w.r.t. R. In the reminder we call this region refinement area, since all objects intersecting this area might be refined in order to detect the domination relation. It should be clear that the smaller this area, the higher the corresponding domination power. Figure 8 exemplarily shows the refinement areas for the 2-dimensional MBRs A and R w.r.t. both criteria $DDC_{\mathbf{MinMax}}$ and $DDC_{Optimal}$, respectively. In this example, object B is detected to be dominated by A only if we apply $DDC_{Optimal}$ instead of $DDC_{\mathbf{MinMax}}$. The refinement areas depend on several conditions such as position, shape, distance and extension of the MBRs specifying the refinement area as well as the dimensionality of the space. For our experiment evaluating the domination power, pairs of MBRs R and A are positioned in $[0, 1]^d$ with a fixed *MinDist* of 0.5 and equal distances in each dimension. The length of each side of the two MBRs was scaled from 0.1 to 0.5 and dimension screened from 1 to 10. The gain of the domination power is measured by the ratio of the volumes of the refinement area w.r.t. $DDC_{Optimal}$ and the refinement area w.r.t. $DDC_{\mathbf{MinMax}}$ by means of Monte-Carlo-Sampling. The results in Figure 9 show that $DDC_{Optimal}$ leads to a much higher decision power. The effect becomes more evident as the number of dimensions and the extension of the MBRs increase. As expected, increasing the extension of the MBRs leads to diminishing completeness of the $DDC_{\mathbf{MinMax}}$ decision criterion. It is notable, that the $DDC_{\mathbf{MinMax}}$ criterion suffers considerably from an increasing dimensionality. Note that we used MBRs of equal side length as we observed that this setting favors the decisions power based on $DDC_{\mathbf{MinMax}}$ in order to make a fair comparison. In fact, the advantage of the gain of the decision power based on $DDC_{Optimal}$ will increase even further for non-quadratic rectangles.

In addition to the above experiment which is more from a theoretical point of view, we compared the number of domination relations detected by applying $DDC_{Optimal}$ and $DDC_{\mathbf{MinMax}}$. Therefore we randomly generated one mil-

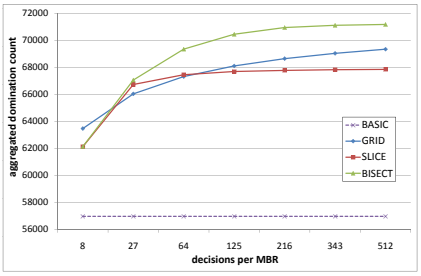


(a) Positive decisions made

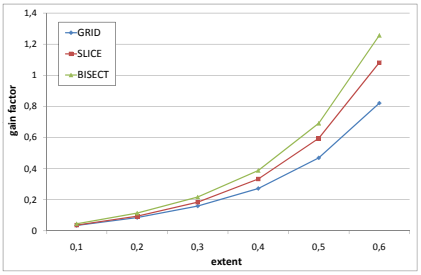


(b) Factor of positive decisions made more by the optimal criterion

Figure 10: Comparison of MinMax- and optimal-criterion on synthetic data



(a) Accuracy vs. efficiency



(b) Performance w.r.t. MBR size

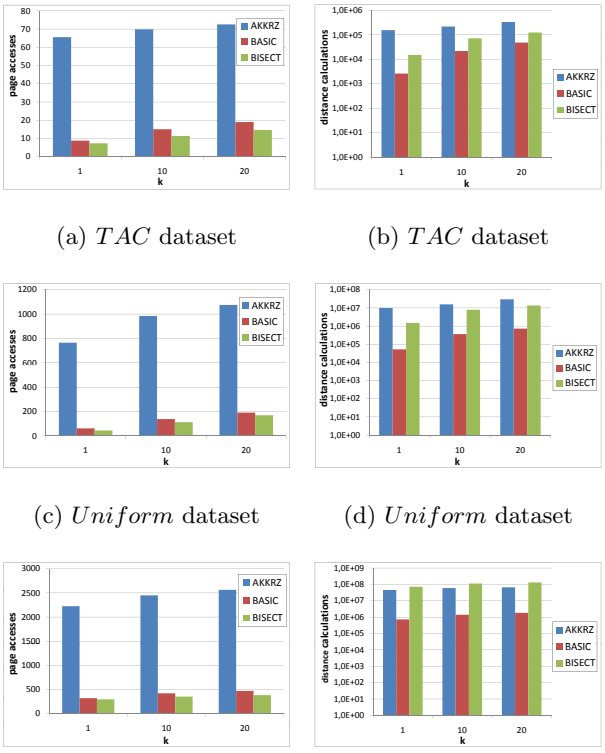
Figure 11: Different heuristics for partial domination

lion triples of rectangles (A, B, R) with a fixed extent (i.e. the sum of side lengths) in the $[0, 1]^2$ space. For each triple we tested if the decision criterion is able to determine whether $Dom(A, B, R)$ holds. Finally we aggregated the number of positive decisions for different extents of the MBRs. The results are illustrated in Figure 10(a). Note that an extent of zero yields points instead of rectangles such that both criteria perform equal. However, we can observe that with increasing extent, the percentage of positive decisions of $DDC_{Optimal}$ compared to $DDC_{\mathbf{MinMax}}$ increases considerably. The gain of the decision power based on $DDC_{Optimal}$ over $DDC_{\mathbf{MinMax}}$ is illustrated in Figure 10(b) showing the factor of positive domination decisions using $DDC_{Optimal}$ in comparison of that using $DDC_{\mathbf{MinMax}}$. We varied the dimensionality of the rectangle space up to 5 dimensions. Here we can observe that the gain increases with increasing extent. In contrast, when increasing the dimensionality, the gain of the decision power decreases. The reason is that in this setting, the extent of the MBRs is fixed for all dimensionality settings such that the average side length per dimension decreases and MBRs converge to points for high dimensionality.

5.2 Domination Count Estimation

The next experiments evaluate the accuracy of the domination count estimation of a rectangle B w.r.t. a rectangle R for the approaches proposed in Section 3.2: Basic Domination Count Estimation (DC_{basic}), grid partitioning (DC_{grid}), slice partitioning (DC_{slice}) and bisection based partitioning (DC_{bisect}). For these experiments, we generated one thousand three-dimensional MBRs with random position. One MBR R was positioned in the center of the data space. Then we computed the conservative domination count w.r.t. R for each MBR using the four approaches mentioned above. We performed several runs for different parametric settings and averaged the results. Figure 11 shows the performance of all four approaches in terms of estimated domination count. First, we want to get a grasp of the relationship between accuracy of the domination count estimation and the cost required for the domination count computation. Therefore, the cost is measured in terms of number of calls of $DDC_{Optimal}$. It should be clear that when increasing the number of MBR partitions, and thus the required number of $DDC_{Optimal}$ calls, the estimation accuracy of all approaches improves, except for the basic approach since it does not use any partitioning. Figure 11(a) shows the results for MBRs with an extent of 0.3. It can be seen that all approaches show a significant improvement compared to DC_{basic} when increasing the number of allowed $DDC_{Optimal}$ calls. In particular, the accuracy increases very fast at the beginning of the partitioning process but slows down later on. We can also observe that DC_{bisect} significantly outperforms the other approaches when allowing more than 27 $DDC_{Optimal}$ calls per MBR, while DC_{grid} performs best for 8 or less $DDC_{Optimal}$ calls.

In the next experiment, as shown in Figure 11(b), we fixed the number $DDC_{Optimal}$ calls per MBR to 64 and varied their extent. We measured the gain of the domination count over DC_{basic}. Here, again, DC_{bisect} outperforms the other approaches in particular for larger MBR sizes. Note that for a given application, the optimal number of partitions



(a) TAC dataset (b) TAC dataset

(c) $Uniform$ dataset (d) $Uniform$ dataset

(e) $Forest$ dataset (f) $Forest$ dataset

Figure 12: *AKKRZ* using different decision criteria. Page accesses (left side) and distance calculations (right side).

depends on the cost for evaluating a candidate object. The higher that cost, the more partitions can be used in order to reduce the total runtime.

5.3 Impact on Standard Spatial Query Processing Methods

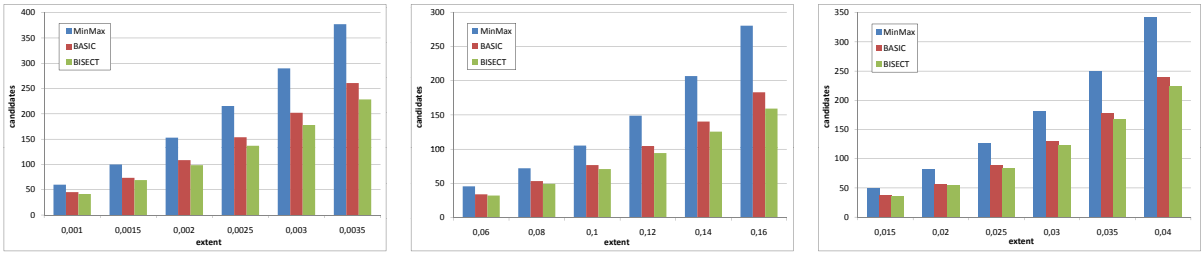
In our last experiments, we evaluate the impact of our approaches on the performance of standard query processing methods. Here, we refer to Section 4, describing how our methods can be plugged into state-of-the-art query processing methods. In particular, we exemplarily consider the most prominent query methods, the k-nearest neighbor (k-NN) search and the reverse k-nearest neighbor (Rk-NN) search. For this evaluation we use one synthetic dataset, containing 100k uniformly distributed 5D points, and two real world datasets TAC[24] consisting of 705099 2D points and $Forest$[15] containing 581012 10D points.

First, we evaluate the impact of our two domination-count estimation approaches DC_{basic} and DC_{bisect} on a reverse k-nearest neighbor search method. As a baseline, we use the algorithm proposed in [1] (in the following referred to $AKKRZ$) for Rk-NN search on the Euclidean space using an R^*-Tree. The $AKKRZ$ algorithm originally uses the Min-/MaxDist decision criterion to conservatively prune candidates. We evaluate the impact by comparing the query performance of the original AKKRZ algorithm with the version where we replace the domination count estimation with our methods. Note, that with except of the domination count estimation method, both Rk-NN versions are identical. The results illustrated in Figures 12(a), 12(c) and 12(e) show the query performance of both Rk-NN versions in terms of average number of page accesses for varying parameter k and different datasets. It is notable that the enhanced algorithm requires less page access by almost a full order of magnitude on all datasets. Using DC_{bisect} to apply the paradigm of partial pruning based on bisections (c.f. Section 3.2.3) with a maximum number of ten splits per MBR, the number of page accesses can be significantly dropped even further. The large performance increase compared to the original version of $AKKRZ$ can be explained by the fact that our domination decision criterion has a much higher pruning power on large MBRs compared to the original version that is based on the Min/MaxDist criterion. This allows us to prune candidates already on a high directory level and, thus, to prune a large number of candidate MBRs very early.

Beside the I/O cost, it is also important to consider the cpu cost since the accuracy of our domination count estimation methods is highly influenced by the cpu cost spent for the estimation process, as shown in the previous section. For this reason, in addition to the I/O cost evaluation we evaluate the cpu cost measured by the number of distance calculations required for the competing techniques as the cpu cost are mainly distance computation bounded. We counted the total number of distance calculations. Calls of $DDC_{Optimal}$ and $DDC_{\mathbf{MinMax}}$ were penalized with two distance calculations[1]. The resulting numbers of total distance calculations are shown in Figures 12(b), 12(d) and 12(f). It can be observed that the enhanced $AKKRZ$ algorithm using DC_{basic} significantly outperforms the basic $AKKRZ$ by close to two orders of magnitude. The rationale for this is that the number of calculations increases quadratic in the number of candidates. However, the high computational cost required when applying partial pruning becomes evident here. Using DC_{bisect} with a maximum of ten splits, the number of distance calculations increases by a factor of about five.

Finally, we evaluate the impact of $DDC_{Optimal}$ and partial domination on k-NN queries among objects approximated by MBRs. These experiments are based on three artificial datasets that rely on the three datasets used in the foregoing experiments (TAC, Uniform, Forest). Each vector in a dataset defines the center of an MBR. For each of the resulting datasets 100 MBRs were chosen randomly as query MBR Q for a 10-NN query on the remaining dataset. Here we did not apply any index structure. The performance of the competing approaches were measured by the average number of candidates that could neither be pruned nor be reported as true hits. The results showing the performance in terms of the number of remaining candidates are depicted in Figure 13 for varying extent of the MBRs. It can be observed, that DC_{basic} significantly reduces the number of candidates compared to pruning based on $DDC_{\mathbf{MinMax}}$ on all datasets. The relative performance boost increases for an increasing extent of the MBRs. We also found out in our experiments, that the parameter k has no significant influence on the relative performance boost. Figure 13 also shows, that DC_{bisect} is able to further boost the performance, especially for large MBRs.

[1]in concordance with run-time experiments omitted here due to space considerations



(a) *TAC* dataset (2D) (b) *Uniform* dataset (5D) (c) *Forest* dataset (10D)

Figure 13: Evaluation of the different decision criteria for 10 nearest neighbor queries.

6. CONCLUSIONS

The concept of domination is a very useful tool to perform spatial pruning on rectangles in a wide field of applications. Current state-of-the-art approaches are either incomplete or scale exponentially in the number of dimensions. In this paper we proposed a decision criterion that is complete and efficiently computable in $O(d)$. In addition, we discuss how this decision criterion can be used to accurately estimate the domination count of objects by incorporating information about partial domination. While all current approaches that use information about partial domination can only be used on two dimensional data our solution can be applied to data of arbitrary dimensionality. Our experimental evaluation shows that our novel decision criterion can be used to vastly increase the pruning power of existing applications by several orders of magnitude. For future work, we plan to plug-in our novel decision criterion to more existing applications. In addition we will explore decision criteria for object representations having non-rectangular shape.

Acknowledgements

This research has been supported in part by the THESEUS program in the MEDICO and CTC projects. They are funded by the German Federal Ministry of Economics and Technology under the grant number 01MQ07020. The responsibility for this publication lies with the authors.

7. REFERENCES

[1] E. Achtert, H.-P. Kriegel, P. Kröger, M. Renz, and A. Züfle. Reverse k-nearest neighbor search in dynamic and general metric databases. In *EDBT*, pages 886–897, 2009.

[2] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-Tree: An efficient and robust access method for points and rectangles. In *Proc. SIGMOD*, 1990.

[3] S. Berchtold, D. A. Keim, and H.-P. Kriegel. The X-Tree: An index structure for high-dimensional data. In *Proc. VLDB*, 1996.

[4] G. Beskales, M. A. Soliman, and I. F. IIyas. Efficient search for the top-k probable nearest neighbors in uncertain databases. *Proc. VLDB Endow.*, 1(1):326–339, 2008.

[5] S. Brecheisen, H.-P. Kriegel, P. Kröger, M. Pfeifle, and M. Schubert. Using sets of feature vectors for similarity search on voxelized CAD objects. In *Proc. SIGMOD*, 2003.

[6] J. Chen and R. Cheng. Efficient evaluation of imprecise location-dependent queries. 2007.

[7] R. Cheng, J. Chen, M. Mokbel, and C. Chow. Probabilistic verifiers: Evaluating constrained nearest-neighbor queries over uncertain data. 2008.

[8] R. Cheng, D. Kalashnikov, and S. Prabhakar. Querying imprecise data in moving object environments. In *IEEE TKDE*, 2004.

[9] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 551–562, New York, NY, USA, 2003. ACM.

[10] T. Emrich, H.-P. Kriegel, P. Kröger, M. Renz, and A. Züfle. Constrained reverse nearest neighbor search on mobile objects. In *ACM SIGSPATIAL GIS*, 2009.

[11] T. Emrich, H.-P. Kriegel, P. Kröger, M. Renz, and A. Züfle. Incremental reverse nearest neighbor ranking in vector spaces. In *SSTD*, pages 265–282, 2009.

[12] V. Gaede and O. G"unther. Multidimensional access methods. *ACM CSUR*, 30(2):170–231, 1998.

[13] A. Guttman. R-Trees: A dynamic index structure for spatial searching. In *Proc. SIGMOD*, pages 47–57, 1984.

[14] M. Hadjieleftheriou, G. Kollios, J. Tsotras, and D. Gunopulos. Indexing spatiotemporal archives. *The VLDB Journal*, 15(2):143–164, 2006.

[15] S. Hettich and S. D. Bay. The uci kdd archive., 1999.

[16] G. R. Hjaltason and H. Samet. Ranking in spatial databases. In *Proc. SSD*, 1995.

[17] E. Keogh. Exact indexing of dynamic time warping. In *Proc. VLDB*, 2002.

[18] X. Lian and L. Chen. Probabilistic inverse ranking queries over uncertain data. 2009.

[19] N. Roussopoulos, S. Kelley, and F. Vincent. Nearest neighbor queries. In *SIGMOD '95*, pages 71–79, 1995.

[20] Y. Tao, R. Cheng, X. Xiao, W. K. Ngai, B. Kao, and S. Prabhakar. Indexing multi-dimensional uncertain data with arbitrary probability density functions. In *VLDB '05*. VLDB Endowment, 2005.

[21] Y. Tao, D. Papadias, and X. Lian. Reverse kNN search in arbitrary dimensionality. In *Proc. VLDB*, 2004.

[22] Y. Tao, D. Papadias, and Q. Shen. Continuous nearest neighbor search, 2002.

[23] S. Šaltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez. Indexing the positions of continuously moving objects. *SIGMOD Rec.*, 29(2):331–342, 2000.

[24] N. Zacharias and M. I. Zacharias. The twin astrographic catalog on the hipparcos system. *The Astronomical Journal*, 118(5):2503–2510, 1999.

APPENDIX

Lemma 5 states the following: Let D be a one dimensional vector database using L_p-Norm. Let A and B be intervals. The function $f : \mathbb{R} \rightarrow \mathbb{R}$:

$$f(x) = maxDist(A, x)^p - minDist(B, x)^p$$

has no local maximum.

PROOF. By definition, the value of $MaxDist(A, x)$ only depends on $A.mean$ while the value of $MinDist(B, x)$ only depends on $B.min$ and $B.max$. Since $A.min \geq A.max$, this leads to the three possible cases depicted in Figure 14. In the following, we will show for each of the cases that there may not exist any point for which it holds that $f(x)$ is increasing to the left of x and decreasing to the right of x.

Case 1: $A.mean \leq B.min$ (Figure 14(a)):

- For the interval $]-\infty, A.mean]$ Equations 2 and 3 yield: $f(x) = dist(x, A.max)^p - dist(x, B.min)^p = (|A.max - x|)^p - (|B.min - x|)^p$. This is constant if $p = 1$ or if $A.max = B.min$ since both $|A.max - x|$ and $|B.min - x|$ are decreasing. If $p > 1$, this term can be either **monotonically increasing or monotonically decreasing**, depending on whether $A.max$ is greater than $B.min$ or not.
- For the interval $]A.mean, B.min]$ Equations 2 and 3 yield: $f(x) = dist(x, A.min)^p - dist(x, B.min)^p$, where dist(x,A.min) is increasing and dist(x,B.min) is decreasing, thus $f(x)$ is **monotonically increasing** for any p.
- For the interval $]B.min, B.max]$ Equations 2 and 3 yield: $f(x) = dist(x, A.min)^p - 0$. This term is **monotonically increasing** since $dist(x, A.min)$ is increasing.
- For the interval $]B.max, \infty[$ Equations 2 and 3 yield: $f(x) = dist(x, A.min)^p - dist(x, B.max)^p = (|A.min - x|)^p - (|B.max - x|)^p$. This is constant for $p = 1$ since both $|A.min - x|$ and $|B.max - x|$ are increasing. Since $A.min < A.mean < B.min < B.max$, above term is **monotonically increasing** for $p > 1$.

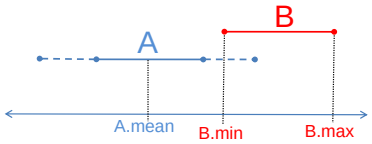
Putting the monotonic pieces together, it is clear that $f(x)$ may have one local minimum at $A.mean$, but definitely has no local maximum.

Case 2: $B.min < A.mean < B.max$ (Figure 14(b)):

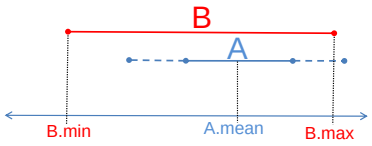
- For the interval $]-\infty, B.min]$ Equations 2 and 3 yield: $f(x) = dist(x, A.max)^p - dist(x, B.min)^p = (|A.max - x|)^p - (|B.min - x|)^p$. This term is constant for $p = 1$ since both $|A.max - x|$ and $|B.min - x|$ are decreasing. Since $|A.max - x| \geq |A.mean - x| > |B.min - x|$, above term is **monotonically decreasing** for $p > 1$
- For the interval $]B.min, A.mean]$ Equations 2 and 3 yield: $f(x) = dist(x, A.max)^p - 0$. This is **monotonically decreasing** for any $p \geq 1$.
- For the interval $]A.mean, B.max]$ Equations 2 and 3 yield: $f(x) = dist(x, A.min)^p - 0$. This is **monotonically increasing** for any $p \geq 1$.
- For the interval $]B.max, \infty[$ Equations 2 and 3 yield: $f(x) = dist(x, A.min)^p - dist(x, B.max)^p = (|A.min - x|)^p - (|B.max - x|)^p$.. This is constant for $p = 1$ since both $|A.min - x|$ and $|B.max - x|$ are increasing. Since $|A.min - x| \leq |A.mean - x| < |B.min - x|$, above term is **monotonically increasing** for $p > 1$.

Thus, $f(x)$ has one local minimum at $A.mean$, but definitely no local maximum.

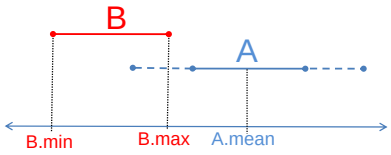
Case 3: $A.mean \geq B.max$ (Figure 14(c)): This case is analogous to the first case. □



(a) $A.mean \leq B.min$



(b) $B.min < A.mean < B.max$



(c) $A.mean \geq B.min$

Figure 14: Illustration of Lemma 5.