Predicting Optimal Facility Location without Customer Locations

Emre Yilmaz
Computer Engineering Department
Bilkent University
Ankara, Turkey
emre.yilmaz@cs.bilkent.edu.tr

Sanem Elbasi
Computer Engineering Department
Bilkent University
Ankara, Turkey
sanem.elbasi@bilkent.edu.tr

Hakan Ferhatosmanoglu
Computer Engineering Department
Bilkent University
Ankara, Turkey
hakan@cs.bilkent.edu.tr

ABSTRACT

Deriving meaningful insights from location data helps businesses make better decisions. One critical decision made by a business is choosing a location for its new facility. Optimal location queries ask for a location to build a new facility that optimizes an objective function. Most of the existing works on optimal location queries propose solutions to return best location when the set of existing facilities and the set of customers are given. However, most businesses do not know the locations of their customers. In this paper, we introduce a new problem setting for optimal location queries by removing the assumption that the customer locations are known. We propose an optimal location predictor which accepts partial information about customer locations and returns a location for the new facility. The predictor generates synthetic customer locations by using given partial information and it runs optimal location queries with generated location data. Experiments with real data show that the predictor can find the optimal location when sufficient information is provided.

CCS CONCEPTS

•Information systems →Data analytics; *Uncertainty;* **•Theory of computation →***Facility location and clustering;*

KEYWORDS

Location Analytics; Optimal Location Queries; Uncertainty; Prediction; Data Generation

1 INTRODUCTION

Location analytics is the process or the ability to gain insight from the location data. Businesses use location analytics in many ways [8] such as finding the best place to locate a new facility, identifying the performances of stores, analyzing sales in different regions to offer products and prices most suitable for these regions, and managing insurance risks based on the potential of disasters in given locations. In this work, we consider the problem of selecting the optimal location which is a common location-based analysis that seeks the best location to open a new facility optimizing an objective function given a set of existing facilities and a set of customers. The objective function may vary depending on the aim of the business. For instance, a business may want to maximize the customers attracted by the new facility. In that case, the new facility must be the closest facility of as many customers as possible.

KDD'17, August 13–17, 2017, Halifax, NS, Canada.

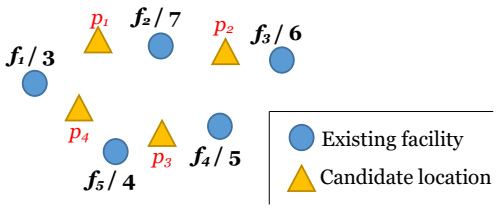


Figure 1: An example scenario for the problem.

Another objective can be minimizing the average distance of the customers to their closest facilities. For instance, delivery services pay attention to decreasing the average distance to reduce their logistics costs.

Previous works on optimal location queries focus on returning the best candidate as fast as possible [6, 15, 21]. Some of these works select the optimal location from a given region, whereas the others select from a set of candidate locations. The common approach is to use pruning based algorithms and index structures to decrease the processing times, instead of sequentially checking each possible location. The methods in the literature mostly find the optimal location when the locations of existing facilities and the locations of customers are given. Hence, businesses need to know the locations of their customers in order to use these algorithms. However, this is rarely the case. Most businesses do not have the knowledge of customer locations. For example, fast food restaurant chains or coffeehouse chains typically do not know the addresses of their customers. Therefore, when these businesses plan to open new branches, they cannot use the existing techniques for finding the optimal location.

In this paper, we introduce a new setting for the optimal location problems: A business that wants to find the optimal location for its new facility does not know the location of its customers. Instead, some partial information is known by the business such as the total number of customers attracted by each existing facility. Although many businesses do not know exact locations of their customers, they naturally have the number of customers for each existing facility. Figure 1 shows an example scenario for the addressed problem, where a business has five existing facilities and it has the knowledge of total number of customers attracted by each facility. For instance, there are 3 customers whose nearest facility is f_1 in Figure 1. The business needs to decide the best location among the candidates to open a new facility.

We develop a method that generates synthetic customer locations by using the partial information known by the business and

predicts the optimal location after running the query on generated location data. Customer locations are generated based on the total number of customers attracted by each facility. We form the Voronoi diagram for existing facilities and generate the customers of each facility in its Voronoi region. Instead of just uniformly distributing the customers within each Voronoi region, we use the density of customers in neighbor facilities in Voronoi diagram by dividing the Voronoi region of each facility into triangular regions and generating the customer locations in these smaller regions. It is possible to apply other constraints in the generation of customer location data such as removing the areas where no one lives (e.g. seas and forests). We performed experiments on real datasets containing Foursquare check-ins in New York City and Tokyo to show how each additional information increases the accuracy of the predictor.

The key contributions of the paper are summarized as follows:

- We study the optimal location selection problem by removing the assumption that the customer locations are known to businesses.
- We develop an optimal location predictor for choosing a location for the new facility by generating customer locations based on the density of the customers in each existing facility and the given auxiliary information.
- Our experiments with real location data from New York City and Tokyo show that the proposed predictor finds the optimal location for the new facility among several candidates even though the customer locations are not known.

The rest of the paper is organized as follows. Related work is reviewed in Section 2. Section 3 formulates the problem. We explain the optimal location predictor in Section 4 and evaluate the performance of the predictor through experiments with real data in Section 5. Finally, Section 6 concludes the paper.

2 RELATED WORK

Nearest neighbor (NN) query is a well-studied problem with many variants in the literature [1, 7, 14]. Reverse nearest neighbor (RNN) query finds the set of points that have the query point as the nearest neighbor [11]. In most of the real-life applications bichromatic reverse nearest neighbor (BRNN) query is used. In BRNN, points are divided into two categories such as customers and facilities. Given a facility f, BRNN query finds the set of customers that have f as the nearest facility. BRNN query is a fundamental query for optimal location studies because generally it is assumed that each customer prefers her closest facility. Hence, BRNN of a facility is the set of customers who are attracted by that facility.

Identifying the optimal location for a new facility has been widely studied in the literature with applications in decision-support systems and strategic planning of businesses. An optimal location query asks for a location to build a new facility that optimizes an objective function. For different types of facilities, different objective functions can be used. We consider two different objectives in this work: (i) *max-inf:* maximizing total number of customers attracted by the new facility and (ii) *min-dist:* minimizing the average distance between each customer and her nearest facility.

Max-inf optimal location query: Given a set $\mathcal{F}$ of existing facilities and a set C of customers, max-inf optimal location query finds a location p for the new facility with maximum influence. In [6], the influence of a location is defined as the total weight of its BRNN. Each customer has a weight and the query computes a location p in a given region Q which maximizes the total weight of customers who are closer to p than to any facility. The problem is studied in L_1-norm space and the authors propose methods using different index structures such as R^*-tree, OL-tree and virtual OL-tree. Maximizing the BRNN of the new facility in L_2-norm space is studied in [19]. Utilizing the region-to-point transformation, the authors solve the problem by searching a limited number of points instead of searching all possible points in the space. The same problem is studied assuming that each facility has a given capacity [3]. Another study returns top-k locations from a set of candidate locations instead of the best one [10]. The general assumption in optimal location queries is that each customer prefers her closest facility. In [22], it is assumed that a customer tends to go to her k nearest facilities. Hence, a facility attracts customers if the facility is one of her k nearest facilities. They find an optimal location such that setting up a new facility attracts the maximum number of customers.

Min-dist optimal location query: Given a set $\mathcal{F}$ of existing facilities and a set C of customers, min-dist optimal location query finds a location p such that the average distance from each customer to her closest facility is minimized if the new facility is built at location p. This query is widely used in real-life applications to improve the quality of service or reduce the logistics cost by businesses. It is firstly defined in [21] to select the min-dist optimal location from a given region. Although there are infinite number of locations in a region, the authors prove that it is possible to limit the number of candidate locations in L_1-norm space and the exact result is included in finite number of candidate locations. Qi et al. [15] solve the problem in L_2-norm space for the set of candidate locations and investigate the variant of the problem called min-dist facility replacement problem. Instead of adding a new facility, replacing a facility is aimed in facility replacement problem. Algorithms to solve optimal location queries in road networks have also been studied [4].

Previous works on optimal location queries select the optimal location either from a given region [19, 21] or from a candidate location set [10, 15]. When it is selected from a given region, infinite number of candidate locations is firstly limited. Then it becomes possible to search limited number of candidates. In this work, we select the optimal location from given candidate locations because businesses typically choose the facility locations from several candidates in practice. In addition, existing works focus on efficiently returning the best candidate using pruning techniques and index structures. However, they return the optimal location when the exact customer locations are given. Our work differs from existing works because we remove the the assumption that the customer locations are known to businesses. We introduce a new problem setting in which businesses only know partial information about customer locations.

Query processing over uncertain data has been studied in the literature for different type of queries. Wang et al. [18] presents a survey about data uncertainty and the types of uncertain data queries. Uncertain top-k query returns most probable top-k answers [16]. Soliman et al. [16] propose query processing algorithms in

Table 1: Notations used in the paper.

$\mathcal{F}$	the set of existing facilities
C	the set of customers
$\mathcal{P}$	the set of candidate locations for the new facility
$d(a, b)$	distance between points a and b
$\mathcal{I}(p_i)$	the set of customers attracted by the new facility if it is built at location p_i
$BRNN(f_i)$	the set of customers attracted by the facility f_i for the given set of existing facilities
$\mathcal{A}(p_i)$	the average distance from each customer to her nearest facility if the new facility is built at location p_i
$\|\mathcal{S}\|$	the cardinality of set $\mathcal{S}$
x_p	abscissa of the point p in Euclidean space
y_p	ordinate of the point p in Euclidean space
$\mathcal{R}$	the region considered by the generator
$\mathcal{R}_i$	Voronoi region of the facility f_i
$\mathcal{R}_{i,j}$	triangular region in $\mathcal{R}_i$

which the answer of the query depends on both the tuple scores and probabilities. Tao et al. [17] define range queries on uncertain databases to return objects in a given region whose probability is greater than a given threshold, where each object has an imprecise location. They propose the concept of probabilistically constrained rectangle and an index structure U-Tree for efficiently processing uncertain range queries. The probabilistic nearest neighbor query is firstly proposed in [5]. In order to return all objects which can be the nearest neighbor of the query point with non-zero probability, their algorithm performs a pruning of objects which do not have a chance of nearest neighbor of the query point. Cheema et al. [2] formalize probabilistic reverse nearest neighbor query that returns the objects which can be the RNN of the query point with higher probability than a given threshold. They propose an algorithm using several pruning techniques such as half-space pruning, dominance pruning, metric-based pruning, and probabilistic pruning. Li et al. [12] investigate the problem of probabilistic RkNN query and proposes an efficient and scalable algorithm using probabilistic pruning and spatial pruning techniques. In all of these works, objects are associated with probabilities and the query results are computed based on these probabilities. Their approaches cannot be directly applied to our problem because there is no probability associated with customer locations. The only known information is the number of customers attracted by each existing facility. Hence, a customer can be located at any point in the Voronoi region of her nearest facility. To the best of our knowledge, our work is the first to address processing of optimal location queries under such uncertainty.

3 PROBLEM FORMULATION

We first define max-inf and min-dist optimal location queries and then list the partial and auxiliary information that may be known by businesses to run these queries. Table 1 summarizes the notations used in the paper. All the data objects are represented by points in Euclidean space.

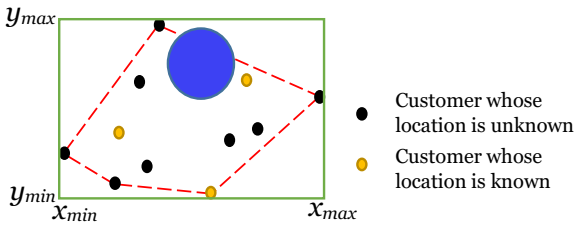


Figure 2: An example scenario for auxiliary information.

Definition 1. *Given a set $\mathcal{F}$ of existing facilities, a set C of customers, and a set $\mathcal{P}$ of candidate locations, the **max-inf optimal location query** finds a location $p \in \mathcal{P}$ for a new facility such that $\forall p' \in \mathcal{P}$,*

$$|\mathcal{I}(p)| \geq |\mathcal{I}(p')|$$

where $\mathcal{I}(p_i) = \{c \mid c \in C \wedge \forall f \in \mathcal{F}, d(c, p_i) \leq d(c, f)\}$.

Definition 2. *Given a set $\mathcal{F}$ of existing facilities, a set C of customers, and a set $\mathcal{P}$ of candidate locations, the **min-dist optimal location query** finds a location $p \in \mathcal{P}$ for a new facility such that $\forall p' \in \mathcal{P}$,*

$$\mathcal{A}(p) \leq \mathcal{A}(p')$$

where $\mathcal{A}(p_i) = \frac{\sum_{c \in C} \{d(c,f_i) \mid f_i \in \mathcal{F} \cup p_i \wedge \forall f_j \in \mathcal{F} \cup p_i,\ d(c,f_i) \leq d(c,f_j)\}}{|C|}$.

The above definitions of optimal location queries state that the set C must be provided. However, in our problem setting, the business that wants to run optimal location queries does not own the set of customer locations (C). We assume that the business knows the total number of customers attracted by each facility. Formally, for each facility $f \in \mathcal{F}$, $|BRNN(f)|$ is known by the business where $BRNN(f_i) = \{c \mid c \in C \wedge \forall f_j \in \mathcal{F}, d(c, f_i) \leq d(c, f_j)\}$. To run optimal location queries, the business can generate a set C' to mimic C based on the total number of customers attracted by each facility. However, businesses may have more yet partial information about customer locations. Here, we list auxiliary information (AI) that may be known by businesses and we explain how to use such partial information during the generation of customer locations in Section 4.

AI 1. *The business may know the overall **minimum and maximum values** for x and y coordinates in Euclidean space. These values can be represented as follows:*

- $x_{min} = min\{x_c \mid c \in C\}$
- $y_{min} = min\{y_c \mid c \in C\}$
- $x_{max} = max\{x_c \mid c \in C\}$
- $y_{max} = max\{y_c \mid c \in C\}$

AI 1 provides the minimum bounding rectangular region for the customer locations. Figure 2 shows an example in which the black and orange points represent customers. AI 1 indicates that all customer data are inside the green rectangle for the example in Figure 2.

AI 2. *The business may know the **minimum bounding convex polygon** of the customer locations.*

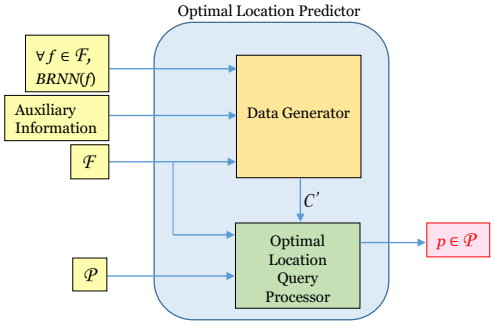


Figure 3: Optimal Location Predictor.

AI 2 provides the convex hull for the customer locations. In Figure 2, the customer locations are bounded with a pentagon drawn with red dotted lines. Hence, the data generator should generate the all customers inside this polygon if AI 2 is known.

AI 3. *The business may know* ***empty regions*** *which does not contain any customer.*

The business can avoid generating synthetic customer data in regions where no one lives (e.g. seas and forests). For instance, blue circle in Figure 2 represents a lake. Therefore, the data generator should not generate a customer location inside this region.

AI 4. *The business may know a* ***subset of*** C.

Although the business does not know C in the problem setting, locations of some customers may be known. In Figure 2, orange points represent the customers whose locations are known by the business. Therefore, during data generation it is enough to generate the locations for the other customers, who are represented with black points. In Section 4, we present the proposed predictor and explain the usage of auxiliary information during data generation. We also analyze the effect of each one on the accuracy of the predictor in Section 5.

4 PREDICTING OPTIMAL LOCATION

In this section, we present our optimal location prediction mechanism, when the business knows only $|BRNN(f)|$ for each facility $f \in \mathcal{F}$. The business may also know auxiliary information about customer locations. To run optimal location queries, $\mathcal{F}$, C, and $\mathcal{P}$ must be given. Since the business does not own the set C, we propose a location data generator to produce synthetic customer locations C' that mimics C. The query processor then returns the optimal location p for given $\mathcal{F}$, C', and $\mathcal{P}$. Figure 3 shows how our predictor works.

Along with the $|BRNN(f)|$ for each facility $f \in \mathcal{F}$, the data generator needs a region $\mathcal{R}$ for generating customers in this region. If the business knows AI 1, the data generator uses the minimum bounding rectangle as $\mathcal{R}$. Otherwise, the business selects a region $\mathcal{R}$ that will include all synthetic customer locations. To represent $\mathcal{R}$ in figures clearly, we used a rectangular region. However, it is not necessary to use a rectangular region. In this region $\mathcal{R}$, the generator locates the existing facilities ($\mathcal{F}$) and creates the

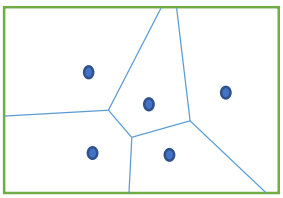


Figure 4: An example region $\mathcal{R}$ after Voronoi Diagram is created.

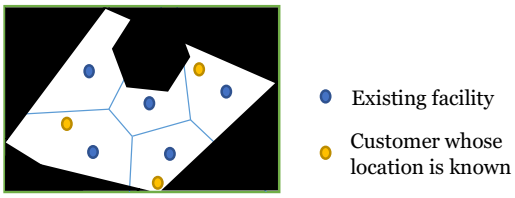


Figure 5: An example region $\mathcal{R}$ after auxiliary information is considered.

Voronoi diagram which is a partitioning of a plane into convex polygons such that each polygon contains one existing facility $f_i \in \mathcal{F}$. Voronoi region of each facility $f_i \in \mathcal{F}$ is the set of all points in $\mathcal{R}$ whose distance to f_i is not greater than their distance to the other facilities. Formally, the Voronoi region of facility $f_i \in \mathcal{F}$ is

$$\mathcal{R}_i = \left\{ r \in \mathcal{R} \mid \forall f_j \in \mathcal{F},\ d(r, f_i) \leq d(r, f_j) \right\}$$

An example Voronoi diagram for 5 facilities can be seen in Figure 4. After creating the Voronoi diagram, the generator identifies the regions in $\mathcal{R}$ which do not contain any customer by checking AI 2 and AI 3. If AI 2 is provided, the generator eliminates the regions in $\mathcal{R}$ but not in the minimum bounding polygon during data generation. If some other empty regions which do not contain a customer (i.e. AI 3) are provided, the generator also eliminates these regions. In Figure 5, these eliminated regions are represented with black. For AI 3, the generator accepts empty regions as polygons. Hence, the business enters the coordinates of the vertices of the polygons for AI 2 and AI 3. In addition, if the business knows a subset of C (i.e. AI 4), the locations of these customers are inserted into $\mathcal{R}$. In Figure 5, orange points represent the customers whose locations are known by the business. Therefore, they will be included in C'.

After considering auxiliary information, the data generator starts generating customer locations for each facility $f_i \in \mathcal{F}$. For a facility f_i, the generator needs to generate $|BRNN(f_i)|$ customers in its Voronoi region $\mathcal{R}_i$. $\mathcal{R}_i$ is a convex polygon and each edge of the polygon is either a common edge with a neighbor facility or a segment of an edge of $\mathcal{R}$. It is expected that there are more customers in the subregions of $\mathcal{R}_i$ which are close to neighbor facilities with high density of customers. Hence, rather than assigning these points uniformly random in each Voronoi region, we use the number of customers attracted by each neighbor facility in Voronoi diagram by assigning a weight. To use the density of customers in neighbors, we divide the region of each facility into triangular

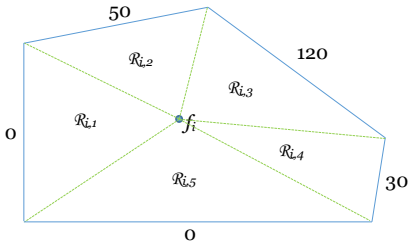


Figure 6: Dividing $\mathcal{R}_i$ into triangular regions.

regions by connecting each facility $f_i \in \mathcal{F}$ with the vertices of its Voronoi region $\mathcal{R}_i$. For the given example in Figure 6, $\mathcal{R}_i$ is divided into 5 triangular regions. The data generator decides the total number of customers to be generated in each triangular region based on:

(1) the area of the region,
(2) the total number of customers attracted by its neighbor facility.

Let the total number of triangular regions in $\mathcal{R}_i$ be m_i and these regions be $\{\mathcal{R}_{i,1}, ..., \mathcal{R}_{i,m_i}\}$. For a region $\mathcal{R}_{i,j}$, let $BRNN$ of its neighbor be $n_{i,j}$. Let $n_i = \sum_{k=1}^{m_i} n_{i,k}$. Hence, n_i is the total number of customers attracted by all of the neighbors of f_i. Then, the total number of customers to be generated in a triangular region $\mathcal{R}_{i,j}$ is calculated as

$$|BRNN(f_i)| \cdot \left(\omega \cdot \frac{n_{i,j}}{n_i} + (1-\omega) \cdot \frac{Area(\mathcal{R}_{i,j})}{Area(\mathcal{R}_i)} \right)$$

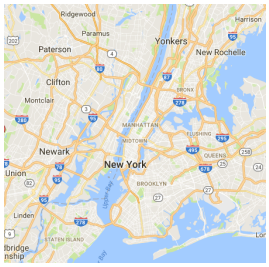
In this formula, ω is the weighting factor that represents the effect of $n_{i,j}$ on the total number of customers to be generated in $\mathcal{R}_{i,j}$. When ω is selected as 0, the generator distributes customers with respect to the area of each triangle in $\mathcal{R}_i$ without considering the number of customers attracted by neighbors.

For instance, if $|BRNN(f_i)|$ is 50 and $\frac{Area(\mathcal{R}_{i,3})}{Area(\mathcal{R}_i)}$ is $\frac{1}{5}$ in Figure 6, the generator generates $50 \cdot \left(0.5 \cdot \frac{120}{200} + (1-0.5) \cdot \frac{1}{5}\right) = 20$ customers in $\mathcal{R}_{i,3}$ if ω is selected as 0.5. For different values of ω in the range of $[0, 1]$, the total number of customers to be distributed in $\mathcal{R}_{i,3}$ varies between 10 and 30.

By using the given formula, the generator decides the number of customers in each triangular region and produces the customer locations. To produce a random location inside a triangle, one can select three random points s_1, s_2, s_3 in the range of [0,1] such that $s_1 + s_2 + s_3 = 1$ and use these three points as barycentric coordinates of the random point inside the triangle. For a triangle with vertices P_1, P_2, and P_3, the random point can be determined as $s_1 \cdot P_1 + s_2 \cdot P_2 + s_3 \cdot P_3$.

If the locations of some customers are given as auxiliary information (i.e. AI 4), the generator generates the locations for the other customers. If some part of the triangular region is removed by AI 2 or AI 3, the area of the remaining region is considered in the formula.

After generating synthetic customer locations, optimal location query is executed by the predictor. In max-inf optimal location query, the size of the influence set ($|\mathcal{I}(p_i)|$) for each candidate



(a) New York City

(b) Tokyo

Figure 7: The regions covering all customer locations on map.

$p_i \in \mathcal{P}$ is calculated. The candidates are ranked with respect to sizes of their influence sets and the candidate with maximum size is returned as the best candidate. In min-dist optimal location query, the average distance ($\mathcal{A}(p_i)$) from each customer to nearest facility is calculated if the new facility is built at the location p_i. Similarly, the candidates are ranked with respect to the average distance values and the candidate with minimum value is returned as the best candidate.

5 EXPERIMENTAL RESULTS

In our experiments, we used datasets [20] containing 227,428 check-ins in New York City and 573,703 check-ins in Tokyo collected from Foursquare from 12 April 2012 to 16 February 2013. Each check-in in the datasets contains time stamp, GPS coordinates, and venue information. We only used GPS coordinates and we considered each check-in as a separate customer. Hence, there are 227,428 customers in C_{NYC} and 573,703 customers in C_{TKY}. For existing facilities, we used the locations of 97 McDonald's restaurants in New York ($\mathcal{F}_{NYC}$) and 76 Yoshinoya restaurants in Tokyo ($\mathcal{F}_{TKY}$). Figure 7a and 7b show the whole regions containing customer locations on map for New York City and Tokyo, respectively. We divided the whole region into a 10x10 grid for each city and selected the center of each grid as a candidate location for the new facility. We removed the candidates that are in empty regions (e.g. seas). Hence, $\mathcal{P}_{NYC}$ and $\mathcal{P}_{TKY}$ contain 69 and 72 candidate locations, respectively.

We implemented our predictor to evaluate its accuracy for max-inf optimal location query and min-dist optimal location query. Initially, we executed these queries using real customer locations (C_{NYC} and C_{TKY}) and we ranked all candidate locations ($\mathcal{P}_{NYC}$ and $\mathcal{P}_{TKY}$) with respect to their optimalities. We determined the best candidates for max-inf optimal location query and min-dist optimal location query. Let r_i be the ranking of the candidate location p_i when real customer location data is used. To observe the accuracy of the predictor, we counted the total number of customers attracted by each existing facility in $\mathcal{F}_{NYC}$ and $\mathcal{F}_{TKY}$. We provided these values ($BRNN(f)$ for each facility $f \in \mathcal{F}_{NYC}$ and $f \in \mathcal{F}_{TKY}$) to the predictor together with auxiliary information. The data generator produced synthetic customer locations (C'_{NYC} and C'_{TKY}) and we observed the rankings of the candidate locations when synthetic data is used in optimal location queries. Let

r'_i be the ranking of the candidate location p_i returned from the predictor. We evaluate the accuracy of the predictor by measuring the standard deviation of the rankings with the following formula:

$$\sqrt{\frac{\sum_{i=1}^{|\mathcal{P}|}(r_i - r'_i)^2}{|\mathcal{P}|}}$$

where $|\mathcal{P}|$ is the number of candidates. We ran the predictor several times to show the effect of auxiliary information on the accuracy of the predictor. We also ran the predictor with different ω values to observe the effect of ω on accuracy. We present the evaluation results for max-inf optimal location query and min-dist optimal location query in Section 5.1 and 5.2, respectively. For each query type, we firstly present the results for $\omega = 0.5$ and then show the effect of ω on the accuracy.

We also illustrate the ranking of the candidates with figures. In these figures, the red plus signs represent the existing facilities ($\mathcal{F}_{NYC}$ and $\mathcal{F}_{TKY}$), the gray circles represent the customers (C_{NYC} and C_{TKY}), the blue diamonds represent candidate locations ($\mathcal{P}_{NYC}$ and $\mathcal{P}_{TKY}$), and the blue lines show the boundaries of the Voronoi regions of existing facilities. We marked the best candidates with circles and second best and third best candidates with rectangles. In addition, we show the ranking of the candidates as $p_i{:}j$, in which p_i refers to a candidate location and j refers to its ranking.

5.1 Max-Inf Optimal Location Query

Max-inf optimal location query returns a candidate location p_i which maximizes the total number of customers attracted by the new facility if it is built at the location p_i. Figure 8 and 9 show the rankings of the candidate locations in $\mathcal{P}_{NYC}$ and $\mathcal{P}_{TKY}$ when real customer locations (C_{NYC} and C_{TKY}) are used.

In New York City, the best candidate for maximizing the total number of customers attracted by the new facility is p_{32} as shown in Figure 8. It attracts 5,341 customers. The other candidates in top five are p_{46}, p_{30}, p_{47}, and p_{41}, and the total number of customers attracted by these candidates are 4,599, 3,551, 3,321, and 3,025, respectively.

In Tokyo, the best candidate returned from max-inf optimal location query is p_{53} and the total number of customers attracted by the new facility is 42,411 if it is built at the location p_{53}. The other candidates in top five are p_{32}, p_9, p_{46}, and p_{55}, and the total number of customers attracted by these candidates are 13,528, 13,384, 13,338, and 10,458, respectively.

In the evaluation of the predictor, we provided $BRNN(f)$ for each facility $f \in \mathcal{F}_{NYC}$ and $f \in \mathcal{F}_{TKY}$ to the predictor. Figure 10 and 11 show the rankings when minimum and maximum coordinates (i.e. AI 1) are also provided to the predictor. For both cities, the predictor returns the same best candidate with the knowledge of AI 1. The predictor estimates the total number of customers attracted by p_{32} as 11,357 in New York City and the total number of customers attracted by p_{53} as 33,989 in Tokyo. In New York City, the predictor also finds the same second best candidate correctly. The standard deviations in the rankings for New York City and Tokyo are 14.4272 and 12.9271, respectively.

When we also provide AI 2 and AI 3 to the predictor, it still returns the same best candidates as shown in Figure 12 and 13.

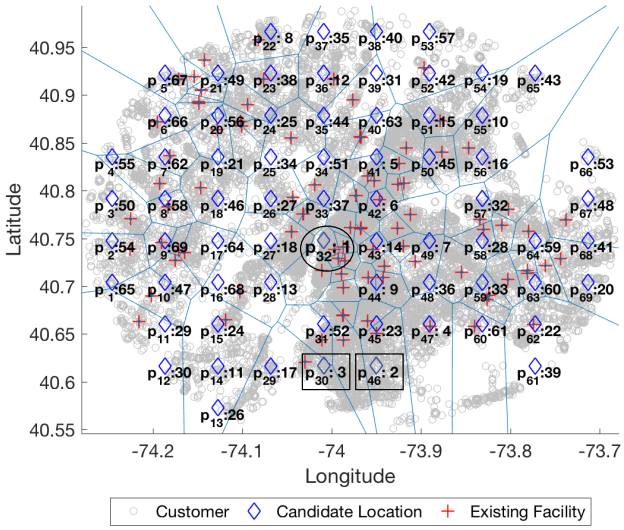


Figure 8: Ranking of candidate locations in NYC when real data is used in max-inf optimal location query.

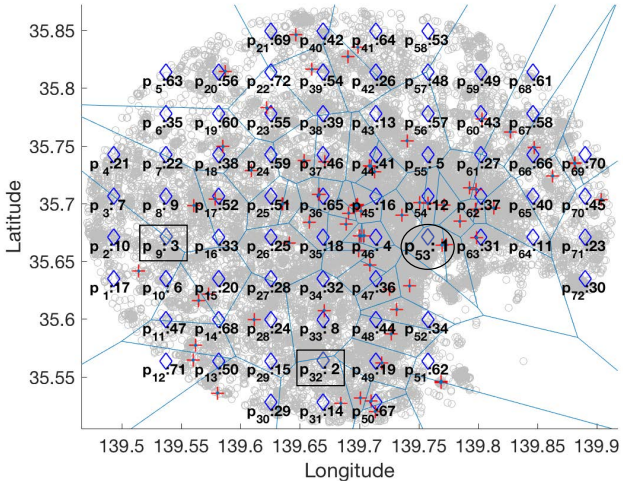


Figure 9: Ranking of candidate locations in Tokyo when real data is used in max-inf optimal location query.

Moreover, using AI 2 and AI 3 decreases the standard deviation of the rankings. The standard deviation decreases from 14.4272 to 12.2451 in New York and decreases from 12.9271 to 11.6583 in Tokyo. This result indicates that providing more information to the predictor improves the accuracy in the rankings, as expected.

To experiment with the case where the locations of some customers are known (i.e. AI 4), we provided varying number of customer locations to the predictor. As evident in Figure 14a, increasing the ratio of known customers decreases the standard deviation of the rankings. For instance, when 50% of the customer locations are known, the standard deviation decreases to nearly 7 in both cities.

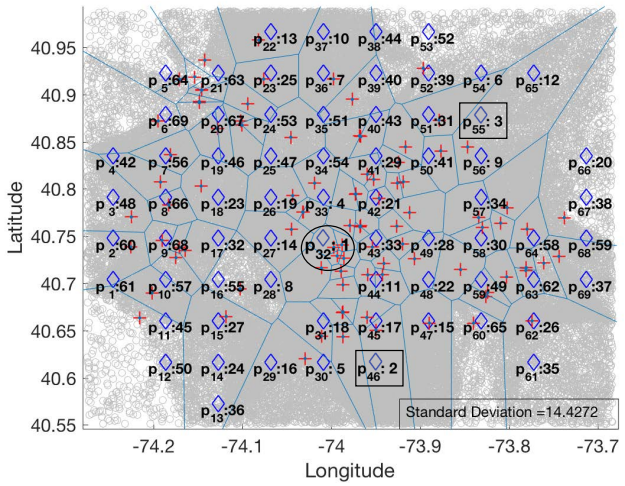


Figure 10: Ranking of candidate locations in NYC when the predictor uses AI 1 in max-inf optimal location query.

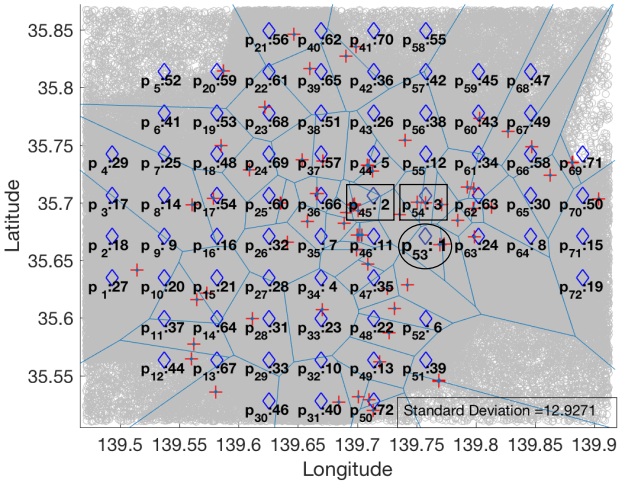


Figure 11: Ranking of candidate locations in Tokyo when the predictor uses AI 1 in max-inf optimal location query.

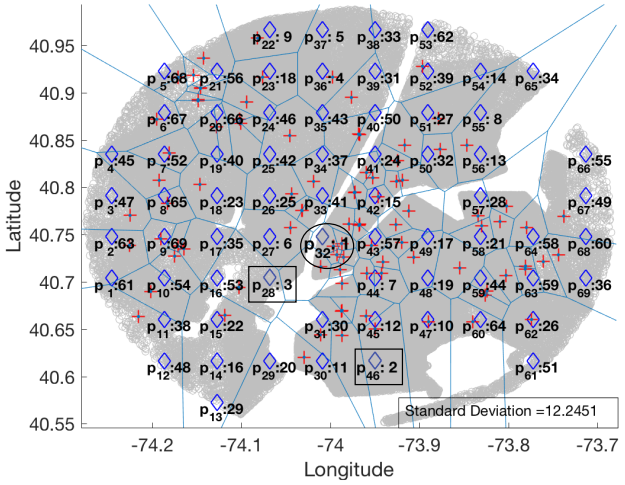


Figure 12: Ranking of candidate locations in NYC when the predictor uses AI 2 and AI 3 in max-inf optimal location query.

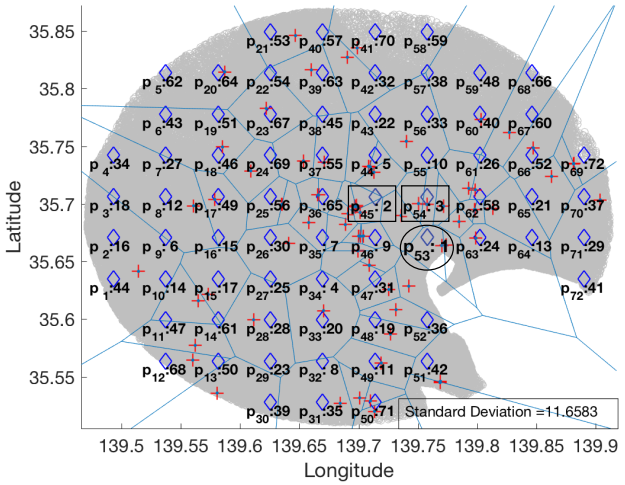
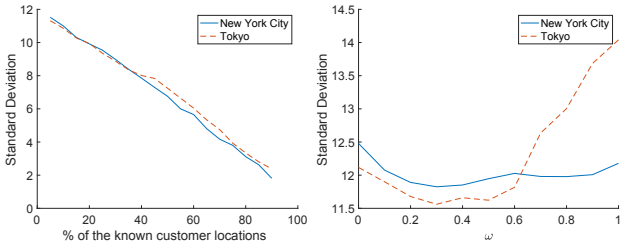


Figure 13: Ranking of candidate locations in Tokyo when the predictor uses AI 2 and AI 3 in max-inf optimal location query.

We also conducted experiments to observe the impact of ω on the standard deviation. As it is mentioned in Section 4, when ω is equal to 0 the distribution is only based on the areas of the triangles. Hence, we use $\omega = 0$ as the baseline which provides a distribution in Voronoi region that is similar to uniform distribution. Figure 14b shows the standard deviation for different values of ω between 0 and 1 when AI 2 and AI 3 are provided to the predictor. For both cities, minimum standard deviation is obtained when ω is selected as 0.3. The standard deviation is 11.8248 in New York City and 11.5614 in Tokyo when ω is equal to 0.3. We also analyzed the rankings and we observed that the predictor's top five candidates are same for $\omega = 0.3$ and $\omega = 0.5$. As evident in Figure 14b, best accuracy is achieved when ω value is in the range of $[0.2, 0.5]$. The standard deviation is lower than the baseline ($\omega = 0$) when ω is selected in this range.

To evaluate the accuracy of the predictor when no AI is known, we provided larger regions than the minimum bounding rectangle (i.e. AI 1). We expanded the height and width of the minimum bounding rectangle iteratively and Figure 15 shows the standard deviation for different expansion percentages. For instance, when we provided a rectangular region whose height and width are 20% greater than the minimum bounding rectangle, the standard deviation increases to nearly 15 in Tokyo and nearly 16 in New York City. As expected, providing larger regions increases the standard



(a) Impact of AI 4 (b) Impact of ω

Figure 14: Impact of AI 4 and ω on the standard deviation of rankings in max-inf optimal location query.

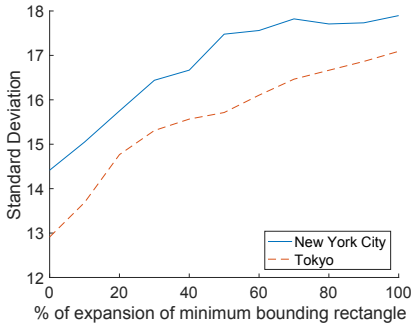


Figure 15: Standard deviation of rankings when no AI is known in max-inf optimal location query.

deviation. However, the predictor returns the same best candidates for both cities without using auxiliary information, because max-inf optimal location query returns the candidate which attracts maximum amount of customers without considering the distances from customers to their nearest facilities. Therefore, the effect of generating customers outside the minimum bounding rectangle on the best candidate is low in max-inf optimal location query.

5.2 Min-Dist Optimal Location Query

Min-dist optimal location query returns a candidate location p_i which minimizes the average distance between each customer and her nearest facility if the new facility is built at the location p_i. We conducted the same set of experiments for this query as well. Figure 16 shows the ranking of candidates ($\mathcal{P}_{NYC}$) in NYC when the real customer locations (C_{NYC}) are used in min-dist optimal location query. In New York City, the average distance of customers to their nearest facilities are minimized if the new facility is built at p_{46}. The average distance becomes 1.4433 km if p_{46} is selected as the location of the new facility. The other candidates in top five are p_{65}, p_{14}, p_{30}, and p_{54}, and building a new facility at these locations decreases the average distances to 1.4567 km, 1.4587 km, 1.4588 km, and 1.4611 km, respectively.

The ranking of candidates ($\mathcal{P}_{TKY}$) in Tokyo is given in Figure 17 when the real customer locations (C_{TKY}) are used. In Tokyo, the best candidate for min-dist optimal location query is p_3. The average distance becomes 1.3346 km, if the new facility is built at p_3. The other candidates in top five are p_2, p_{32}, p_{72}, and p_{53}, and building a

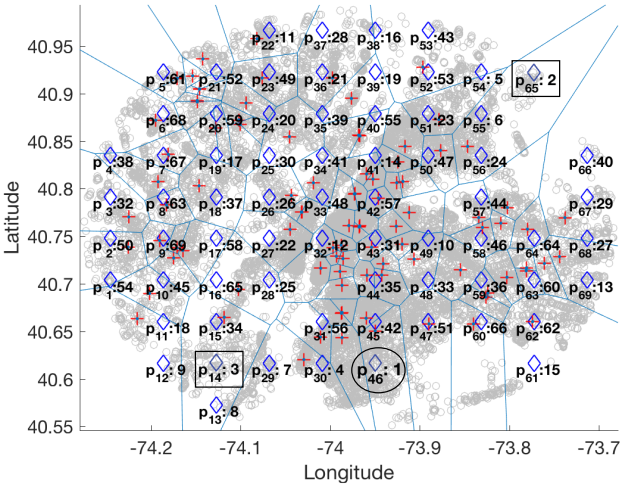


Figure 16: Ranking of candidate locations in NYC when real data is used in min-dist optimal location query.

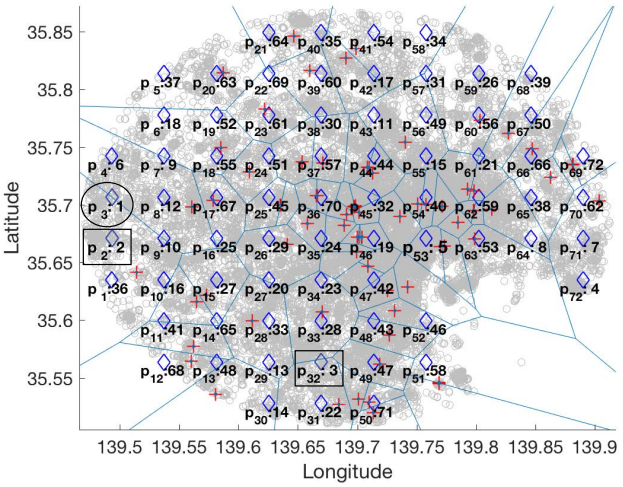


Figure 17: Ranking of candidate locations in Tokyo when real data is used in min-dist optimal location query.

new facility at these locations decreases the average distances to 1.3431 km, 1.3436 km, 1.346 km, and 1.3475 km, respectively.

Table 2 shows the top five candidates for both cities according to the predictor with only AI 1. In New York City, the predictor returns p_{65} as the best candidate, which is actually the second best candidate as shown in Figure 16. The real best candidate (p_{46}) is ranked third by the predictor. In Tokyo, the predictor returns p_{72} as the best candidate; however, its actual rank is 5. The real best candidate (p_3) is ranked second by the predictor. The standard deviation is 12.7632 in New York City and 12.5266 in Tokyo, when only AI 1 is provided to the predictor.

For min-dist optimal location query, only AI 1 is not sufficient for the predictor to return the same best candidate. Since the predictor only uses AI 1, it generates customers in empty areas such as seas.

Table 2: Top five candidate locations when the predictor uses AI 1 in min-dist optimal location query.

	New York City		Tokyo	
Rank	Candidate	Avg. Dist.	Candidate	Avg. Dist.
1	p_{65}	1.8667 km	p_{72}	1.8319 km
2	p_{54}	1.8667 km	p_3	1.8354 km
3	p_{46}	1.891 km	p_{64}	1.8371 km
4	p_{55}	1.8949 km	p_4	1.8387 km
5	p_{32}	1.9011 km	p_{52}	1.8462 km

Table 3: Top five candidate locations when the predictor uses AI 2 and AI 3 in min-dist optimal location query.

	New York City		Tokyo	
Rank	Candidate	Avg. Dist.	Candidate	Avg. Dist.
1	p_{46}	1.6235 km	p_3	1.6487 km
2	p_{54}	1.6265 km	p_{64}	1.6549 km
3	p_{55}	1.6286 km	p_{53}	1.656 km
4	p_{37}	1.6317 km	p_2	1.6593 km
5	p_{28}	1.6333 km	p_4	1.6605 km

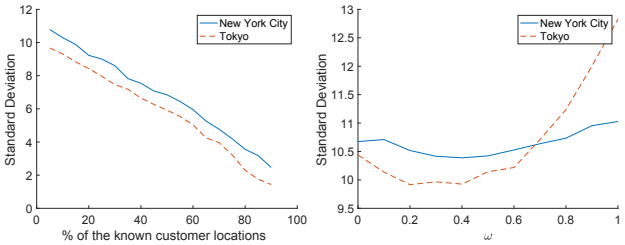
Therefore, distance from a customer to her nearest facility is usually higher than the real one, which affects the accuracy considerably. AI 2 and AI 3 should be provided to the predictor to achieve a better accuracy.

Table 3 shows top five candidates according to the predictor, when we provided AI 2 and AI 3 to the predictor. It found the same best candidates for both New York City and Tokyo. The average distance values are closer to the real values, when the predictor uses AI 2 and AI 3. The standard deviation also decreases from 12.7632 to 11.0362 in New York and decreases from 12.5266 to 10.1009 in Tokyo.

Similar to max-inf optimal location query, the standard deviation of the rankings is inversely proportional to the ratio of known customer locations (i.e. AI 4). Standard deviation for different values of percentage of known customer locations is given in Figure 18a. As evident in Figure 18a, the accuracy of the predictor increases when the locations of more customers are provided to the predictor.

Figure 18b shows the standard deviation for different values of ω between 0 and 1 when AI 2 and AI 3 are provided to the predictor in min-dist optimal location query. In New York City, minimum standard deviation (10.3881) is obtained when $\omega = 0.4$. In Tokyo, standard deviation is minimum (9.9163) when $\omega = 0.2$. In both cities, the best accuracy is achieved when ω value varies between 0.2 and 0.5. Similar to max-inf optimal location query, selecting ω value in the range of $[0.2, 0.5]$ provides better accuracy than the baseline ($\omega = 0$). Moreover, when we analyze the rankings of candidate locations, the predictor's top five candidates are same for all ω values in this range. Therefore, ω should be selected between 0.2 and 0.5 to improve accuracy.

Figure 19 depicts the standard deviation of the rankings when the given region to the predictor is larger than the minimum bounding rectangle. As in max-inf optimal location query, the standard deviation increases when the size of the region increases. Unlike



(a) Impact of AI 4 (b) Impact of ω

Figure 18: Impact of AI 4 and ω on the standard deviation of rankings in min-dist optimal location query.

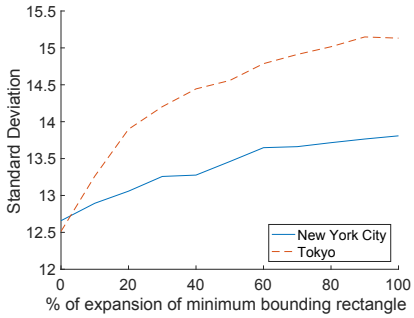


Figure 19: Standard deviation of rankings when no AI is known in min-dist optimal location query.

max-inf optimal location query, the predictor does not return the same best candidates when no auxiliary information is provided. Therefore, providing auxiliary information in min-dist optimal location query is more important than max-inf optimal location query to find the same best candidate.

6 CONCLUSION

We have proposed an optimal location predictor which does not require the customer locations. By using the density of the customers in each existing facility and the given auxiliary information, it returns a candidate location from a set of candidates. After generating customer locations based on given information, the predictor runs a query for finding the best location. During data generation, the predictor does not simply distribute customers uniformly, it considers the density of customers in neighbor facilities that are selected from the Voronoi diagram of the facilities. Two facilities are neighbors of each other, if their Voronoi regions share a common edge. Hence, the data generator divides the Voronoi region of each existing facility into smaller triangular regions and generates customer locations in each smaller region. We performed experiments with real datasets to evaluate the accuracy of the optimal location predictor. The predictor found the real best candidate in both max-inf and min-dist optimal location queries when the convex hull of customer locations (i.e. minimum bounding polygon) and the empty regions in the cities are given. Hence, it is useful to know the boundaries of the region containing customer locations to obtain accurate results. In addition, our experiment results indicate that ω

value used in data generation should be selected between 0.2 and 0.5 to achieve high accuracy.

Since the predictor generates location data randomly, it may not return the best candidate in the following cases:

- if the difference of optimality scores of top two candidates is low. The optimality score of a candidate p_i is calculated as $|\mathcal{I}(p_i)|$ in max-inf optimal location query, and $\mathcal{A}(p_i)$ in min-dist optimal location query. For instance, in max-inf optimal location query, if the best candidate attracts 350 customers and the second best candidate attracts 348 customers, the predictor may not return the real best candidate.
- if the total number of existing facilities (i.e. $|\mathcal{F}|$) is low.
- if the existing facilities have a highly skewed distribution.

In such cases, knowing the locations of some customers by businesses increases the chance of returning the best one. The experiment results show that providing more information improves the accuracy of the predictor. The proposed predictor facilitates running optimal location queries by businesses without knowing their customers' locations.

The proposed approach can be applied to different optimization problems when data is not available. If there is partial information about data such as the number of items in different clusters, synthetic data can be generated similarly and it can be used in optimization. Hence, generating synthetic data for different optimization problems and evaluating their optimization performance is a potential follow up of this work. Another follow up work is to apply bootstrap methods for data generation and evaluating their accuracy for the case where the locations of some customers are known. These methods allow increasing the data size by generating new samples based on the original samples. Therefore, bootstrap methods for spatial data [9, 13] can also be potentially used for data generation if a subset of customer locations (i.e. AI 4) is known.

ACKNOWLEDGMENTS

Hakan Ferhatosmanoglu was supported in part by the Alexander von Humboldt Foundation.

REFERENCES

[1] Christian Böhm and Florian Krebs. 2004. The k-nearest neighbour join: Turbo charging the KDD process. *Knowledge and Information Systems* 6, 6 (2004), 728–749.

[2] Muhammad Aamir Cheema, Xuemin Lin, Wei Wang, Wenjie Zhang, and Jian Pei. 2010. Probabilistic reverse nearest neighbor queries on uncertain data. *IEEE Transactions on Knowledge and Data Engineering* 22, 4 (2010), 550–564.

[3] Fangshu Chen, Huaizhong Lin, Yunjun Gao, and Dongming Lu. 2016. Capacity constrained maximizing bichromatic reverse nearest neighbor search. *Expert Systems with Applications* 43 (2016), 93–108.

[4] Zitong Chen, Yubao Liu, Raymond Chi-Wing Wong, Jiamin Xiong, Ganglin Mai, and Cheng Long. 2014. Efficient algorithms for optimal location queries in road networks. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 123–134.

[5] Reynold Cheng, Dmitri V Kalashnikov, and Sunil Prabhakar. 2003. Evaluating probabilistic queries over imprecise data. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. ACM, 551–562.

[6] Yang Du, Donghui Zhang, and Tian Xia. 2005. The optimal-location query. In *Advances in Spatial and Temporal Databases*. Springer, 163–180.

[7] Hakan Ferhatosmanoglu, Ioanna Stanoi, Divyakant Agrawal, and Amr El Abbadi. 2001. Constrained nearest neighbor queries. In *International Symposium on Spatial and Temporal Databases*. Springer, 257–276.

[8] Lee Garber. 2013. Analytics goes on location with new approaches. *Computer* 46, 4 (2013), 14–17.

[9] Pilar García-Soidán, Raquel Menezes, and Óscar Rubiños. 2014. Bootstrap approaches for spatial data. *Stochastic environmental research and risk assessment* 28, 5 (2014), 1207–1219.

[10] Jin Huang, Zeyi Wen, Jianzhong Qi, Rui Zhang, Jian Chen, and Zhen He. 2011. Top-k most influential locations selection. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2377–2380.

[11] Flip Korn and S Muthukrishnan. 2000. Influence sets based on reverse nearest neighbor queries. In *ACM SIGMOD Record*, Vol. 29. ACM, 201–212.

[12] Jiajia Li, Botao Wang, and Guoren Wang. 2013. Efficient probabilistic reverse k-nearest neighbors query processing on uncertain data. In *International Conference on Database Systems for Advanced Applications*. Springer, 456–471.

[13] Ji Meng Loh. 2008. A valid and fast spatial bootstrap for correlation functions. *The Astrophysical Journal* 681, 1 (2008), 726.

[14] Dimitris Papadias, Qiongmao Shen, Yufei Tao, and Kyriakos Mouratidis. 2004. Group nearest neighbor queries. In *Data Engineering, 2004. Proceedings. 20th International Conference on*. IEEE, 301–312.

[15] Jianzhong Qi, Rui Zhang, Yanqiu Wang, Andy Yuan Xue, Ge Yu, and Lars Kulik. 2014. The min-dist location selection and facility replacement queries. *World Wide Web* 17, 6 (2014), 1261–1293.

[16] Mohamed A Soliman, Ihab F Ilyas, and Kevin Chen-Chuan Chang. 2007. Top-k query processing in uncertain databases. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*. IEEE, 896–905.

[17] Yufei Tao, Xiaokui Xiao, and Reynold Cheng. 2007. Range search on multidimensional uncertain data. *ACM Transactions on Database Systems (TODS)* 32, 3 (2007), 15.

[18] Yijie Wang, Xiaoyong Li, Xiaoling Li, and Yuan Wang. 2013. A survey of queries over uncertain data. *Knowledge and information systems* 37, 3 (2013), 485–530.

[19] Raymond Chi-Wing Wong, M Tamer Özsu, Philip S Yu, Ada Wai-Chee Fu, and Lian Liu. 2009. Efficient method for maximizing bichromatic reverse nearest neighbor. *Proceedings of the VLDB Endowment* 2, 1 (2009), 1126–1137.

[20] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. 2015. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2015), 129–142.

[21] Donghui Zhang, Yang Du, Tian Xia, and Yufei Tao. 2006. Progressive computation of the min-dist optimal-location query. In *Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment, 643–654.

[22] Zenan Zhou, Wei Wu, Xiaohui Li, Mong Li Lee, and Wynne Hsu. 2011. Maxfirst for maxbrknn. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*. IEEE, 828–839.