Optimal Spatial Dominance: An Effective Search of Nearest Neighbor Candidates

Xiaoyang Wang[†], Ying Zhang[‡], Wenjie Zhang[†], Xuemin Lin[†], Muhammad Aamir Cheema[♮]
[†]The University of New South Wales, Australia [‡]University of Technology, Sydney, Australia
[♮]Monash University, Australia
xiaoyangw@cse.unsw.edu.au, Ying.Zhang@uts.edu.au, {zhangw, lxue}@cse.unsw.edu.au, aamir.cheema@monash.edu

ABSTRACT

In many domains such as computational geometry and database management, an object may be described by multiple instances (points). Then the distance (or similarity) between two objects is captured by the pair-wise distances among their instances. In the past, numerous *nearest neighbor* (NN) functions have been proposed to define the distance between objects with multiple instances and to identify the NN object. Nevertheless, considering that a user may not have a specific NN function in mind, it is desirable to provide her with a set of NN candidates. Ideally, the set of NN candidates must include every object that is NN for at least one of the NN functions and must exclude every non-promising object. However, no one has studied the problem of NN candidates computation from this perspective. Although some of the existing works aim at returning a set of candidate objects, they do not focus on the NN functions while computing the candidate objects. As a result, they either fail to include an NN object w.r.t. some NN functions or include a large number of unnecessary objects that have no potential to be the NN regardless of the NN functions.

Motivated by this, we classify the existing NN functions for objects with multiple instances into three families by characterizing their key features. Then, we advocate three spatial dominance operators to compute NN candidates where each operator is *optimal* w.r.t. different coverage of NN functions. Efficient algorithms are proposed for the dominance check and corresponding NN candidates computation. Extensive empirical study on real and synthetic datasets shows that our proposed operators can significantly reduce the number of NN candidates. The comprehensive performance evaluation demonstrates the efficiency of our computation techniques.

Categories and Subject Descriptors

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

Keywords

Spatial dominance; NN candidates; NN functions

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGMOD'15, May 31–June 4, 2015, Melbourne, Victoria, Australia.
Copyright © 2015 ACM 978-1-4503-2758-9/15/05 ...$15.00.
http://dx.doi.org/10.1145/2723372.2749442.

1. INTRODUCTION

Nearest neighbor (NN) search identifies the closest (or most similar) object to a given query object. It is a classical problem with applications in many domains such as computational geometry, multi-media database, information retrieval and machine learning, to name a few. In many real-world applications, an object may be described by a set of points (i.e., instances). Each point is associated with a value (e.g., weight and probability). For example, in multi-media database, an object (e.g., an image) may be mapped onto a set of feature vectors (i.e., points) [19, 28, 35]. The performance of an NBA player can be evaluated by his historical game records [26, 25, 42, 37]. Suppose the number of points, assistants and rebounds are kept for each game record, a player can be described by a set of points in a three dimensional space. Due to the limitations of measuring equipment, the location of an object (e.g., a person or a mobile device) may be uncertain [40] and hence be represented by a set of possible locations. Each location is associated with an *occurrence probability.*

In traditional NN search problem, each object (and query) is modeled as a single point in a multi-dimensional space. The distance between the object and the query is computed using the given distance metric (e.g., Euclidean distance) and the object with the smallest distance value is returned. However, it is non-trivial to extend the NN semantics for objects with multiple instances because the similarity between two objects is described by their distance distribution (i.e., pair-wise distances). Consequently, a large body of NN functions are proposed in the literature. In this paper, we propose three models to characterize the intrinsic nature of these NN functions (details are introduced in Section 3).

- ($\mathcal{N}_1$) **All pairs** based NN functions (e.g., *min*, *max*, *expected*, and *quantile* distances [17, 37]).
- ($\mathcal{N}_2$) **Possible world** based NN functions (e.g., NN probability [21, 7], expected rank [40], and parameterized rank [23]).
- ($\mathcal{N}_3$) **Selected pairs** based NN functions (e.g., Hausdorff distance [27], Earth Mover's distance [10, 28, 35] and Netflow distance [27]).

These three types of NN functions cover the popular NN functions for both discrete uncertain/probabilistic objects (e.g.,[12, 4, 22, 3]) and traditional multi-valued/multi-instance objects (e.g.,[37, 42, 20, 34]). These two object models have different semantics [37]. Specifically, a *discrete uncertain object* is a discrete random variable where each instance is associated with a probability value. The uncertain object model enforces the exclusive property when the possible world based NN functions ($\mathcal{N}_2$) are used because two instances of an uncertain object (e.g., two possible locations of an object) cannot occur in the same possible world. Uncertain object model has also been used by some NN functions in $\mathcal{N}_1$ (e.g., expected distance [12]) and $\mathcal{N}_3$ (e.g., Earth Mover's distance [28, 35]). *A multi-valued object* consists of a set of points (instances), and each instance carries a weight to reflect its significance. Note that the instances of a multi-valued object will co-exist and hence the multi-valued object model is not suitable to the possible world semantics (i.e., NN functions in $\mathcal{N}_2$). Functions in $\mathcal{N}_1$ (e.g., expected distance and quantile distance [37]) and $\mathcal{N}_3$ (e.g., Earth Mover's distance [10]) are used for multi-valued objects.

While the scope of this paper covers both discrete uncertain objects and multi-valued objects, we observe that the stochastic order [29] and its variants are effective tools to derive the candidates for various NN functions. To this end, despite of the different semantics, we may treat a multi-valued object as a random variable if the weights of its instances can be normalized. In particular, we say the multi-valued objects can be normalized w.r.t. an NN function f if their NN ranks remain the same after the instance weights normalization. It can be immediately verified that the multi-valued objects can be normalized for NN functions investigated in this paper, if the total weight mass of each object is the same, which is common in practice. In this way, we can generate *correct* NN candidates w.r.t. corresponding NN functions. Thereafter, we may use a discrete uncertain object (i.e., a discrete random variable) to denote an object with multiple instances.

Nearest Neighbor Candidates (NNC) Search. In practice, users often do not have a specific NN function in mind. One solution is that the system explains different NN functions before a user performs NN search, so she can choose one or a few desirable NN functions. Considering that it might be difficult for an inexperienced user to choose some complicated NN functions as well as the settings of the parameters, as a valuable complement, system may provide the user a small set of "good" NN candidates. Then she can browse the NN candidates and make her decision, e.g., by using visualization tools. Such an approach is also useful when an inexperienced or "greedy" user wants to make personal trade-off among the results of many different NN functions, which rank objects from various perspectives. To the best of our knowledge, there are two existing approaches which can provide nearest neighbor candidates (NNC).

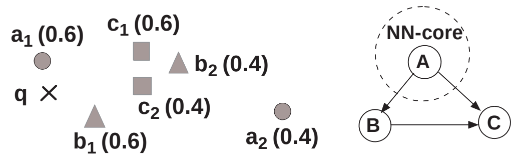
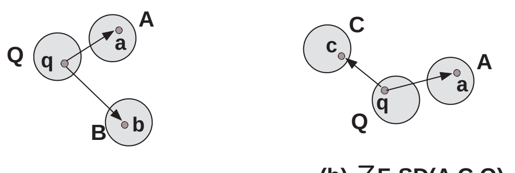


Figure 1: Example of NN-core

In [36], NN candidates (namely *NN-core*) are derived based on the pair-wise competition among the objects where an object U is said to supersede another object V if U is more likely to be closer to the query than V. Then, NN-core is the minimal set of objects such that each object in NN-core supersedes every object not in the NN-core. Figure 1 illustrates an example of the NN-core of three objects $\{A, B, C\}$ where the query object has a single instance q and each of the objects consists of two instances with probability 0.6 and 0.4, respectively. In this example, A is closer to q than B with probability 0.6, and hence A supersedes B (denoted by $A \rightarrow B$ in Figure 1). Similarly, we have A supersedes C and B supersedes C, and hence the NN-core is $\{A\}$.



(a) F-SD(A,B,Q) (b) $\neg$F-SD(A,C,Q)

Figure 2: Examples of F-$\mathcal{SD}$

The *full spatial dominance* (**F-SD**) operator has been used recently to prune non-promising NN candidates where each object is approximated by a rectangle [16] or a hypersphere [25]. Given two objects U and V, and a query Q, we say U fully spatially dominates V w.r.t. Q, denoted by F-$\mathcal{SD}(U,V,Q)$, if and only if for every instance $q \in Q$, U is closer to q than V; that is, $\delta(u, q) \leq \delta(v, q)$ for every $u \in U$ and every $v \in V$ where $\delta(x, y)$ denotes the distance between two points (instances) x and y. Then, NN candidates are the objects which are not fully spatially dominated by any other object. Figure 2 shows examples of F-$\mathcal{SD}$ where instances of each object are bounded by a circle. Then we have F-$\mathcal{SD}(A,B,Q)$ in Figure 2(a) as $\delta(a, q) < \delta(b, q)$ for *every* $q \in Q$, $a \in A$, and $b \in B$. Figure 2(b) shows that F-$\mathcal{SD}(A,C,Q)$ does not hold, denoted by $\neg$F-$\mathcal{SD}(A,C,Q)$, since we have $\delta(c, q) < \delta(a, q)$ for the given $q \in Q$, $a \in A$, and $c \in C$.

Motivation and Challenges. As discussed earlier, there are many possible NN functions that can be used to retrieve the NN objects. Intuitively, the computation of NN candidates should consider these functions and, ideally, the NN candidates must include every object that can be the NN object for at least one NN function and must not contain any object that can never be the NN object regardless of which function is used. Unfortunately, none of the existing works investigates the problem from this perspective. Specifically, NN-core [36] is too aggressive in the sense that NN object may be missed from NN-core for some popular NN functions. For instance, in Figure 1, C is the NN object if function *max* is used while B becomes the NN object if the *expected distance* is considered. Unlike NN-core that is too aggressive, F-$\mathcal{SD}$ is overly pessimistic and may return a large number of objects which can never be the NN of Q.

Motivated by the above facts, we devise spatial dominance (SD) operators that compute NN candidates by carefully considering various NN functions. Let $f(X)$ denote the distance between X and Q computed using an NN function f. We say that a spatial dominance operator $\mathcal{SD}$ is **optimal** w.r.t. a family $\mathcal{N}$ of NN functions if it satisfies the following two properties.

- **Correctness.** $\mathcal{SD}(U,V,Q)$ implies that $f(U) \leq f(V)$ for every $f \in \mathcal{N}$, i.e., V is not closer to Q than U for every function $f \in \mathcal{N}$. In this paper, we say that $\mathcal{SD}$ *covers* the functions in $\mathcal{N}$ if it is *correct* w.r.t. $\mathcal{N}$.
- **Completeness**. $\neg\mathcal{SD}(U,V,Q)$ implies that there exists a function $f \in \mathcal{N}$ such that $f(V) < f(U)$, i.e., V is closer to Q than U w.r.t. f.

The *correctness* property of the $\mathcal{SD}$ guarantees that it is safe to exclude the object V from NN candidates w.r.t. $\mathcal{N}$. The *completeness* property indicates that $\mathcal{SD}$ is the *tightest* dominance operator w.r.t. $\mathcal{N}$ in the sense that U cannot exclude V from NN candidates if and only if there exists a function $f \in \mathcal{N}$ for which V is ranked better than U.

Given a family of NN functions, a straightforward way to compute NN candidates is to retrieve NN objects for all possible NN functions. However, this is infeasible not only due to the unnecessary computations involved but also because the number of NN functions is infinite, e.g., ϕ-quantile distance depends on the input parameter ϕ that may take every value between 0 to 1 [37].

Next, we give an informal but intuitive explanation of three *optimal* spatial dominance (SD) operators to obtain NN candidates, which are *optimal* w.r.t. different coverage of the NN functions.

Stochastic SD (S-$\mathcal{SD}$, opt. w.r.t. $\mathcal{N}_1$). A function $f \in \mathcal{N}_1$ is an aggregate function on the *distance distributions* of the two objects, i.e., all pair-wise distances. We apply the widely employed *usual stochastic order* [29] to model the S-$\mathcal{SD}$ operator which is optimal w.r.t. $\mathcal{N}_1$. Given two random variables X and Y, we say X *is smaller than* (or dominates) Y in usual stochastic order, denoted by $X \preceq_{st} Y$, if $Pr(X \leq \lambda) \geq Pr(Y \leq \lambda)$ for every $\lambda \in R$. Note that smaller value is preferred in this paper. Thus, we have S-$\mathcal{SD}(U,V,Q)$ if $U_Q \preceq_{st} V_Q$ where U_Q (V_Q) denotes the distance distribution of U (V) w.r.t. Q. Intuitively, S-$\mathcal{SD}(U,V,Q)$ implies that, for every λ, the probability that the distance between U and Q is at most λ is not smaller than that of V.

Figure 3(a) shows an example where each object (query) has two instances with the same probability (0.5), and Figure 3(b) depicts the distance distributions of three objects regarding the query. In each distance distribution (e.g., A_Q), each pair of instances between the query and the object (e.g., q_1a_1) carries probability 0.25, and they are sorted based on their distance values. Clearly, we have S-$\mathcal{SD}(A,B,Q)$, S-$\mathcal{SD}(A,C,Q)$, and $\neg$S-$\mathcal{SD}(B, C, Q)$.

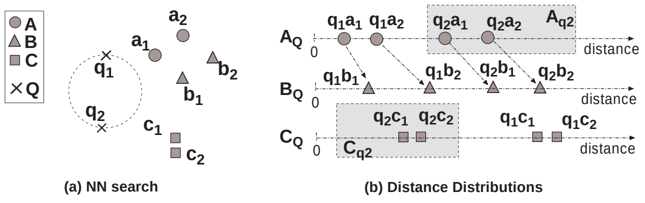

A
B
C
Q
a_2
a_1
q_1
b_2
b_1
q_2
c_1
c_2
(a) NN search
A_Q
q_1a_1 q_1a_2 q_2a_1 q_2a_2 A_{q2}
B_Q
q_1b_1 q_1b_2 q_2b_1 q_2b_2
C_Q
q_2c_1 q_2c_2 q_1c_1 q_1c_2
C_{q2}
0
distance
(b) Distance Distributions

Figure 3: Examples of S-$\mathcal{SD}$ and SS-$\mathcal{SD}$

Strict Stochastic SD (SS-$\mathcal{SD}$, opt. w.r.t. $\mathcal{N}_{1,2}$). NN functions in $\mathcal{N}_2$ apply the possible world semantics. Generally, each possible world contains one instance from each object and the query. Then, the rank of an object in a possible world can be calculated following the traditional NN semantics. The final score can be derived for each object based on their performance in all possible worlds.

Since each possible world contains exactly one instance from each object and the query, two instance pairs q_1a_1 and q_2c_1 in Figure 3(b) will never be evaluated in the same possible world because q_1 and q_2 cannot appear in the same possible world. This may lead to the case in which an object C is an NN object while C is stochastically dominated by another object A w.r.t. Q. In Figure 3, we have S-$\mathcal{SD}(A,C,Q)$. However, note that C is always closer to q_2 than A and B (i.e., C beats A and B in half of all 16 possible worlds) and hence the NN probability of the object C is 0.5, which is larger than that of A and B (0.375 and 0.125 respectively). Therefore, C is the NN object if the NN probability based NN function is used, which indicates that S-$\mathcal{SD}$ does not cover the possible world based NN functions (i.e., $\mathcal{N}_2$). This motivates us to propose Strict Stochastic SD (SS-$\mathcal{SD}$) operator that covers all functions in $\mathcal{N}_1 \cup \mathcal{N}_2$ (denoted as $\mathcal{N}_{1,2}$).

SS-$\mathcal{SD}$ enforces the stochastic order for each individual query instance $q \in Q$; that is, SS-$\mathcal{SD}(U,V,Q)$ holds if and only if $U_q \preceq_{st} V_q$ for every $q \in Q$ where U_q (V_q) is the distance distribution of U (V) w.r.t. q. In Section 4.2, we show that SS-$\mathcal{SD}$ is optimal w.r.t. $\mathcal{N}_{1,2}$. In the example of Figure 3, we have SS-$\mathcal{SD}(A,B,Q)$, and $\neg$SS-$\mathcal{SD}(A,C,Q)$.

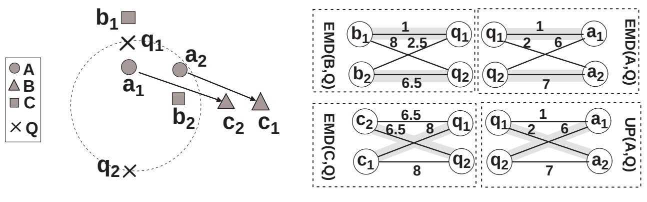


Figure 4: Examples of P-$\mathcal{SD}$

Peer SD (P-$\mathcal{SD}$, opt. w.r.t. $\mathcal{N}_{1,2,3}$). A function in $\mathcal{N}_3$ selects a *subset* of pair-wise distances for computation. Earth Movers Distance (EMD) [10] is an example of such functions. In Figure 4, we assume each object (query) has two instances with probability 0.5, and a distance distribution can be regarded as a fully connected bipartite graph where the label of an edge represents the distance between two vertexes (instances). In this example, EMD(A,Q) corresponds to the minimal weighted sum of any two *disjoint* edges in the bipartite graph of A and Q, where the weight of each edge is 0.5. For instance, two shaded edges in EMD(A,Q) (i.e., $\langle a_1, q_1\rangle$ and $\langle a_2, q_2\rangle$) represent the subset of pairs chosen for the computation. Thus, we have EMD(A,Q)$= 0.5 \times 1 + 0.5 \times 7 = 4$ and EMD(B,Q) $= 0.5 \times 1 + 0.5 \times 6.5 = 3.75$. Although both S-$\mathcal{SD}(A,B,Q)$ and SS-$\mathcal{SD}(A,B,Q)$ hold in Figure 4, we have EMD(A,Q)>EMD(B,Q). This implies that both S-$\mathcal{SD}$ and SS-$\mathcal{SD}$ do not cover $\mathcal{N}_3$.

To accommodate the functions in $\mathcal{N}_3$, we propose the *peer spatial dominance* (P-$\mathcal{SD}$) operator. We say P-$\mathcal{SD}(U,V,Q)$ holds if there exists a one-to-one mapping between two objects U and V such that, for every $u \in U$, u is not further than v w.r.t. all query instances, denoted by $u \preceq_Q v$, where v is the *peer* instance of u (i.e., u is mapped to v). As shown in Figure 4, there is a mapping: $a_1 \to c_2$ and $a_2 \to c_1$ such that $a_1 \preceq_Q c_2$ and $a_2 \preceq_Q c_1$. Thus, P-$\mathcal{SD}(A,C,Q)$ holds. Meanwhile, we have $\neg$P-$\mathcal{SD}(A,B,Q)$. Section 4.2 shows that P-$\mathcal{SD}$ is *optimal* w.r.t. $\mathcal{N}_{1,2,3}$.

Comparison of SD operators. Section 4.1 shows that the four spatial dominance operators are closely related in the sense that we have F-$\mathcal{SD} \subset$ P-$\mathcal{SD} \subset$ SS-$\mathcal{SD} \subset$ S-$\mathcal{SD}$. Here, $\mathcal{SD}_1 \subset \mathcal{SD}_2$ denotes that $\mathcal{SD}_1$ is *covered* by $\mathcal{SD}_2$; that is, $\mathcal{SD}_1$ has stronger dominance condition which may result in more NN candidates as well as larger coverage of NN functions. Figure 5 illustrates the NN candidates and their coverage on different dominance operators, which suggests that: (*i*) F-$\mathcal{SD}$ includes redundant objects (i.e., is not *complete*) regarding the NN function families studied in this paper since P-$\mathcal{SD}$ has the same coverage but smaller candidate size, and (*ii*) users may make a trade-off between the NN candidate size and coverage among P-$\mathcal{SD}$, SS-$\mathcal{SD}$, and S-$\mathcal{SD}$. Particularly, as shown in Section 4, the P-$\mathcal{SD}$ can be applied to current popular NN ranking mechanisms on objects with multiple instances ($\mathcal{N}_{1,2,3}$). If users are only interested in NN functions in $\mathcal{N}_{1,2}$, SS-$\mathcal{SD}$ can be used to reduce the number of candidate objects. The candidate size can be further reduced if only the simple NN functions in $\mathcal{N}_1$

are considered, where S-$\mathcal{SD}$ will be applied. Moreover, Section 4.1 shows that P-$\mathcal{SD}$, SS-$\mathcal{SD}$, and S-$\mathcal{SD}$ are equivalent to each other when the query has only one instance.

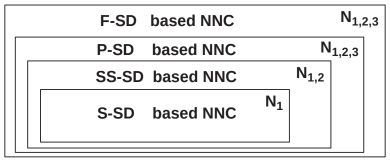


Figure 5: NN Candidates w.r.t. Different SDs

NN Candidates (NNC) Computation. Given a spatial dominance operator $\mathcal{SD}$, the NN candidates are the objects which are not dominated by any other objects. Regarding S-$\mathcal{SD}$ operator, NNC in the example of Figure 3 is $\{A\}$ whereas NNC is $\{A, C\}$ if SS-$\mathcal{SD}$ is employed. Similarly, NNC is $\{A\}$ and $\{A, B\}$ if SS-$\mathcal{SD}$ and P-$\mathcal{SD}$ are employed in Figure 4, respectively.

In addition to the semantics and theoretical properties of the spatial dominance operators, it is imperative to develop efficient dominance check algorithms as well as the NN candidate computation algorithms. In particular, we show that the dominance check of S-$\mathcal{SD}$ and SS-$\mathcal{SD}$ can be finished with one single scan of the pair-wise distances if they are already sorted. Moreover, efficient statistic-based and cover-based filtering techniques are proposed to enhance the performance. Regarding the dominance check of P-$\mathcal{SD}$, the problem can be reduced to the well known max-flow problem [11]. Besides the effective pruning and verification rules, we further improve the performance by utilizing some interesting geometric properties. We also propose a general framework for the computation of NN candidates regarding four spatial dominance operators.

Contributions. Our principle contributions in this paper can be summarized as follows.

- This is the first work to systematically investigate the problem of NN candidate search for objects with multiple instances with careful consideration of NN functions.
- Three families of NN functions are formalized, including all pairs based NN functions ($\mathcal{N}_1$), possible world based NN functions ($\mathcal{N}_2$), and selected pairs based NN functions ($\mathcal{N}_3$). We show that they cover popular NN ranking mechanisms.
- We advocate three *spatial dominance* (SD) operators, namely *stochastic* SD (S-$\mathcal{SD}$), *strict stochastic* SD (SS-$\mathcal{SD}$), and *peer* SD (P-$\mathcal{SD}$). We show that S-$\mathcal{SD}$, SS-$\mathcal{SD}$ and P-$\mathcal{SD}$ are *optimal* dominance operators w.r.t. $\mathcal{N}_1$, $\mathcal{N}_{1,2}$, and $\mathcal{N}_{1,2,3}$, respectively.
- Efficient dominance testing algorithms are proposed for three spatial dominance operators. Novel pruning and validation rules are developed to significantly speed up the computation. Then a general framework is proposed to efficiently calculate the NN candidates.
- Comprehensive experiments demonstrate the effectiveness and efficiency of our NN candidate search techniques.

Roadmap. The rest of the paper is organized as follows. Section 2 introduces the problem studied in this paper. Section 3 provides three general models to summarize the existing NN functions. Section 4 studies important properties of the spatial dominance operators proposed in this paper. Efficient dominance check algorithms as well as NN candidate computation algorithms are proposed in Section 5. The experimental results are reported in Section 6. We conclude the paper in Section 7. In the appendix of the paper, we introduce the details of some NN functions (Section A), details of the proof (Section B), additional experiments and analysis (Section C) and related works (Section D).

2. BACKGROUND

In this section, we provide formal definitions of the spatial dominance operators as well as some important notations. Table 1 summarizes the notations frequently used throughout the paper.

Notation	Meaning
U, V, A, B	objects with multiple instances
u, v, a, b	instances (points)
$Q(q)$	query object (instance)
$\|U\|(\|Q\|)$	number of instances in U (Q)
$p(u)$	probability of an instance u
$\Upsilon(U)$	ranking score of U
$M_{U,V}$	a match between U and V
$\mathcal{M}_{U,V}$	all possible matches between U and V
$u \preceq_Q v$	u is not further than v w.r.t. every $q \in Q$
U_Q	distance distribution of U w.r.t. Q
U_q	distance distribution of U w.r.t. q
X, Y	univariate random variables
$X \preceq_{st} Y$	X is smaller than Y in stochastic order
$X \preceq_M Y$	X is smaller than Y in match order
W, $\mathcal{W}$	a possible world, all possible worlds
$\mathcal{W}_q$	possible worlds in which q occurs
$\mathcal{SD}$	spatial dominance (SD) operator
$\mathcal{SD}_1 \subset \mathcal{SD}_2$	$\mathcal{SD}_1$ is *covered* by $\mathcal{SD}_2$
$\mathcal{SD}$(U,V,Q)	U *spatially dominates* V regarding Q
$\neg\mathcal{SD}$(U,V,Q)	$\mathcal{SD}$(U,V,Q) does not hold

Table 1: The summary of notations.

2.1 Problem Definition

A point (instance) p is in a d-dimensional space and the i-th dimensional coordinate value of p is denoted by $p[i]$. $\delta(u, v)$ denotes the distance between two instances u and v. Although we assume that $\delta(u, v)$ represents Euclidean distance between u and v, our techniques can be trivially extended to other metric distances. We use $\delta_{min}(x, S)$ to denote the minimal distance between a point x and a set S of points, i.e., $\delta_{min}(x, S) = min_{y \in S}\, \delta(x, y)$.

Modeling object with multiple instances. In this paper, we model an object with multiple instances as a **discrete random variable** (i.e., discrete uncertain object). Particularly, an object U consists of a set $\{u_1, u_2, \ldots, u_m\}$ of instances (points), and a *discrete probability mass function* assigns each instance u_i a probability value, denoted by $p(u_i)$. In this paper, we assume $\sum_{i=1}^{m} p(u_i) = 1$, and objects are independent to each other. Moreover, we assume the weights of the multi-valued objects can be normalized to probabilities where $p(u_i) = \frac{w(u_i)}{\sum_{j=1}^{m} w(u_j)}$ and $w(u_j)$ is the weight of the instance u_j so that a multi-valued object is also treated as a discrete random variable. We remark that the transformation from multi-valued objects to discrete uncertain objects is only for the dominance check purpose (i.e., NN candidates computation), and will not introduce any new NN semantics (i.e., new NN functions). We use the term *object* to refer to the object with multiple instances whenever clear by context.

Distance distribution. Given an object U and a query Q, the distance distribution between U and Q, denoted by U_Q,

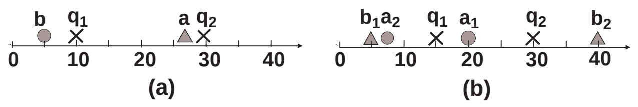


Figure 6: Examples for S-$\mathcal{SD}$ and SS-$\mathcal{SD}$

is a discrete random variable consisting of all pair-wise distances. Specifically, U_Q consists of every instance pair (q, u) for every $q \in Q$ and every $u \in U$, where each instance pair (q, u) is assigned two values: $\delta(q, u)$ denotes the distance and $p(q) \times p(u)$ denotes the probability of the instance pair. For a query instance $q \in Q$, the distance distribution between U and q, denoted by U_q, consists of only the instance pairs where q is involved; that is, U_q contains every (q, u) for all $u \in U$ where each pair (q, u) is associated with $\delta(q, u)$ denoting the distance and $p(u)$ denoting the probability of the instance pair.

Example 1. *In Figure 6(b), both the query Q and the object A have two instances with the same probability 0.5. Then the distance distribution A_Q is $\{(\delta(q_1, a_1), 0.25), (\delta(q_1, a_2), 0.25), (\delta(q_2, a_1), 0.25), (\delta(q_2, a_2), 0.25)\} = \{(5, 0.25), (8, 0.25), (10, 0.25), (23, 0.25)\}$. Meanwhile, the distance distribution between the query instance q_1 and the object A, A_{q_1}, is $\{(\delta(q_1, a_1), 0.5), (\delta(q_1, a_2), 0.5)\} = \{(5, 0.5), (8, 0.5)\}$.*

Our first two spatial dominance operators are proposed based on the **stochastic order** [29] which has been widely used in various domains to compare the "goodness" of two random variables (distributions).

Definition 1. ***Stochastic Order.*** *Given two independent random variables X and Y, we say X is smaller than Y in usual stochastic order, denoted by $X \preceq_{st} Y$, if $Pr(X \le \lambda) \ge Pr(Y \le \lambda)$ for every $\lambda \in R$.*

Then we immediately have the definitions of two stochastic order based dominance operators.

Definition 2. ***Stochastic SD(S-$\mathcal{SD}$).*** *Given two objects U and V, and the query Q, we have S-$\mathcal{SD}$(U,V,Q) if and only if $U_Q \preceq_{st} V_Q$ and $U_Q \neq V_Q$.*

Definition 3. ***Strict Stochastic SD (SS-$\mathcal{SD}$).*** *Given two objects U and V, and the query Q, we have SS-$\mathcal{SD}$(U,V,Q) if and only if $U_q \preceq_{st} V_q$ for every $q \in Q$ and $U_Q \neq V_Q$.*

Note that the condition $U_Q \neq V_Q$ is used to avoid the scenario that two objects dominate each other.

Example 2. *We assume all instances of the same object (query) has the same probability value. In Figure 6(a), each of objects A and B have one instance and the query Q has two instances q_1 and q_2. Since $A_Q = \{(3, 0.5), (17, 0.5)\}$ and $B_Q = \{(5, 0.5), (25, 0.5)\}$ we have S-$\mathcal{SD}$(A,B,Q). Nevertheless, since $A_{q_1} = \{(17, 1.0)\}$ and $B_{q_1} = \{(5, 1.0)\}$, we have $\neg$SS-$\mathcal{SD}$(A,B,Q). In Figure 6(b), since $A_{q_1} = \{(5, 0.5), (8, 0.5)\}$, $A_{q_2} = \{(10, 0.5), (23, 0.5)\}$, $B_{q_1} = \{(10, 0.5), (25, 0.5)\}$, $B_{q_2} = \{(10, 0.5), (25, 0.5)\}$, we have $A_{q_1} \preceq_{st} B_{q_1}$ and $A_{q_2} \preceq_{st} B_{q_2}$, and hence SS-$\mathcal{SD}$(A,B,Q) holds.*

Before we formally define the third dominance operator, Peer SD (P-$\mathcal{SD}$), we first extend the semantics of *one-to-one mapping* for the general case where different objects may have different number of instances and instance probabilities. We formally define a *match* between two discrete random variables (e.g., objects and distance distributions), which can actually be regarded as a one-to-one mapping by splitting some instances.

Definition 4. ***Match*** *($M_{U,V}$). Given two discrete random variables U and V, a match between U and V, denoted by $M_{U,V}$, consists of a set of tuples $\{t\langle u, v, p\rangle\}$ where $t.u \in U$, $t.v \in V$ and $t.p$ is the probability associated with t. Moreover, we have $\sum_{t \in M_{U,V} \wedge t.u = u} t.p = p(u)$ and $\sum_{t \in M_{U,V} \wedge t.v = v} t.p = p(v)$.*

In this paper, we use $\mathcal{M}_{U,V}$ to denote all possible matches between U and V. Figure 7 shows two objects A=$\{(a_1, 0.5), (a_2, 0.3), (a_3, 0.2)\}$ and B=$\{(b_1, 0.5), (b_2, 0.5)\}$. Each shadowed edge corresponds to one tuple in the match with probability labeled, e.g., Figure 7(a) and Figure 7(b) show two matches $\{\langle a_1, b_1, 0.5\rangle, \langle a_2, b_2, 0.3\rangle, \langle a_3, b_2, 0.2\rangle\}$ and $\{\langle a_1, b_1, 0.2\rangle, \langle a_1, b_2, 0.3\rangle, \langle a_2, b_1, 0.3\rangle, \langle a_3, b_2, 0.2\rangle\}$, respectively. According to Definition 4, the assignment in Figure 7(c) is not a match.

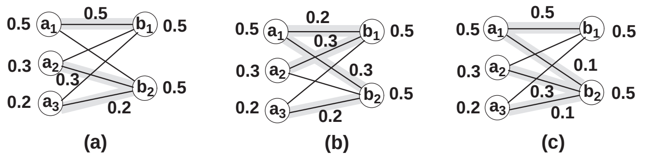


Figure 7: Example of Matches

We say that a point u is *closer* to the query Q than v, denoted by $u \preceq_Q v$, if we have $\delta(u, q) \le \delta(v, q)$ for every $q \in Q$. Following is a formal definition of P-$\mathcal{SD}$.

Definition 5. ***Peer SD (P-$\mathcal{SD}$).*** *Given two objects U and V, and query Q, we have P-$\mathcal{SD}$(U,V,Q) if and only if there is a match $M_{U,V}$ such that $t.u \preceq_Q t.v$ for every $t \in M_{U,V}$, and $U_Q \neq V_Q$.*

Example 3. *In Figure 8, given the match between A and B $M_{A,B} = \{(a_1, b_1, 0.5), (a_2, b_2, 0.5)\}$, we have $a_1 \preceq_Q b_1$ and $a_2 \preceq_Q b_2$ since $\delta(a_1, q_1) = 5 < 10 = \delta(b_1, q_1)$, $\delta(a_1, q_2) = 15 < 20 = \delta(b_1, q_2)$, $\delta(a_2, q_1) = 20 < 25 = \delta(b_2, q_1)$, $\delta(a_2, q_2) = 10 < 15 = \delta(b_2, q_2)$. Consequently, P-$\mathcal{SD}$(A,B,Q) holds.*

Next, we formally define NN candidates and the *cover* relationship among spatial dominance operators.

Definition 6. ***NN Candidates (NNC).*** *Given a set $\mathcal{O}$ of objects, a query Q and a spatial dominance operator SD, the nearest neighbor candidates, denoted by NNC($\mathcal{O}$,Q,SD), consist of every object that is not dominated by any other object w.r.t. SD.*

Definition 7. ***Cover.*** *Given two spatial dominance operators $\mathcal{SD}_1$ and $\mathcal{SD}_2$, we say $\mathcal{SD}_2$* ***covers*** *$\mathcal{SD}_1$, denoted by $\mathcal{SD}_1 \subset \mathcal{SD}_2$, if $\mathcal{SD}_1$(U,V,Q) implies $\mathcal{SD}_2$(U,V,Q) but the converse does not hold.*

Note that $\mathcal{SD}_1 \subset \mathcal{SD}_2$ implies that the dominance condition of $\mathcal{SD}_1$ is *stronger* than that of $\mathcal{SD}_2$, and hence NNC($\mathcal{O}$,Q,$\mathcal{SD}_2$) $\subseteq$ NNC($\mathcal{O}$,Q,$\mathcal{SD}_1$). In Section 4.1, we show that F-$\mathcal{SD}$ $\subset$ P-$\mathcal{SD}$ $\subset$ SS-$\mathcal{SD}$ $\subset$ S-$\mathcal{SD}$. Based on the relations among four dominance operators and the definition of NNC, as shown in Figure 5, we have NNC($\mathcal{O}$,Q,S-$\mathcal{SD}$)

$\subseteq$ NNC($\mathcal{O}$,Q,SS-$\mathcal{SD}$) $\subseteq$ NNC($\mathcal{O}$,Q,P-$\mathcal{SD}$) $\subseteq$ NNC(Q,$\mathcal{O}$,F-$\mathcal{SD}$). As shown in Section 4.1, P-$\mathcal{SD}$, SS-$\mathcal{SD}$ and S-$\mathcal{SD}$ are equivalent to each other when the query has only one instance.

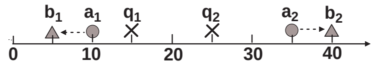


Figure 8: Examples for P-$\mathcal{SD}$

3. MODELING NN FUNCTIONS

In this section, we first discuss the *stable* property of aggregate functions. Then we propose three general models for three representative families of NN ranking functions (denoted by $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$), which well capture the existing NN ranking mechanisms.

3.1 Stable Aggregate Function

Similar to [12], we argue that the aggregate functions used by NN ranking mechanism should satisfy the *stable* property.

Definition 8. ***Stable Aggregate Function****. Let g denote an aggregate function on a random variable, we say g is a stable aggregate function if and only if for any two random variables X and Y, $X \preceq_{st} Y$ implies that $g(X) \leq g(Y)$.*

We also introduce the *match order* which is essentially another interpretation of *usual stochastic order*. We remark that both orders are frequently used in our theoretical analysis since they interpret the dominance relationship from different perspectives.

Definition 9. ***Match Order****. Given two independent random variables X and Y, we say X is smaller than Y in match order, denoted by $X \preceq_M Y$, if there is a match $M_{X,Y}$ between X and Y with $t.x \leq t.y$ for every $t \in M_{X,Y}$.*

Following theorem indicates that the two orders are equivalent to each other.

Theorem 1. *The match order is equivalent to the usual stochastic order.*

Proof. See Appendix B.1 □

According to Definition 9, given two random variables X and X^* with $X^* \preceq_M X$ (i.e., $X^* \preceq_{st} X$), X^* can be regarded as an enhancement of X by, roughly speaking, pushing some of the probability mass of X to the left (assuming left represents smaller values). It is rather intuitive that we should have $g(X^*) \leq g(X)$ for any reasonable aggregate function used in NN search.

3.2 All pairs based NN ($\mathcal{N}_1$)

Model. A simple and intuitive way to identify the nearest neighbor is to apply an aggregation function on the distance distribution (i.e., all pair-wise distances) of each object. The NN function family $\mathcal{N}_1$ is defined as follows,

$$\mathcal{N}_1 := \{f \mid f(U) = g(U_Q), g \in \mathcal{G}\} \tag{1}$$

where $\mathcal{G}$ represents the family of stable aggregate functions.

Instantiations. In the literature, the aggregate functions such as *min*, *max*, *mean* (a.k.a., *expected*), and *quantile* distances (e.g., [29, 37]) have been widely employed to identify the NN objects. It is immediate that both *min* and *max* are stable aggregate functions. Given $X \preceq_M Y$, there is a match $M_{X,Y}$ with $t.x \leq t.y$ for every $t \in M_{X,Y}$. Since we have $mean(X) = \sum_{t \in M_{X,Y}} t.x \times t.p$, it is immediate that *mean* function is a stable. Below, we can show that *quantile* function also has the stable property.

Definition 10. **ϕ-quantile** *($quan_\phi$). Given a random variable X, and we assume its instances are sorted in non-decreasing order. The ϕ-quantile ($0 < \phi \leq 1$) of X, denoted by $quan_\phi(X)$, is the value of x_i where i is the first instance such that $\sum_{1 \leq j \leq i} p(x_j) \geq \phi$.*

According to the above definition, *ϕ-quantile* is a stable aggregate function since we have $Pr(X \leq quan_\phi(Y)) \geq Pr(Y \leq quan_\phi(Y))$ for any ϕ ($0 < \phi \leq 1$) if $X \preceq_{st} Y$.

Besides the above premier NN functions, we may easily come up with various stable NN functions based on their combinations or some linear weighted functions on distance distribution. Note that the expected distance is an example of the linear weighted function.

3.3 Possible world based NN ($\mathcal{N}_2$)

In recent years, a large body of work have addressed the problem of NN search on uncertain objects following the possible world semantics.

Possible world semantics. Given a set $\mathcal{O} \cup Q$ of objects and the query, a possible world W is a set of instances with one instance from each object in $\mathcal{O} \cup Q$. Let $\mathcal{W}$ denote the set of all possible worlds. Then, $\sum_{W \in \mathcal{W}} Pr(W) = 1$ where $Pr(W)$ is the occurrence probability of the possible world W. Moreover, we use $\delta(U, W)$ to denote the distance between U and Q in the possible world W. By $r(U, W)$ we mean U is ranked at $r(U, W)$-th position according to its distance to the query point in W.

Model. In each possible world W, a score $s(U, W)$ is assigned to each object U which is either determined by $r(U, W)$ or by $\delta(U, W)$. Particularly, we assume that $s(U, W) \leq s(V, W)$ if $\delta(U, W) < \delta(V, W)$ because we are in favor of the object closer to the query in each possible world W. The *score distribution* of U is denoted by $U_{\mathcal{W}}$ = $\{s(U, W), Pr(W)\}$ for all $W \in \mathcal{W}$. Then the set of possible world based NN functions, denoted by $\mathcal{N}_2$, is defined as follows.

$$\mathcal{N}_2 := \{f \mid f(U) = g(U_{\mathcal{W}}), g \in \mathcal{G}\} \tag{2}$$

Instantiations. Jian *et* al. [23] show that the existing popular ranking methods for uncertain data can be unified by the parameterized ranking model. We use $\Upsilon(U)$ to denote the NN score of an object U,

$$\Upsilon(U) = \sum_i \omega(i) \times Pr(r(U) = i) \tag{3}$$

where $Pr(r(U) = i)$ denotes the probability that U is ranked as the i-th closest object and $\omega(i)$ is the weight of the position i. In practice, we have $\omega(i) \leq \omega(j)$ for any two ranking positions i and j with $i < j$ because the higher position is usually at least as desirable as those behind it and thus should be given a smaller weight. As shown in [23], other popular ranking methods can be unified by setting the value of $\omega(i)$ appropriately. In the context of NN search, we have $\omega(i) = -1$ and $\omega(i) = i$ in U-top-k rank model [32] and expected rank model [12], respectively. For global top k model [39], $\omega(i) = -1$ for $i \leq k$, and $\omega(i) = 0$ otherwise. Note that the NN probability function is its special case with $k = 1$.

Now we show that NN functions based on parameterized ranking model belong to $\mathcal{N}_2$. By setting $s(U, W) =$

$\omega(r(U,W))$ and g as the *mean* function on $U_{\mathcal{W}}$, we have $f(U) = \sum_{W \in \mathcal{W}} s(U,W) \times Pr(W) = \sum_i \omega(i) \times Pr(r(U) = i)$ $= \Upsilon(U)$. Clearly, there are infinite number of possible NN functions in $\mathcal{N}_2$ due to various settings of ω values.

We remark that the multi-valued object model should not be applied to the possible world semantics due to the co-exists of the instances from the same object.

3.4 Selected pairs based NN ($\mathcal{N}_3$)

Model. A function $f \in \mathcal{N}_3$ computes the score of an object U based on a *subset of distance distribution* U_Q which is denoted by $\sigma_U(U_Q)$= $\{t\langle\delta(u,q),p\rangle\}$ where $t.u \in U$, $t.q \in Q$ and $t.p$ is the probability value associated with the pair tuple t. Remark that $t.p$ here is not necessarily $p(t.u) \times p(t.q)$ and may take a different value depending on the function $f \in \mathcal{N}_3$. $\sigma_V(U_Q)$ is the counterpart of V w.r.t. U, which will be explained later. Then we have $f(U) = g(\sigma_U(U_Q))$ where g is a stable aggregate function.

The key property of the selected pairs based NN function f is that it must be *counterpart computable*. First, we introduce the concept of *counterpart* and then explain counterpart computable. Given $\sigma_V(V_Q)$ and a match $M_{U,V}$, $\sigma_V(U_Q)$ contains a subset of pairs from U_Q that are selected using $\sigma_V(V_Q)$ and $M_{U,V}$. Informally, if v_i matches to u_j in $M_{U,V}$ and (v_i, q_k) is a pair in $\sigma_V(V_Q)$, then the pair (u_j, q_k) will be inserted in $\sigma_V(U_Q)$. Formally, the tuples in $\sigma_V(U_Q)$ are selected as follows: for each tuple m $\langle\delta(v,q),p\rangle \in \sigma_V(V_Q)$, we find every tuple $t \in M_{U,V}$ for which $t.v = m.v$. Then, for each such t, we create a tuple $m'\langle$ $\delta(t.u, m.q)$, $\frac{t.p \times m.p}{p(v)}\rangle$ and insert m' in $\sigma_V(U_Q)$.

Example 4. *Figure 4(b) illustrates a simple example of counterpart of A w.r.t. C, i.e., $\sigma_C(A_Q)$. We have $\sigma_C(C_Q)$ $= \{\langle\delta(c_1,q_1)$,0.5$\rangle$, $\langle\delta(c_2,q_2)$,0.5$\rangle\}$. Regarding the match $M_{A,C}$ =$\{\langle a_1, c_2$,0.5$\rangle$, $\langle a_2, c_1$,0.5$\rangle\}$ in Figure 4(a), $\sigma_C(A_Q)$ corresponds to shadowed edges in UP(A,Q) where $\sigma_C(A_Q)$ =$\{\langle\delta(a_2,q_1)$,0.5$\rangle$, $\langle\delta(a_1,q_2)$,0.5$\rangle\}$.*

We say that a function is *counterpart computable* if, for every match $M_{U,V}$ between U and V, $g(\sigma_V(U_Q)) \geq g(\sigma_U(U_Q))$, i.e., the score of an object U (i.e., $f(U) = g(\sigma_U(U_Q))$) is no larger than the score of the counterpart of U w.r.t. V. In the example of Figure 4, $\sigma_A(A_Q)$ =$\{\langle\delta(a_1,q_1), 0.5\rangle$, $\langle\delta(a_2,q_2), 0.5\rangle$ $\}$. It can be verified that $g(\sigma_C(A_Q)) > g(\sigma_A(A_Q))$, i.e., EMD is counterpart computable.

Following is a formal definition of NN functions in $\mathcal{N}_3$.

$$\mathcal{N}_3 := \{f \mid f(U) = g(\sigma_U(U_Q)), g \in \mathcal{G}\} \qquad (4)$$

where f is a counterpart computable function.

Instantiations. We show that four popular NN ranking functions belong to $\mathcal{N}_3$ (the detailed definitions are given in Appendix A) . In *Hausdorff distance* and *Sum of Minimal distance* [27], an instance $u \in U$ will pick its closest instance in Q to form an instance-pair. Let q' denote the instance picked for u in the counterpart of U, clearly we have $\delta_{min}(u,Q) \leq \delta(u,q')$. Same observation holds for the instances $q \in Q$. Consequently, both distances are counterpart computable. As their aggregate functions are stable, *Hausdorff distance* and *Sum of Minimal distance* belong to $\mathcal{N}_3$.

Under our problem settings where each object has a probability mass 1, the definitions of Earth Mover's distance [10] and Netflow distance [27] are equivalent to each other. Taking the Earth Mover's distance EMD(U,Q) as an example, a *feasible* solution corresponds to a *match* between U and Q with cost $\sum_{t \in M_{U,Q}} \delta(t.u, t.q) \times t.p$. Given the pair-wise distances chosen in EMD(V,Q) (i.e.,$\sigma_V(V_Q)$) and a match $M_{V,U}$, it is easy to see that the counterpart in U_Q, i.e., $\sigma_V(U_Q)$, corresponds to a feasible solution with cost not less than EMD(U,Q), because EMD(U,Q) represents the minimal cost of all feasible solutions. Consequently, Earth Mover's distance and Netflow distance are both *counterpart computable.*

4. PROPERTIES OF SD OPERATORS

In this section, we investigate important properties of the four spatial dominance operators and their relationships.

4.1 Comparison of SD Operators

This subsection investigates the "cover" relationships among four spatial dominance operators.

Theorem 2. *F-$\mathcal{SD}$ $\subset$ P-$\mathcal{SD}$ $\subset$ SS-$\mathcal{SD}$ $\subset$ S-$\mathcal{SD}$.*

Proof. See Appendix B.2. □

Theorem 2 is not only of theoretical interest but it also enables effective pruning and validation rules to speed up the dominance check, as shown in Section 5. Particularly, given $\mathcal{SD}_1 \subset \mathcal{SD}_2$, we can safely claim $\mathcal{SD}_2(U,V,Q)$ holds (i.e., **validation**) if $\mathcal{SD}_1(U,V,Q)$ is verified. Similarly, we have $\neg\mathcal{SD}_1(U,V,Q)$ if $\neg\mathcal{SD}_2(U,V,Q)$ (i.e.,**pruning**).

The following theorem suggests that P-$\mathcal{SD}$, SS-$\mathcal{SD}$ and S-$\mathcal{SD}$ are equivalent to each other for the special case when the query has only one instance, which is not uncommon in practice. However, F-$\mathcal{SD}$ still has stronger dominance check condition even for this special case.

Theorem 3. *When $|Q| = 1$, we have F-$\mathcal{SD}$ $\subset$ P-$\mathcal{SD}$ $=$ SS-$\mathcal{SD}$ $=$ S-$\mathcal{SD}$.*

Proof. See Appendix B.3. □

In [16], efficient filtering technique is proposed to check F-$\mathcal{SD}$ against the minimal bounding rectangles (MBR) of the objects in $O(d)$ time, where d is the dimensionality. The following theorem suggests that we can directly apply the technique in [16] for validation purpose. The correctness is immediate based on the definition of F-$\mathcal{SD}$ and Theorem 2.

Theorem 4. *F-$\mathcal{SD}$(U_{mbr},V_{mbr},Q_{mbr}) implies that we have F-$\mathcal{SD}$(U,V,Q), P-$\mathcal{SD}$(U,V,Q), SS-$\mathcal{SD}$(U,V,Q), and S-$\mathcal{SD}$(U,V,Q).*

Clearly, filtering technique in [25] may also be applied if objects are approximated by hyperspheres.

4.2 Optimality Property

In this subsection, we show that three spatial dominance operators proposed in this paper are optimal w.r.t. $\mathcal{N}_1$, $\mathcal{N}_{1,2}$ and $\mathcal{N}_{1,2,3}$ respectively.

Theorem 5. *S-$\mathcal{SD}$ is optimal w.r.t. $\mathcal{N}_1$, and S-$\mathcal{SD}$ does not cover $\mathcal{N}_{2,3}$.*

Proof. See Appendix B.4. □

Theorem 6. *SS-$\mathcal{SD}$ is optimal w.r.t. $\mathcal{N}_{1,2}$, and SS-$\mathcal{SD}$ does not cover $\mathcal{N}_3$.*

Proof. See Appendix B.5. □

Theorem 7. *P-$\mathcal{SD}$ is optimal w.r.t. $\mathcal{N}_{1,2,3}$.*

Proof. See Appendix B.6. □

The following theorem indicates that although both P-$\mathcal{SD}$ and F-$\mathcal{SD}$ cover functions in $\mathcal{N}_{1,2,3}$, F-$\mathcal{SD}$ is not as tight as P-$\mathcal{SD}$ since it does not have the completeness property. This results in redundant NN candidate objects w.r.t. functions in $\mathcal{N}_{1,2,3}$.

Theorem 8. *F-$\mathcal{SD}$ is correct w.r.t. $\mathcal{N}_{1,2,3}$, but does not satisfy the completeness property w.r.t. $\mathcal{N}_{1,2,3}$.*

Proof. See Appendix B.7 □

4.3 Transitivity Property

Given three objects $\{U,V,Z\}$ and the query Q, we say that a dominance operator $\mathcal{SD}$ has the transitivity property if $\mathcal{SD}(U,V,Q)$ and $\mathcal{SD}(V,Z,Q)$ imply $\mathcal{SD}(U,Z,Q)$. The following theorem shows that all four spatial dominance operators have the transitivity property, which is critical in the computation of NN candidates in Section 5.

Theorem 9. *S-$\mathcal{SD}$, SS-$\mathcal{SD}$, P-$\mathcal{SD}$ and F-$\mathcal{SD}$ have the transitivity property.*

Proof. See Appendix B.8 □

5. NN CANDIDATES COMPUTATION

In this Section, we develop efficient spatial dominance based NN candidate computation algorithms.

5.1 Spatial Dominance Check

In this subsection, we present dominance check algorithms for three spatial dominance operators, S-$\mathcal{SD}$, SS-$\mathcal{SD}$ and P-$\mathcal{SD}$ as well as effective filtering techniques.

5.1.1 S-$\mathcal{SD}$ and SS-$\mathcal{SD}$

The dominance check algorithms for S-$\mathcal{SD}$ and SS-$\mathcal{SD}$ are very efficient if pair-wise distances are sorted in non-decreasing order. By recording the differentiation of the accumulated instance probabilities of two objects, denoted by $F(x) = Pr(U_Q \leq x) - Pr(V_Q \leq x)$, we terminate the search and claim $\neg$S-$\mathcal{SD}(U,V,Q)$ if there exists an x such that $F(x) < 0$, otherwise we have S-$\mathcal{SD}(U,V,Q)$. Similarly, we maintain $|Q|$ indicators for the distance distributions of the query instances, and claim $\neg$SS-$\mathcal{SD}$ (U,V,Q) if any of them becomes negative. Suppose instances of each object are organized by an R-tree, we may easily extend the above algorithms to conduct dominance check in a level-by-level fashion.

Time Complexity of the dominance check is $O(m \times |Q| \log(m \times |Q|))$ for both S-$\mathcal{SD}$ and SS-$\mathcal{SD}$ where m ($|Q|$) is the average number of instances in the object (query). Following theorem implies that our dominance check algorithms for S-$\mathcal{SD}$ and SS-$\mathcal{SD}$ are optimal in the worst case where the distance distribution has $m \times |Q|$ instances.

Theorem 10. *Given two random variables X and Y with n instances each, any algorithm to determine whether $X \preceq_{st} Y$ must have complexity of at least $\Omega(n \log(n))$ if the algorithm is based only on comparisons between elements from X and Y.*

Proof. See Appendix B.9 □

Below we introduce efficient and effective pruning and validation techniques, which avoid exploring all pair-wise distances.

Pruning. Theorem below indicates that we may claim that the dominance relation does not hold (i.e., *pruning*) based on some simple statistic values of two distance distributions.

Theorem 11. *Given two random variables X and Y, $X \preceq_{st} Y$ implies that $min(X) \leq min(Y)$, $mean(X) \leq mean(Y)$ and $max(X) \leq max(Y)$.*

Proof. Since three functions (*min*, *max* and *mean*) are *stable* functions as shown in Section 3.2. The correctness of is immediate according to Definition 8. □

Then we have the following statistic-based pruning rule.

<u>*Statistic-based Pruning*</u>. If $min(U_Q) > min(V_Q)$, $mean(U_Q) > mean(V_Q)$, or $max(U_Q) > max(V_Q)$, we have $\neg$S-$\mathcal{SD}(U,V,Q)$.

Similar pruning rule goes to SS-$\mathcal{SD}$ with the same rationale where statistics are kept for every query instance.

On the other hand, as shown in Definition 7, we have $\neg\mathcal{SD}_1$ implies $\neg\mathcal{SD}_2$ if $\mathcal{SD}_2$ covers $\mathcal{SD}_1$. The following pruning rule is immediate based on Theorem 5.

<u>*Cover-based Pruning*</u>. We have $\neg$SS-$\mathcal{SD}(U, V, Q)$ if $\neg$S-$\mathcal{SD}(U, V, Q)$.

Validation. According to Theorem 4, we may utilize the efficient filtering technique in [16] to validate the dominance based on the minimal bounding rectangles of the objects and the query.

<u>*Cover-based Validation*</u>. We have S-$\mathcal{SD}(U, V, Q)$ and SS-$\mathcal{SD}(U, V, Q)$ if F-$\mathcal{SD}(U_{mbr}, V_{mbr}, Q_{mbr})$ holds.

5.1.2 P-$\mathcal{SD}$

Given two objects U and V, a straightforward way to determine P-$\mathcal{SD}(U,V,Q)$ need to consider all possible matches between U and V which is cost-prohibitive because we may have to verify up to $m!$ possible matches. In this paper, we first show that P-$\mathcal{SD}$ check can be reduced to the max-flow problem [11], then a set of techniques are proposed to enhance the performance.

Let N (E) denote a set of vertices (edges) and C records the capacities of the edges, G=<N,E,C> is a network with s , $t \in N$ being *source* and *sink* respectively. A feasible flow f of the network G maps each edge $e \in E$ a non-negative value $f(e)$ following the capacity constraint and conservation of flows constraint [11]. The value $|f|$ of the flow is defined as $|f| = \sum_{\langle s,x\rangle \in E} f(\langle s,x\rangle)$. Then the network for two objects U and V w.r.t. a query Q, denoted by $G_{U,V}$, can be constructed as follows.

- Vertices s and t for source and sink respectively.
- Each instance $u \in U$ contributes a vertex u and an edge <s,u> with $c(s,u) = p(u)$.
- Each instance $v \in V$ contributes a vertex v and an edge <u,t> with $c(v,t) = p(v)$.
- For any two instances $u \in U$ and $v \in V$, there is an edge $\langle u,v\rangle$ with $c(u,v) = \infty$ if $u \preceq_Q v$.

The following theorem indicates that we can apply the max-flow algorithm to check Peer spatial dominance.

Theorem 12. *Let $G_{U,V}$ be the network constructed based on two objects U, V and the query Q, we have P-$\mathcal{SD}(U,V,Q)$ if and only if $|f^*| = 1$ where f^* is a max-flow of $G_{U,V}$.*

Proof. See Appendix B.10 □

Example 5. *Figure 9 illustrates an example of P-$\mathcal{SD}$ check, which is reduced to max-flow problem. In Figure 9(a),*

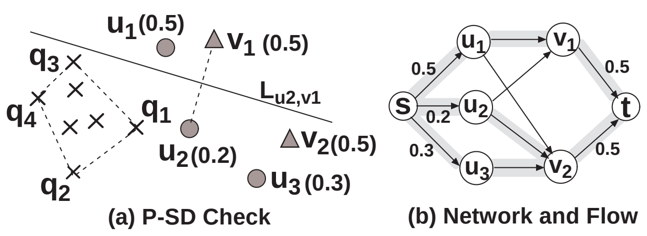


Figure 9: Example of P-$\mathcal{SD}$ Check

the object U (V) consists of 3 (2) instances and the probability of each instance is labeled. Since we have $u_1 \preceq_Q v_1$, $u_1 \preceq_Q v_2$, $u_2 \preceq_Q v_1$, $u_2 \preceq_Q v_2$, and $u_3 \preceq_Q v_2$, the corresponding network $G_{U,V}$ is depicted in Figure 9(b) where the capacities of the edges are labeled. Note that the edges unlabeled have capacity ∞. There is a match $M_{U,V} = \{\langle u_1, v_1, 0.5\rangle, \langle u_2, v_2, 0.2\rangle, \langle u_3, v_2, 0.3\rangle\}$, which justifies that P-$\mathcal{SD}(U,V,Q)$. The corresponding flow will go through the shaded edges in Figure 9(b).

Time Complexity. It takes $O(d|Q|)$ time to check if an instance $u \in U$ can be assigned to $v \in V$. So the network construction time is $O(d|Q|m^2)$ where m is the average number of instances per object. Let $|E|$ denote the number of edges in the network, the max-flow computation takes $O(m|E|\log(|E|))$ time [11].

Pruning. Since P-$\mathcal{SD}$ is covered by S-$\mathcal{SD}$ and SS-$\mathcal{SD}$, we have the following cover-based pruning rule.

<u>*Cover-based Pruning*</u>. We have $\neg$F-$\mathcal{SD}(U,V,Q)$ if $\neg$S-$\mathcal{SD}(U,V,Q)$ or $\neg$SS-$\mathcal{SD}(U,V,Q)$.

Validation. According to Theorem 4, we have the following cover-based validation rule.

<u>*Cover-based Validation*</u>. We have P-$\mathcal{SD}(U,V,Q)$ if F-$\mathcal{SD}(U_{mbr},V_{mbr},Q_{mbr})$ holds.

Utilizing Geometrical Properties. As observed in some recent work (e.g., [30, 15]) $u \preceq_Q v$ implies that all query instances fall on the left side (i.e., the same side with u) of the half-space divided by the bisector between u and v. In Figure 9(a), we have $u_2 \preceq_Q v_1$ since all query instances reside on the left side of the bisector line L_{u_2,v_1}. Consequently, we only need to consider query points on the convex hull of the query, denoted by $\mathcal{CH}(Q)$. In the example of Figure 9(a), only 4 query points need to be involved in the dominance check. Moreover, it is immediate that P-$\mathcal{SD}(U,V,Q)$ does not hold for any object U if an instance of V falls in the convex hull of the query Q.

Suppose there are k query points in $\mathcal{CH}(Q)$, then each instance u can be mapped to a point u' in a k-dimensional space with $u[i] = \delta(u, q_i)$. For each instance $v \in V$, we may issue a rectangular range query with lower corner $(0, \ldots, 0)$ and upper corner v', and we have $u \preceq v$ if u' is contained by the range search. By taking advantage of the efficient range search in spatial indexing techniques (e.g., R-Tree), we can efficiently improve the network construction time.

Level-by-level Pruning/Validation. Instead of directly testing P-$\mathcal{SD}(U,V,Q)$ against the instances of U and V, we may conduct the testing in a top down manner such that the computation may terminate at a high level, and hence significantly reduce the computational costs.

Let U^i denote a set of instances of U with minimal bounding rectangle U^i_{mbr}, and $p(U^i)$ represents the total probability mass of the instances in U^i. We assume U consists of k disjoint sets $U^1, \ldots, U^k$. Similarly we have $V = \{V^1, \ldots, V^k\}$. By regarding each set U^i (V^i) as a virtual instance, we may conduct dominance check on these virtual instances for *validation* purpose. In particular, since F-$\mathcal{SD}(U^i_{mbr}, V^j_{mbr}, Q_{mbr})$ implies that for every $q \in Q$, we have $u \preceq_Q v$ for every $u \in U^i$ and $v \in V^j$. There is an edge from U^i to V^j if F-$\mathcal{SD}(U^i_{mbr}, V^j_{mbr}, Q_{mbr})$ holds. Let $G^-_{U,V}$ denote the corresponding network, we have $|f^-| \leq |f|$ where $|f^-|$ and $|f|$ are max-flow of $G^-_{U,V}$ and $G_{U,V}$, respectively. Consequently, we can claim P-$\mathcal{SD}(U,V,Q)$ if $|f^-| = 1$. Regarding the *pruning*, an edge is assigned from U^i to V^j if $\neg$F-$\mathcal{SD}(V^j_{mbr},U^i_{mbr},Q_{mbr})$. This is because $\neg$F-$\mathcal{SD}(V^j_{mbr},U^i_{mbr},Q_{mbr})$ indicates that we have chance to find $u \in U^i$ and $v \in V^j$ such that $u \preceq_Q v$. Let $G^+_{U,V}$ denote the corresponding network, we have $|f^+| \geq |f|$ where $|f^+|$ and $|f|$ are max-flow of $G^+_{U,V}$ and $G_{U,V}$, respectively. Thus, we have $\neg$P-$\mathcal{SD}(U,V,Q)$ if $|f^+| < 1$.

Suppose instances of U and V are organized by hierarchical structure (e.g., R-Tree), we can iteratively conduct the above pruning and validation procedures in a level-by-level fashion until the algorithm is terminated at high level or instances of the objects are reached.

5.2 NN candidate computation

In this subsection, we introduce the NN candidates (NNC) computation algorithm for a given spatial dominance operator $\mathcal{SD}$.

Suppose MBRs of the objects are organized by a *global* R-Tree $R_{\mathcal{O}}$, Algorithm 1 outlines our NNC computation algorithm. By maintaining a min heap H, entries and objects are visited in a non-decreasing ordering based on their minimal distance to the query Q. In Line 5-9, each visited object V will be evaluated against existing NNC objects based on the dominance checking algorithms proposed in Section 5.1. Then V becomes an NN candidate if it survives the dominance check (Line 10-11). According to the cover-based verification rules, an entry e (i.e., objects associated with e) is discarded if F-$\mathcal{SD}(U_{mbr},e_{mbr},Q_{mbr})$ holds (Line 13-16). Otherwise, its child entries or associated objects will be pushed into H for further processing (Line 17-21). The algorithm terminates when H becomes empty. The objects in $\mathcal{NCC}$ are NN candidates of $\mathcal{O}$ for the given query Q.

Correctness. According to Theorem 4, F-$\mathcal{SD}(U_{mbr},V_{mbr},Q_{mbr})$ implies that we have F-$\mathcal{SD}(U,V,Q)$, P-$\mathcal{SD}(U,V,Q)$, SS-$\mathcal{SD}(U,V,Q)$, and S-$\mathcal{SD}(U,V,Q)$. Since V_{mbr} is contained by e_{mbr} for every object V in the entry e, it is safe to exclude an entry e in Algorithm 1 if F-$\mathcal{SD}(U_{mbr},e_{mbr},Q_{mbr})$ holds at Line 15. Moreover, V is not an NNC object if $\mathcal{SD}(U,V,Q)$ holds at Line 8. Consequently, all objects excluded in Algorithm 1 are not NNC objects. Now we show every non-candidate object will be discarded. Since objects are accessed based on their minimal distance to Q, an object V cannot be dominated by any object accessed after V according to the statistic-based pruning rules and cover-based pruning rules. Together with the fact that all four spatial dominance operators have the *transitivity* property, we only need to conduct dominance check against NNC objects seen so far to guarantee that all non-candidate objects will be excluded.

Progressive property of Algorithm 1. As stated in the above proof, we can safely claim that an object is an NN candidate object if it is not dominated by any objects seen as far. This implies that Algorithm 1 is a progressive

Algorithm 1: NNC($\mathcal{O}$,Q,$\mathcal{SD}$)

```
Input  : $\mathcal{O}$ : a set of objects $\mathcal{O}$
         $Q$ : query, $\mathcal{SD}$ : SD operator
Output: $\mathcal{NNC}(\mathcal{O}, Q, \mathcal{SD})$
1  $\mathcal{NNC} := \emptyset$ ;
2  push root of $R_{\mathcal{O}}$ into a heap $H$;
3  while $H \neq \emptyset$ do
4      $e := H.deheap()$;
5      if $e$ is an object $V$ then
6          $F := false$;
7          for each object $U \in \mathcal{NNC}$ do
8              if $\mathcal{SD}(U, V, Q)$ then
9                  $F := true$; break;
10         if $F = false$ then
11             $\mathcal{NNC} := \mathcal{NNC} \cup V$;
12     else
13         $F := false$;
14         for each object $U \in \mathcal{NNC}$ do
15             if F-SD($U_{mbr}$, $e_{mbr}$, $Q_{mbr}$) then
16                 $F := true$; break;
17         if $F = false$ then
18             if $e$ is a data entry associated with object $U$ then
19                 push $U$ into $H$;
20             else
21                 Push all child entries of $e$ into $H$;
22 return $\mathcal{NNC}$
```

algorithm in the sense that an object can be output before exploring all objects.

6. PERFORMANCE EVALUATION

In this section, we present the results of a comprehensive performance study to evaluate the effectiveness and efficiency of the proposed techniques in this paper.

Algorithms. We implement **SSD**, **SSSD**, **PSD**, **FSD** and **F$^+$SD** which are NN candidate search methods based on Algorithm 1 by replacing $\mathcal{SD}$ at Line 8 with S-$\mathcal{SD}$, SS-$\mathcal{SD}$, P-$\mathcal{SD}$, F-$\mathcal{SD}$ and F$^+$-$\mathcal{SD}$ respectively. Here, F$^+$-$\mathcal{SD}$ (U,V,Q) is F-$\mathcal{SD}$(U_{mbr},V_{mbr},Q_{mbr}) where MBRs of objects are organized by an R-Tree, and the implementation is from [16]. In addition, pruning and validation rules, level by level searching mechanism and geometrical properties introduced in Section 5.1 are also included for the corresponding dominance operators.

Note that the dominance check of F-$\mathcal{SD}$(U,V,Q) on instance level is not investigated in [16, 25]. Thus, we propose an efficient dominance check algorithm for performance evaluation. Similar to P-$\mathcal{SD}$, we observe that only the query points on the convex hull need to be considered for F-$\mathcal{SD}$. In particular, for each $q \in \mathcal{CH}(Q)$, we issue an NN search and a furthest neighbor search against the instances of V and U, respectively. We may immediately claim $\neg$F-$\mathcal{SD}$(U,V,Q) if there is a query instance q with $\delta_{max}(q, U) > \delta_{min}(q, V)$, otherwise F-$\mathcal{SD}$($U$,$V$,$Q$) holds. The dominance check is rather efficient because instances of each object are organized by a local R-Tree in the experiment.

Remark 1. *As shown in Section 1, NN-core approach [36] may miss important NN objects, and cannot even cover popular NN functions in $\mathcal{N}_1$. Therefore, we do not evaluate the NN-core approach in our experiment.*

Datasets. Two real datasets, **NBA**[1] and **GW**[2], are deployed in the experiments. NBA dataset is extracted from NBA players' game statistics, of 1,313 players with 298,051 records, each of which consists of the number of points, assistances and rebounds for a player in a game. Thus, each player is an object and each record is treated as a 3-dimensional instance (point). GW is obtained from GoWalla check-in dataset, consisting of 107,092 users with 6,442,890 check-ins. In GW, each user is regarded as an object and its check-ins are 2-dimensional instances (points). Instances in the same object have the same occurrence probability in this experiment.

Also, three real datasets, **HOUSE**[3], **CA**[4] and **USA**[5] are employed to represent the centers of objects as semi-real data. HOUSE is a 3-dimensional dataset with 127,932 records, each of which represents the percentage of an American family's annual income on 3 types of expenditures. CA is a 2-dimensional spatial dataset of 62k locations in California. USA dataset is obtained from the U.S. Geological Survey (USGS) with 1 million locations. USA dataset is used to evaluate the scalability of the proposed methods. By using methodologies in [8], we also generate synthetic centers for objects following the *anti-correlated* (A) or *independent* (E) distribution, where anti-correlated is the default distribution for synthetic data. The dimensionality (d) varies from 2 to 5, with default value 3. The number of objects (n) we used in synthetic dataset is kept as 100k. To generate instances for object centers, the expected edge length (h_d) of minimal bounding box (MBB) for objects varies from 100 to 500 with default value 400. Given a h_d, the lengths of objects are randomly chosen between 0 and $2 \times h_d$. Each object has m_d instances on average which follows $Normal$ (N) distribution, with standard deviation $\frac{h_d}{2}$. m_d varies from 20 to 100 with default value 40. All dimensions are normalized to domain [0, 10000]. Note that, the total number of instances in the default synthetic dataset is 100k×40 =**4 million**, and for the largest dataset USA, the instance number reaches **40 million**. The experiment parameters are list in Table 2, where the default value is in bold font.

Evaluation parameter	Values
dimensionality d	2, **3**, 4, 5
# of objects n	**100k**, 200k, 400k, 600k, 1M
# of object instances m_d	20, **40**, 60, 80, 100
edge length of object h_d	100, 200, 300, **400**, 500
object center distribution	**anti (A)**, indep (E)
# of query instances m_q	10, 20, **30**, 40, 50
edge length of query h_q	100, **200**, 300, 400, 500

Table 2: The summary of experiment parameter.

Workload. The query workload for NN candidates search consists of 100 randomly selected objects or centers from the underlying dataset. For the semi-real and synthetic data, the instance distribution of each query is the same as the objects. Moreover, the edge length of query (h_q) varies from

[1]http://www.nba.com
[2]http://snap.stanford.edu/data/loc-gowalla.html
[3]http://www.ipums.org
[4]http://www.census.gov/geo/www/tiger
[5]http://www.usgs.gov/

100 to 500 with 200 as default. The number of instances (m_q) varies from 10 to 50 with default value 30.

All algorithms are implemented in C++ with GNU GCC 4.8.2. Experiments are conducted on a PC with Intel Xeon 3.4GHz CPU and 32G memory using Ubuntu Linux. Similar to existing works (e.g., [26, 21, 37]), we employ $n+1$ R-trees to organize the n objects. Particularly, we use a global R-Tree to organize MBRs of the uncertain objects, and the page size is 4096 bytes. For each individual object, its instances are kept in a local R-Tree with fan-out 4, and we load the whole local R-tree into the main memory if it could not be pruned based on its MBR. The code for convex hull related computation is obtained from `www.qhull.org`.

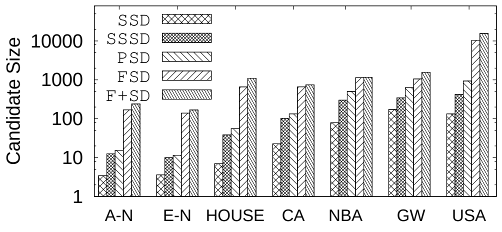


Figure 10: Candidate Size of Different Datasets

6.1 Effectiveness Evaluation

In this subsection, we evaluate the effectiveness of the algorithms through the average number of NN candidates.

In the first set of experiments, Figure 10 reports effectiveness of SSD, SSSD, PSD, FSD and F$^+$SD against all the datasets A-N, E-N, HOUSE, CA, NBA, GW and USA under the default setting. Particularly, A-N denotes the 3 dimensional synthetic data whose centers and instances follow the *anti-correlated* and *Normal* distribution, respectively. Similar definition goes to E-N. As expected, the candidate size of SSD, SSSD, PSD, FSD and F$^+$SD increase level by level in all datasets due to the NN candidates inclusion relationship illustrated in Figure 5. Moreover, we observe that the candidate size of PSD is much smaller than that of FSD and F$^+$SD, given that the three operators have the same coverage of NN functions. SSD and SSSD can further significantly reduce the NN candidates size with smaller coverage of NN functions. For instance, in HOUSE dataset, SSD, SSSD and PSD have 7, 38, and 55.8 candidates respectively, while FSD and F$^+$SD have reached 657 and 1100, respectively. The gap becomes more significant in A-N, E-N and USA. Particularly, in USA dataset, the candidate size of FSD and F$^+$SD have reached 10415 and 15580.4 respectively, while the candidate size of SSD, SSSD and PSD are quite stable, with 133.2, 422.5 and 933 respectively. In NBA and GW datasets, the instances of objects are highly overlapped, which renders an increase in the candidate size for all algorithms. Nevertheless, the performance of SSD, SSSD and PSD still outperforms that of FSD and F$^+$SD by a large margin.

We also evaluate the impact of different parameters against the NN candidate size on synthetic dataset and USA dataset. Figure 11(a) and Figure 11(b) depict the candidate size as a function of the number of object instances (m_d) and the edge length (h_d) on A-N dataset, respectively. It is shown that SSD, SSSD and PSD are well scalable to the growth of m_d and h_d. However, the performance of FSD and F$^+$SD is rather sensitive to the growth of h_d. This is because the full spatial dominance heavily relies on the boundaries (i.e., $\delta_{min}(q,U)$ and $\delta_{max}(q,U)$) of the objects, while S-$\mathcal{SD}$,

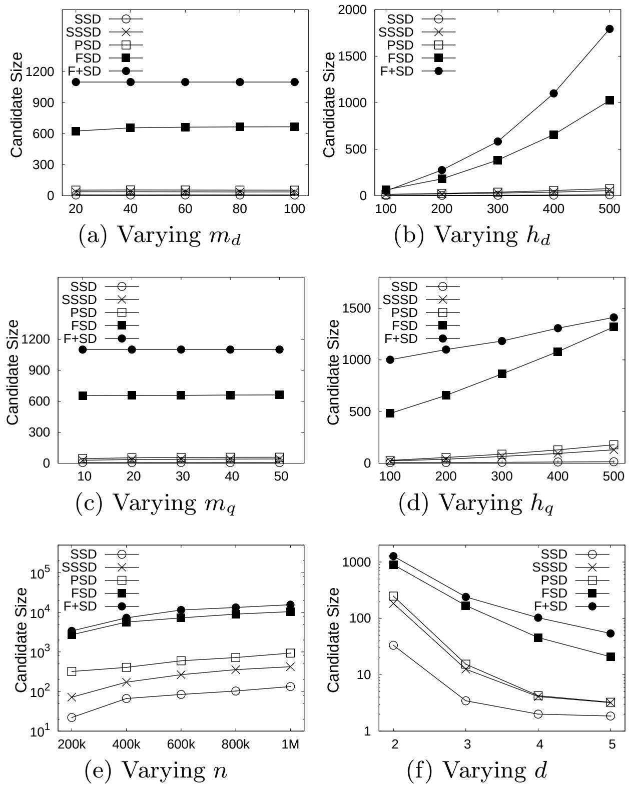


Figure 11: Impact of Diff. Parameters on Effectiveness

SS-$\mathcal{SD}$ and P-$\mathcal{SD}$ are much less sensitive, since the distance distributions are considered. Similar observation is reported in Figure 11(c) and Figure 11(d) where the number of instances and the edge length grow regarding the queries on A-N dataset.

To evaluate the scalability of proposed operator on the number of object (n), we conduct experiment on USA dataset by using 200k, 400k, 600k, 800k and 1M data. It is reported that the performance of FSD and F$^+$SD deteriorates with the growth of n, while SSD, SSSD and PSD are less sensitive. In Figure 13(f), we evaluate the impact of dimension (d) on A-N dataset. It shows that the number of candidates drops dramatically when the dimensionality varies from 2 to 5. This is because when the dimensionality d grows, the volume of minimal bound box of the objects becomes relatively smaller, which results in less overlapping among objects.

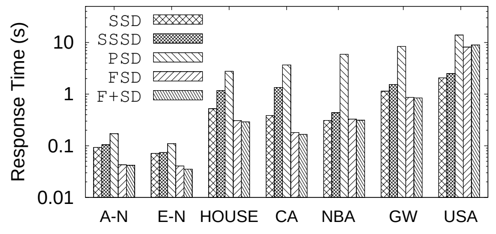
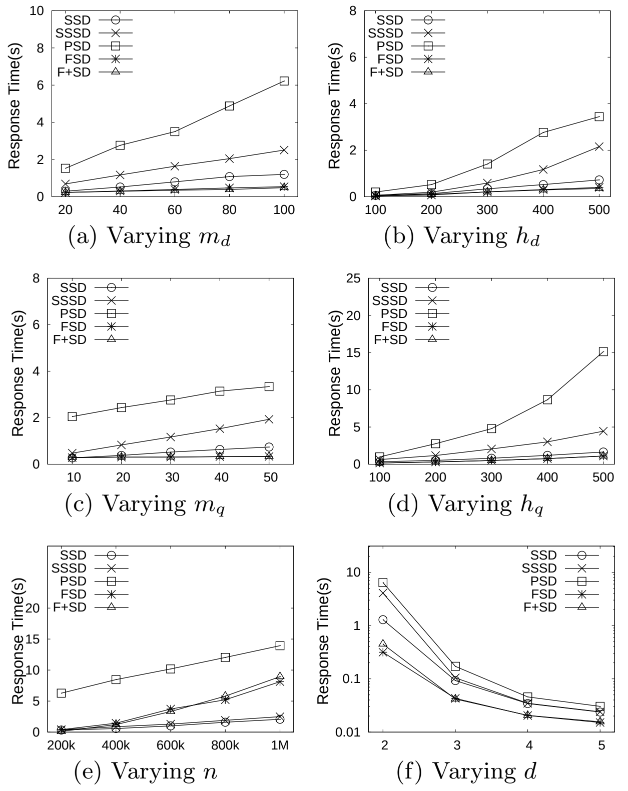


Figure 12: Response Time of Different Datasets

6.2 Evaluation of Efficiency

In this subsection, we demonstrate the efficiency of the algorithms by reporting the average query response time.

Figure 12 reports the performance of 5 algorithms against A-N, E-N, HOUSE, CA, NBA, GW and USA datasets un-



(a) Varying m_d (b) Varying h_d

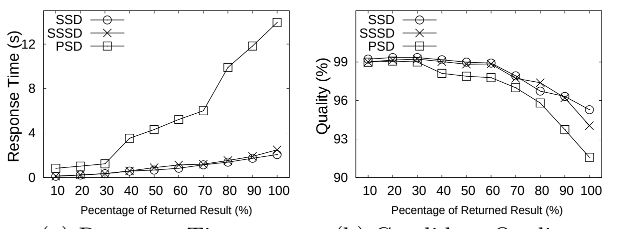
(c) Varying m_q (d) Varying h_q

(e) Varying n (f) Varying d

Figure 13: Impact of Diff. Parameters on Efficiency

der the default setting. Due to the simplicity of dominance check, FSD and F$^+$SD demonstrate superior performance in all datasets except USA. This is because, in USA dataset, the candidate size of both operators is too large (up to 15000), which renders plenty of verification cost in the search of candidates. Moreover, it is not surprising that PSD has the worst performance among 5 algorithms due to the expensive dominance checking costs. Nevertheless, considering that PSD keeps the same coverage of NN functions compared to FSD and F$^+$SD, but provides much smaller candidate set, the gain of PSD is significant. This is because a smaller candidate size is more crucial in our problem settings, e.g., the user may need to manually check the candidates and choose the object of her choice. As expected, SSD and SSSD outperform PSD because of the smaller candidate size and the cheaper dominance checking cost. Moreover, it is noticed that the performance of SSD and SSSD is even comparable to that of FSD and F$^+$SD on the hard datasets (e.g., NBA and GW) and outperforms FSD and F$^+$SD on USA dataset, due to the large number of candidates obtained by FSD and F$^+$SD. This suggests that when users are only interested in the ranking functions in $\mathcal{N}_{1,2}$ [1, 23], which actually already covers the majority of the NN functions, SSSD can provide NN candidates with much smaller size than that of PSD with a faster query response time. The performance can be further improved in terms of both candidate size and query delay if users only consider stable aggregation functions ($\mathcal{N}_1$) [2, 37], which have been widely used in many applications.

Figure 13 demonstrates the impact of various parameters on 5 algorithms on synthetic dataset and USA dataset. It is observed that FSD and F$^+$SD outperform other competitors when the number of object instances (m_d) , the edge length of object (h_d), the number of object instances (m_q) and the edge length of object (h_q) increases. However, in Figure 13(e), SSD and SSSD outperform FSD and F$^+$SD towards the number of object (n), when the dataset size increases from 200k to 1M. As explained before, this is because the significant growth in candidate size of FSD and F$^+$SD leading to lots of verification cost. Nevertheless, SSD and SSSD also perform reasonably well and are less sensitive to the change in m_d, h_d, m_q, h_q and n. As reported in Figure 11(f), the number of candidates is significantly reduced towards the dimensionality. Consequently, the same trend is observed for the time efficiency in Figure 13(f).



(a) Response Time (b) Candidate Quality

Figure 14: Performance of Progressive Property

Next we demonstrate the progressive property of the algorithm on USA dataset. As shown in Figure 14, x-axis is the candidate return progress, i.e., the percentage of candidate returned, and the corresponding response time and candidate quality (avg. number of objects dominated by returned candidates) are reported. As shown in Figure 14(a), for PSD, 20 percentage candidates are returned in less than 1s and 70 percentage candidates are return in half time, since as the number of returned candidates grows the verification cost also increases. In Figure 14(b), we can observe that higher quality candidates are often tendered to be returned first. Thus, the progressive property of the algorithm is well suited for users to explore the candidates in a progressive way without waiting until all candidates are returned, like web search engine. Due to the space limitation, we defer the experimental results of the effectiveness of different filtering techniques and experiment summary to the Appendix C.

7. CONCLUSION

To address new challenges arising in the NN search on objects with multiple instances, we studied the problem of NN candidates search which can effectively provide users a small set of candidates for a given query object. By modeling various existing NN ranking functions as three representative families, we devised three spatial dominance operators which are *optimal* w.r.t. different coverage of NN functions with theoretical underpinnings. Effective and efficient filtering and verification techniques have been proposed for the dominance check and NN candidates computations. Our comprehensive experiments justified the effectiveness of the new spatial dominance operators, which confirmed that users can make a good trade-off among the NN candidates size and the coverage of NN functions. Moreover, our extensive performance evaluation demonstrated the efficiency of our techniques proposed in this paper.

Acknowledgments. Ying Zhang is supported by ARC DE140100679 and DP130103245. Wenjie Zhang is supported by ARC DP150103071 and DP150102728. Xuemin Lin is supported by ARC DP150102728, DP140103578, and DP120104168. Muhammad Aamir Cheema is supported by ARC DE130101002 and DP130103405.

8. REFERENCES

[1] P. K. Agarwal, B. Aronov, S. Har-Peled, J. M. Phillips, K. Yi, and W. Zhang. Nearest neighbor searching under uncertainty II. In *PODS*, pages 115–126, 2013.

[2] P. K. Agarwal, A. Efrat, S. Sankararaman, and W. Zhang. Nearest-neighbor searching under uncertainty. In *PODS*, pages 225–236, 2012.

[3] L. Antova, T. Jansen, C. Koch, and D. Olteanu. Fast and simple relational processing of uncertain data. In *ICDE*, pages 983–992, 2008.

[4] D. Barbará, H. Garcia-Molina, and D. Porter. The management of probabilistic data. *TKDE*, 4(5):487–502, 1992.

[5] I. Bartolini, P. Ciaccia, and M. Patella. The skyline of a probabilistic relation. *TKDE*, 2012.

[6] M. Ben-Or. Lower bounds for algebraic computation trees (preliminary report). In *STOC*, pages 80–86, 1983.

[7] G. Beskales, M. A. Soliman, and I. F. Ilyas. Efficient search for the top-k probable nearest neighbors in uncertain databases. *PVLDB*, 1(1), 2008.

[8] S. Börzsönyi, D. Kossmann, and K. Stocker. The skyline operator. In *ICDE 2001*.

[9] R. Cheng, J. Chen, M. F. Mokbel, and C.-Y. Chow. Probabilistic verifiers: Evaluating constrained nearest-neighbor queries over uncertain data. In *ICDE*, 2008.

[10] S. D. Cohen and L. J. Guibas. The earth mover's distance under transformation sets. In *ICCV*, pages 1076–1083, 1999.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (second edition)*. 2001.

[12] G. Cormode, F. Li, and K. Yi. Semantics of ranking queries for probabilistic data and expected ranks. In *ICDE*, 2009.

[13] N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In *VLDB*, pages 864–875, 2004.

[14] T. Eiter and H. Mannila. Distance measures for point sets and their computation. *Acta Inf.*, 34(2):109–133, 1997.

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

[16] T. Emrich, H. Kriegel, P. Kröger, M. Renz, and A. Züfle. Boosting spatial pruning: on optimal pruning of mbrs. In *SIGMOD*, pages 39–50, 2010.

[17] B. Fitzenberger, R. Koenker, and J. A. Machado. *Economic Applications of Quantile Regression (Studies in Empirical Economics)*. 2002.

[18] M. Hua, J. Pei, W. Zhang, and X. Lin. Ranking queries on uncertain data: A probabilistic threshold approach. In *SIGMOD*, 2008.

[19] H. Kriegel, S. Brecheisen, P. Kröger, M. Pfeifle, and M. Schubert. Using sets of feature vectors for similarity search on voxelized CAD objects. In *SIGMOD*, pages 587–598, 2003.

[20] H. Kriegel, A. Pryakhin, M. Schubert, and A. Zimek. COSMIC: conceptually specified multi-instance clusters. In *ICDM*.

[21] H.-P. Kriegel, P. Kunath, and M. Renz. Probabilistic nearest-neighbor query on uncertain objects. In *DASFAA*, pages 337–348, 2007.

[22] L. V. S. Lakshmanan, N. Leone, R. B. Ross, and V. S. Subrahmanian. Probview: A flexible probabilistic database system. *ACM Trans. Database Syst.*, 22(3):419–469, 1997.

[23] J. Li, B. Saha, and A. Deshpande. A unified approach to ranking in probabilistic databases. *VLDB J.*, 20(2), 2011.

[24] X. Lin, Y. Zhang, W. Zhang, and M. A. Cheema. Stochastic skyline operator. In *ICDE*, pages 721–732, 2011.

[25] C. Long, R. C. Wong, B. Zhang, and M. Xie. Hypersphere dominance: an optimal approach. In *SIGMOD*, pages 111–122, 2014.

[26] J. Pei, B. Jiang, X. Lin, and Y. Yuan. Probabilistic skylines on uncertain data. In *VLDB*, 2007.

[27] J. Ramon and M. Bruynooghe. A polynomial time computable metric between point sets. *Acta Inf.*, 37(10):765–780, 2001.

[28] B. E. Ruttenberg and A. K. Singh. Indexing the earth mover's distance using normal distributions. *PVLDB*, 5(3):205–216, 2011.

[29] M. Shaked and J. G. Shanthikumar. *Stochastic Orders and Their Applications*. Academic Press, 2007.

[30] M. Sharifzadeh and C. Shahabi. The spatial skyline queries. In *VLDB*, pages 751–762, 2006.

[31] S. Singh, C. Mayfield, S. Mittal, S. Prabhakar, S. E. Hambrusch, and R. Shah. Orion 2.0: native support for uncertain data. In *SIGMOD Conference*, 2008.

[32] M. A. Soliman, I. F. Ilyas, and K. C. Chang. Top-k query processing in uncertain databases. In *ICDE 2007*.

[33] W. Son, M. Lee, H. Ahn, and S. Hwang. Spatial skyline queries: An efficient geometric algorithm. In *SSTD*, pages 247–264, 2009.

[34] H. Wang, F. Nie, and H. Huang. Learning instance specific distance for multi-instance classification. In *AAAI*, 2011.

[35] J. Xu, Z. Zhang, A. K. H. Tung, and G. Yu. Efficient and effective similarity search over probabilistic data based on earth mover's distance. *VLDB J.*, 21(4):535–559, 2012.

[36] S. M. Yuen, Y. Tao, X. Xiao, J. Pei, and D. Zhang. Superseding nearest neighbor search on uncertain spatial databases. *IEEE Trans. Knowl. Data Eng.*, 22(7):1041–1055, 2010.

[37] W. Zhang, X. Lin, M. A. Cheema, Y. Zhang, and W. Wang. Quantile-based knn over multi-valued objects. In *ICDE*, 2010.

[38] W. Zhang, X. Lin, Y. Zhang, M. A. Cheema, and Q. Zhang. Stochastic skylines. *TODS*, 37(2), 2012.

[39] X. Zhang and J. Chomicki. On the semantics and evaluation of top-k queries in probabilistic databases. In *DBRank*, 2008.

[40] Y. Zhang, X. Lin, G. Zhu, W. Zhang, and Q. Lin. Efficient rank based knn query processing over uncertain data. In *ICDE*, 2010.

[41] Y. Zhang, W. Zhang, X. Lin, M. A. Cheema, and C. Zhang. Matching dominance: capture the semantics of dominance for multi-dimensional uncertain objects. In *SSDBM*, 2014.

[42] Y. Zhang, W. Zhang, J. Pei, X. Lin, Q. Lin, and A. Li. Consensus-based ranking of multi-valued objects: A generalized borda count approach. *TKDE*, 26(1):83–96, 2014.

APPENDIX

A. SOME NN FUNCTIONS

In this section, we introduce three popular NN functions that belong to $\mathcal{N}_3$, which derive the NN score based on a subset of pair-wise distances.

Informally, two sets of points are close in the Hausdorff distance [14, 27] if every point of either set is close to some points of the other set. Given two objects U and Q, following is a formal definition of their Hausdorff distance.

DEFINITION 11. ***Hausdorff distance.***

$$\begin{aligned} D_h(U,Q) = max\{&\max_{u \in U}\ \delta_{min}(u,Q), \\ &\max_{q \in Q}\ \delta_{min}(q,U)\} \end{aligned} \tag{5}$$

Under our problem settings where each object has multiple instances with total probability mass 1, the definitions of **Earth Mover's distance** [10] and **Netflow distance** [27] are equivalent to each other. Below we formally introduce the Netflow distance [27].

Distance Network. Let N (E) denote a set of vertices (edges) and W (C) records the weights (capacities) of the edges. G=<N,E,C,W> is a network with s, $t \in N$ being *source* and *sink*, respectively. A feasible flow f of the network G maps each edge $e \in E$ a non-negative value $f(e)$ following the capacity constraint and conservation of flow constraint [11]. The value $|f|$ of the flow is defined as $|f| = \sum_{\langle s,x\rangle \in E} f(\langle s,x\rangle)$, and the cost $c(f)$ of the flow is defined as $c(f) = \sum_{e \in E} w(e)f(e)$.

In this paper, we assume that the total probability (weight) of the instances for each object is 1. Then the distance network between the object U and the query Q, denoted by $G_{U,Q}$, can be constructed as follows.

- Vertices s and t for source and sink respectively.
- Each instance $q \in Q$ contributes a vertex q and an edge $<s, q>$ with $c_{s,q} = p(q)$ and $w_{s,q} = 0$.
- Each instance $u \in U$ contributes a vertex u and an edge $<u, t>$ with $c_{u,t} = p(u)$ and $w_{u,t} = 0$.
- For any two instances $q \in Q$ and $u \in U$, there is an edge $<u, q>$ with $c_{u,q} = \infty$ and $w(\langle u,q\rangle) = \delta(u, q)$.

DEFINITION 12. ***Netflow distance.*** *Given the distance network $G_{U,Q}$ of the object U and query Q, the netflow distance between U and Q, denoted by $M_{nd}(U, Q)$, is the minimal cost of the maximal flow $c(f)$ of $G_{U,Q}$ with $|f| = 1$.*

B. PROOFS

B.1 Proof of Theorem 1

THEOREM 1. *Match order is equivalent to the usual stochastic order.*

PROOF. "$\Rightarrow$": Given $X \preceq_M Y$, there is a match $M_{X,Y}$ where $t.x \leq t.y$ for every $t \in M_{X,Y}$. Then we have $X \preceq_{st} Y$ because for every $t \in M_{X,Y}$, $t.y \leq \lambda$ implies that $t.x \leq \lambda$ for every $\lambda \in R$.
"$\Leftarrow$": Given $X \preceq_{st} Y$, we show how to iteratively find a feasible match $M_{X,Y}$ such that $t.x \leq t.y$ for every $t \in M_{X,Y}$. We visit the instances of Y (X) in non-decreasing order. Let x_i and y denote the smallest unvisited instance of X and Y, respectively. Since $Pr(X \leq y) \geq Pr(Y \leq y)$ and the same amount of probability mass has been consumed in previous instance matches, there is a set of instance $\{x_i, \ldots, x_k\}$ where x_k is the smallest instance such that $\sum_{i \leq j \leq k} p(x_j) \geq p(y)$. If $\sum_{i \leq j \leq k} p(x_j) > p(y)$, we may split the instance x_k such that $\sum_{i \leq j \leq k} p(x_j) = p(y)$. For every $x_j \in \{x_i, \ldots, x_k\}$, we generate a tuple $t\langle x_j, y, p(x_j)\rangle$. Figure 3(b) illustrates a simple example of the match between A_Q and B_Q. In this way, we generate a feasible match $M_{X,Y}$ and $X \preceq_M Y$ follows. □

B.2 Proof of Theorem 2

THEOREM 2. *F-$\mathcal{SD}$ $\subset$ P-$\mathcal{SD}$ $\subset$ SS-$\mathcal{SD}$ $\subset$ S-$\mathcal{SD}$.*

PROOF. SS-$\mathcal{SD}$ $\subset$ S-$\mathcal{SD}$: Given two objects U and V, and the query Q, we show that SS-$\mathcal{SD}(U,V,Q)$ implies S-$\mathcal{SD}(U,V,Q)$ but not vice versa. Given SS-$\mathcal{SD}(U,V,Q)$, we have $\Pr(U_q \leq \lambda) \geq \Pr(V_q \leq \lambda)$ for every $\lambda \in R$ regarding every $q \in Q$. Since we have $\Pr(U_Q \leq \lambda) = \sum_{q \in Q} Pr(U_q \leq \lambda) \times p(q)$ and $\Pr(V_Q \leq \lambda) = \sum_{q \in Q} Pr(V_q \leq \lambda) \times p(q)$, it is immediate that we have $\Pr(U_Q \leq \lambda) \geq \Pr(V_Q \leq \lambda)$, and hence $U_Q \preceq_{st} V_Q$. Thus, S-$\mathcal{SD}(U,V,Q)$ holds. On the other hand, Figure 3 illustrates an example where we have S-$\mathcal{SD}(A,C,Q)$ and $\neg$SS-$\mathcal{SD}(A,C,Q)$. Therefore, SS-$\mathcal{SD}$ $\subset$ S-$\mathcal{SD}$ follows.

P-$\mathcal{SD}$ $\subset$ SS-$\mathcal{SD}$: P-$\mathcal{SD}(U,V,Q)$ implies that there is a match $M_{U,V}$ such that $t.u \preceq_Q t.v$ for each $t \in M_{U,V}$. Therefore, for each $q \in Q$, we have $\delta(t.u, q) \leq \delta(t.v, q)$, and hence $U_q \preceq_M V_q$. Then we have $U_q \preceq_{st} V_q$ according to Theorem 1. Thus, SS-$\mathcal{SD}(U,V,Q)$ holds. Figure 4 gives an example where SS-$\mathcal{SD}(A,B,Q)$ but $\neg$P-$\mathcal{SD}(A,B,Q)$.

F-$\mathcal{SD}$ $\subset$ P-$\mathcal{SD}$: F-$\mathcal{SD}(U,V,Q)$ implies that for every $q \in Q$, $\delta_{max}(q, U) \leq \delta_{min}(q, V)$. Thus, for every match $M_{U,V}$, we have $\delta(t.u, q) \leq \delta(t.v, q)$ for every $t \in M_{U,V}$ and $q \in Q$. Thus, P-$\mathcal{SD}(U,V,Q)$ holds. In the example of Figure 4, we have P-$\mathcal{SD}(A,C,Q)$ but $\neg$F-$\mathcal{SD}(A,C,Q)$. □

B.3 Proof of Theorem 3

THEOREM 3. *When $|Q| = 1$, we have F-$\mathcal{SD}$ $\subset$ P-$\mathcal{SD}$ = SS-$\mathcal{SD}$ = S-$\mathcal{SD}$.*

PROOF. Since $|Q| = 1$, it is immediate that SS-$\mathcal{SD}$ is equivalent to S-$\mathcal{SD}$. Moreover, according to Theorem 1, we have P-$\mathcal{SD}$ = SS-$\mathcal{SD}$ = S-$\mathcal{SD}$. Figure 15 illustrates an example where we have P-$\mathcal{SD}(A,B,Q)$, SS-$\mathcal{SD}(A,B,Q)$, S-$\mathcal{SD}(A,B,Q)$ but $\neg$F-$\mathcal{SD}(A,B,Q)$. Together with Theorem 2, we have F-$\mathcal{SD}$ $\subset$ P-$\mathcal{SD}$. □

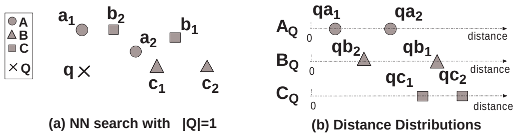


Figure 15: Example of Theorem 3

B.4 Proof of Theorem 5

THEOREM 5. *S-$\mathcal{SD}$ is optimal w.r.t. $\mathcal{N}_1$, and S-$\mathcal{SD}$ does not cover $\mathcal{N}_{2,3}$.*

PROOF. *Correctness.* Given two independent objects U and V, and query Q, U_Q and V_Q are two independent random variables and S-$\mathcal{SD}(U,V,Q)$ implies that $U_Q \preceq_{st} V_Q$. Since a *stable* aggregate function will be used against all pair-wise distances for a function $f \in \mathcal{N}_1$, it is immediate that we have $f(U) \leq f(V)$ according to Definition 8.

Completeness. $\neg$S-$\mathcal{SD}(U,V,Q)$ implies that there is a $\lambda \in R$ such that $\Pr(U_Q \leq \lambda) < \Pr(V_Q \leq \lambda)$. Let $\phi = \Pr(V_Q \leq \lambda)$, we have $quan_\phi(U_Q) > quan_\phi(V_Q)$ according to Definition 10 where *quantile* is a function in $\mathcal{N}_1$.

Not cover $\mathcal{N}_{2,3}$. In the example of Figure 3, we have S-$\mathcal{SD}(A,C,Q)$ but C is NN object if NN probability is used, which is a function in $\mathcal{N}_2$. Moreover, the example in Figure 4 shows that S-$\mathcal{SD}(A,C,Q)$ but EMD(A,Q)>EMD(C,Q) where EMD is a function in $\mathcal{N}_3$. Consequently, S-$\mathcal{SD}$ does not cover $\mathcal{N}_{2,3}$. □

B.5 Proof of Theorem 6

THEOREM 6. *SS-$\mathcal{SD}$ is optimal w.r.t. $\mathcal{N}_{1,2}$, and SS-$\mathcal{SD}$ does not cover $\mathcal{N}_3$.*

PROOF. *Correctness.* The correctness is immediate regarding $\mathcal{N}_1$ since SS-$\mathcal{SD}$ $\subset$ S-$\mathcal{SD}$ and S-$\mathcal{SD}$ is optimal w.r.t. $\mathcal{N}_1$. Now we only consider $\mathcal{N}_2$. Given objects U and V, and the query Q, SS-$\mathcal{SD}(U,V,Q)$ indicates that for every $q \in Q$ we have $U_q \preceq_{st} V_q$. So for every $\lambda \in R$, we have $\Pr(U_q \leq \lambda) \geq \Pr(V_q \leq \lambda)$ regarding a given query instance q since U_q and V_q are independent. Recall that $\delta(U,W)$ and $s(U,W)$ denote the distance and score of the object U in the possible world W, respectively, and we have $s(U,W) \le s(V,W)$ if $\delta(U,W) < \delta(V,W)$ for every function in $\mathcal{N}_2$. Let $\mathcal{W}_q$ denote the possible worlds in which q occurs, we have $\Pr(U_q \le \lambda) \ge \Pr(V_q \le \lambda)$ for every $\lambda \in R$. This implies that $\sum_{W \in \mathcal{W}_q \wedge s(U,W) \le \xi} Pr(W) \ge \sum_{W \in \mathcal{W}_q \wedge s(V,W) \le \xi} Pr(W)$ for every possible score ξ. Consequently, we have $U_{\mathcal{W}} \preceq_{st} V_{\mathcal{W}}$ as $U_q \preceq_{st} V_q$ held for every $q \in Q$. Since the stable aggregate function is used to derive the NN score, we have $f(U) \le f(V)$ for every $f \in \mathcal{N}_{1,2}$.

Completeness. Given $\neg$SS-$\mathcal{SD}$(U,V,Q), there is at least one query instance $q_1 \in Q$ such that $U_{q_1} \not\preceq_{st} V_{q_1}$; that is, there exists a $\lambda_1 \in R$ where $\Pr(U_{q_1} \le \lambda_1) < \Pr(V_{q_1} \le \lambda_1)$, i.e., $\Pr(U_{q_1} > \lambda_1) > \Pr(V_{q_1} > \lambda_1)$. Then we can construct a function f as follows. Let the aggregate function g be the weighted sum of scores where the weight of a score $s(U,W)$ is set to $Pr(W)$ if $W \in \mathcal{W}_{q_1}$, and it is set to 0 otherwise. Clearly, g is a stable function. Moreover, we set $s(U,W)$ to 1 if $\delta(U,W) > \lambda_1$, and 0 otherwise. In this way, we have $f(U) = \Pr(U_q > \lambda_1) \times p(q_1)$ and $f(V) = \Pr(V_q > \lambda_1) \times p(q_1)$. Thus, we have $f(U) > f(V)$.

Not cover $\mathcal{N}_3$. In the example of Figure 4, we have SS-$\mathcal{SD}$(A,C,Q) and EMD(A,Q) > EMD(C,Q) where EMD is a function in $\mathcal{N}_3$. □

B.6 Proof of Theorem 7

THEOREM 7. *P-$\mathcal{SD}$is optimal w.r.t. $\mathcal{N}_{1,2,3}$.*

PROOF. *Correctness.* The correctness is immediate regarding $\mathcal{N}_{1,2}$ since P-$\mathcal{SD} \subset$ SS-$\mathcal{SD}$ and SS-$\mathcal{SD}$ is optimal w.r.t. $\mathcal{N}_{1,2}$. Now we only consider $\mathcal{N}_3$. Given objects U and V, and the query Q, P-$\mathcal{SD}$(U,V,Q) implies that there is a match $M_{U,V}$ such that $t.u \preceq_Q t.v$ for every $t \in M_{U,V}$. For every function $f \in \mathcal{N}_3$ with aggregate stable function g, $\sigma_V(V_Q)$ denotes the corresponding selected pair-wise distances from V_Q during the computation of $f(V)$. Based on the match $M_{U,V}$, we can construct the counterpart $\sigma_V(U_Q)$ for object U. For every pair $\langle \delta(v,q), p \rangle$ in $\sigma_V(V_Q)$, its counterpart in U_Q is $\langle \delta(u,q), p \rangle$ with $\delta(u,q) \le \delta(v,q)$ since v is assigned to u in $M_{U,V}$. Consequently, we have $\sigma_V(U_Q) \preceq_{st} \sigma_V(V_Q)$, and hence $g(\sigma_V(U_Q)) \le g(\sigma_V(V_Q))$. Since f is a counterpart computable function, we have $f(U) \le g(\sigma_V(U_Q))$, and then $f(U) \le f(V)$ holds.

Completeness. $\neg$P-$\mathcal{SD}$(U,V,Q) implies that for every possible match $M_{V,U}$, we can find a pair $\langle \delta(v,q), p \rangle$ from V_Q such that $\delta(v,q)$ is always smaller than its counterpart in U_Q. By simply selecting the pair $\langle \delta(v,q), p \rangle$ in V_Q and its couterpart pair in U_Q regarding a match $M_{U,V}$, we have $f(U) > f(V)$. □

B.7 Proof of Theorem 8

THEOREM 8. *F-$\mathcal{SD}$ is correct w.r.t. $\mathcal{N}_{1,2,3}$, but does not have completeness property w.r.t. $\mathcal{N}_{1,2,3}$.*

PROOF. Since we have F-$\mathcal{SD} \subset$ P-$\mathcal{SD}$, the correctness of F-$\mathcal{SD}$ is immediate. In Figure 4, we have $\neg$F-$\mathcal{SD}$ (A,C,Q) because $\delta(a_2, q_2) > \delta(c_2, q_2)$. Nevertheless, we have $f(A) \le f(C)$ for every $f \in \mathcal{N}_{1,2,3}$ because P-$\mathcal{SD}$ (A,C,Q) holds. Consequently, F-$\mathcal{SD}$ is not complete w.r.t. $\mathcal{N}_{1,2,3}$. □

B.8 Proof of Theorem 9

THEOREM 9. *S-$\mathcal{SD}$, SS-$\mathcal{SD}$, P-$\mathcal{SD}$ and F-$\mathcal{SD}$ have the transitivity property.*

PROOF. Given S-$\mathcal{SD}$(U,V,Q) and S-$\mathcal{SD}$(V,Z,Q), we have $\Pr(U_Q \le \lambda) \ge (V_Q \le \lambda)$ and $\Pr(V_Q \le \lambda) \ge (Z_Q \le \lambda)$ for every $\lambda \in R$. This immediately implies that $\Pr(U_Q \le \lambda) \ge \Pr(Z_Q \le \lambda)$. Thus, S-$\mathcal{SD}$($U$,$Z$,$Q$) holds. We show that SS-$\mathcal{SD}$ also satisfies this property with similar rationale.

Given F-$\mathcal{SD}$(U,V,Q) and F-$\mathcal{SD}$(V,Z,Q), for every $q \in Q$, we have $\delta_{max}(q,U) \le \delta_{min}(q,V)$ and $\delta_{max}(q,V) \le \delta_{min}(q,Z)$. Then we have $\delta_{max}(q,U) \le \delta_{min}(q,Z)$, and hence F-$\mathcal{SD}$($U$,$Z$,$Q$) holds.

Given P-$\mathcal{SD}$(U,V,Q) and P-$\mathcal{SD}$(V,Z,Q) we have two matches $M_{U,V}$ and $M_{V,Z}$ with $t.u \preceq_Q t.v$ and $m.v \preceq_Q m.z$ for every $t \in M_{U,V}$ and $m \in M_{V,Z}$. By splitting instances, we can come up with two new matches with the same number of tuples, denoted by $M^*_{U,V}$ and $M^*_{V,Z}$. For every tuple $t \in M^*_{U,V}$ and its counterpart tuple $m \in M^*_{V,Z}$, we generate a tuple $t^*\langle t.u, m.z, t.p \rangle$ for the new match $M_{U,Z}$. With similar rationale to F-$\mathcal{SD}$, $u \preceq_Q v$ and $v \preceq_Q z$ implies that $u \preceq_Q z$. Therefore, P-$\mathcal{SD}$ satisfies the transitivity property because P-$\mathcal{SD}$(U,Z,Q) holds. □

B.9 Proof of Theorem 10

THEOREM 10. *Given two random variables X and Y with n instances each, any algorithm to determine whether $X \preceq_{st} Y$ must have complexity of at least $\Omega(n\log(n))$ if the algorithm is based only on comparisons between elements from X and Y.*

PROOF. Given two random variables X and Y, it is immediate that $X = Y$ if $X \preceq_{st} Y$ and $Y \preceq_{st} X$ since for any value λ, $P(X \le \lambda) = P(Y \le \lambda)$. Since a set can be regarded as a random variable in which each element has the same probability, we can determine whether two sets are equal by conducting two stochastic dominance checks. As shown in [6], for any computation tree solving the problem of set equality, its lower bound time complexity must be $\Omega(n \log n)$ where n is the size of the set. This implies that any comparison based algorithm to determine whether $X \preceq_{st} Y$ musth have complexity of $\Omega(n\log(n))$. □

B.10 Proof of Theorem 12

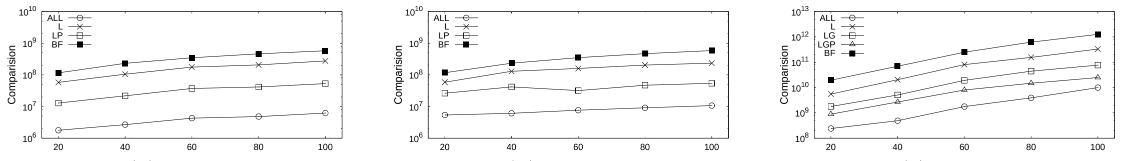
THEOREM 12. *Let $G_{U,V}$ be the network constructed based on two objects U, V and the query Q, we have P-$\mathcal{SD}$(U,V,Q) if and only if $|f^*| = 1$ where f^* is a max-flow of $G_{U,V}$.*

PROOF. P-$\mathcal{SD}$(U,V,Q) implies that there is a match $M_{U,V}$ between U and V where $t.u \preceq_Q t.v$ for every $t \in M_{U,V}$. Then we can construct a flow f of $G_{U,V}$ as follows. For every tuple $t \in M_{U,V}$, we set $f(\langle t.u, t.v \rangle) = t.p$. Moreover, for every $u \in U$ and $v \in V$, we set $f(s,u) = p(u)$ and $f(v,t) = p(v)$. Clearly this flow f is feasible (i.e., satisfying capacity and conservation constraints) with $|f| = 1$. which is a max-flow since $\sum_{u \in U} c(s,u) = 1$. With similar rationale, given $|f^*| = 1$ we can construct a match $M_{U,V}$ to verify that P-$\mathcal{SD}$(U, V, Q) holds. □

C. EXPERIMENTS AND ANALYSIS

C.1 Additional Experiments

In this experiments, Figure 16 demonstrates the effectiveness of different filtering techniques proposed in Section 5.1 on SSD, SSSD and PSD. The number of data instance (m_d) varies from 20 to 100 on HOUSE dataset and the number of average instance comparison is reported. We explore the effectiveness of filtering strategies by adding them to the



(a) SSD (b) SSSD (c) PSD

Figure 16: Effectiveness of Diff. Filtering Techniques

brute force method one by one. In particular, **All** is the algorithm with all filtering techniques; **BF** is method without any filtering techniques, i.e., the brute force method; **L** is the algorithm that adds level by level searching mechanism to BF; **LP** stands for the method adding pruning rules to L; **LG** stands for the method adding geometrical strategy on L; **LGP** stands for the method adding pruning rules on LG. It is reported that all four techniques make remarkable contribution in improving the efficiency of the NN candidates searching. Given the filtering strategies, SSD and PSD save up to 2 order of magnitude of cost and SSSD saves 60 times of cost comparing with the brute force method respectively.

C.2 Summary

In this section, we evaluate the NN candidates computation algorithms based on five spatial dominance operators. Following are some important observations.

- Experiments confirm that F+SD [16, 25] and FSD algorithms always lead to a much larger number of NN candidates compared with SSD, SSSD and PSD. Thus, they are not suitable to provide NN candidates for users.
- It is observed that three spatial dominance operators (S-$\mathcal{SD}$, SS-$\mathcal{SD}$ and P-$\mathcal{SD}$) proposed in this paper can significantly reduce the number of candidate objects, which is crucial in our problem since users need to make decision among the NN candidates. Moreover, experiments shows that users may make a good trade-off between the candidate size (time efficiency) and the coverage of NN functions.
- Experiments demonstrate that our algorithms can instantly output a large portion of high quality NN candidates. This can enhance the users' satisfaction since they can immediately start to browse the promising results once the query is issued.

D. RELATED WORK

In this section, we review the existing works on objects with multiple instances which are closely related to this paper.

D.1 NN Search

NN candidate search on objects with multiple instances has been studied in different contexts. As discussed in Section 3, given the distance measure of two points (instances), there are three representative families of NN ranking mechanisms including all pairs based NN search ($\mathcal{N}_1$), possible world based NN search ($\mathcal{N}_2$), and selected pairs based NN search ($\mathcal{N}_3$).

There is a long history of study for NN functions in $\mathcal{N}_1$ and $\mathcal{N}_3$. In many applications such as economics and social study, distance between two objects is derived based on an aggregation function against the pair-wise distances. Typical examples include *min*, *max*, *mean*, and *quantile* distances (e.g., [29, 37]). To describe the closeness of two set of points or distributions, various distances such as Hausdorff distance [27], Earth Mover's distance [10], and Netflow distance [27] have been proposed based on a subset of the pair-wise distances.

In recent years, the inherent uncertainty of data in many applications lead to the emergence of many uncertain database models (e.g., [13, 31]). A variety of ranking methods have been proposed in the literature for uncertain data processing such as U-top k[32], global top k [39, 18], *expected rank* [12], and *parameterized ranking* [23], which have been employed in nearest neighbor search on uncertain objects (e.g., [21, 7, 9, 40, 2, 1]). Besides the possible world semantics based approaches, Earth Mover's distance has also been applied to retrieve NN object on uncertain data (e.g., [35, 28]).

D.2 NN Candidate Search

The *spatial dominance* operator is first introduced by Sharifzadeh *e*t al. in [30]. Informally, we say a point x spatially dominates another point y w.r.t. a set of query points if x is closer to every query point than y. Efficient algorithms [30, 33] have been developed to compute spatial skyline. Nevertheless, the problem studied in this paper is inherently different, since the work in [30, 33] focuses on the case where each object has only one instance. The full spatial dominance (F-$\mathcal{SD}$) operator in [16, 25] can be regarded as its natural extension in the context of NN search on multi-instance objects. Various efficient algorithms (e.g., [16, 25, 15]) have been proposed to check the spatial dominance in the context of NN search, all NN search, and reverse NN search on both certain and uncertain data. We remark that these works focus on how to efficiently prune the non-promising objects at high level, based on the approximations (e.g., rectangles and hyperspheres) of the objects to speed up the computation, instead of providing the end-user with a set of NN candidates with consideration of popular NN ranking functions. In [36], Yuen *et al.* introduce the concept of NN-core to retrieve NN candidates based on the pair-wise competitions among the objects. However, as discussed in Section 1, NN objects regarding some popular NN ranking functions may be missed by NN-core.

D.3 Skyline Computation

The problem of skyline computation has been investigated in the context of uncertain data. Probabilistic skyline on uncertain data is first tackled in [26]. Efficient techniques are proposed following the bounding-pruning-refining framework. P-*domination* is proposed by Bartolini *et al.* [5] to capture the dominance among uncertain objects based on the tuple-level uncertain object model. The stochastic order based skyline computation for multi-dimensional uncertain objects is investigated in [24, 38]. Recently, Zhang *e*t al. [41] propose a layer based indexing technique for multi-dimensional uncertain objects based on the match based dominance operator. Our problem can be regarded as the skyline computation based on new spatial dominance operators. Nevertheless, the problem is inherently different from previous studies since a query object with multiple instances is involved.