Journal of AI and Data Mining
Vol 5, No 1, 2017, 101-109

DOI: 10.22044/jadm.2016.733

Non-zero probability of nearest neighbor searching

A. Mesrikhani [1*] and M. Davoodi [2]

Department of Computer Science & Information Technology, Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan, Iran.

Received 27 April 2015; Accepted 12 September 2016
**Corresponding author: mesrikhani@gmail.com (A. Mesrikhani).*

Abstract

Nearest neighbor (NN) searching is a challenging problem in data management, and has been widely studied in data mining, pattern recognition, and computational geometry. The goal of NN searching is an efficient report of the data nearest to a given object as a query. In most studies, both the data and the query are assumed to be precise. However, due to the real applications of NN searching such as the tracking and locating services, GIS, and data mining, it is possible for both the data and the query to be imprecise. In such situations, a natural way to handle the issue is to report the data that has a non-zero probability (called the *non-zero NN)* as the NN of a given query. Formally, let P be a set of n uncertain points modeled by some regions. We first consider the following variation in an NN searching problem under uncertainty. If the data is certain and the query is an uncertain point modeled by an axis-aligned parallel segment, we propose an efficient algorithm in $O(n\log n)$ pre-processing and $O(\log n+k)$ query time, where k is the number of non-zero NNs. If both the query and the data are uncertain points modeled by distinct unit segments parallel to the x-axis, we propose an efficient algorithm that reports the non-zero NNs under Manhattan metric in $O(n^2\alpha(n^2))$ pre-processing and $O(\log n+k)$ query time, where $\alpha(.)$ is the extremely slow growing functional inverse of the Ackermann function. Finally, for the arbitrarily length segments parallel to the x-axis, we propose an approximation algorithm that reports a non-zero NN with a maximum error L in $O(n^2\alpha(n^2))$ pre-processing and $O(\log n+k)$ query time, where L is the query length.

Keywords: *Nearest Neighbor Searching, Uncertainty, Imprecision, Non-zero Probability.*

1. Introduction

Nearest Neighbor (NN) searching, which is a classic problem in computational geometry, has many applications in robot path planning, facility location, data mining, target tracking, and geographic information systems. In this problem, the goal is the proper pre-processing of a set of n data points in order to report efficiently the data nearest to a given query point. Due to several reasons such as noise, security issues, limited computations, and limited precision of measuring devices, gathering and analyzing real data come with some inevitable errors. Thus the algorithms that work based on the assumption that the data (and also computations) are completely precise fail in face with such a real input [1, 2]. For example, in facial recognition systems, we need to identify a person using some features in a database containing original face images. Due to the nature of such problems, extracting the features from the original faces (in different positions or video frames), and also the query features are uncertain, and, therefore, the query does not match exactly to one of the original ones, and consequently, it should be handled under uncertainty circumstances [3]. One geometric approach implemented to smooth such uncertainty issues is to consider a tolerance for data, e.g. considering a region —called the *uncertainty region*— like a segment, a rectangle or a disk instead of an uncertain point [4, 5]. Thus an uncertain region is a region containing all the *instances* of an imprecise point. Therefore, since the distance between two imprecise points is not defined precisely, different cases may happen. In fact, the distance between two imprecise points can be defined as the distance between any selected instances from their uncertainty region, especially, the instances resulting in minimum and maximum distances. Such instances can be useful in applications for obtaining the worst and best cases of a solution under imperfect information or uncertain circumstances.

1.1. Problem definition

One way to consider NN searching under uncertainty is to report the data that has a non-zero probability to be an NN of a given query [8]. It means that there is at least one *placement* of instances of uncertain regions such that the reported data is NN of the query. Let $P = \{p_1, ..., p_n\}$ be a set of n uncertain points in a plane whose uncertainty regions are modeled by n regions, e.g. segments. (In this case, we assume that the data has some error only in one direction.) The uncertainty region of $p_i \in P$ is the set of all possible points (*instances*) in which p_i is located. For a query point q, we aim to report all points in P that have a *non-zero probability* to be the NN of q —called *non-zero NN*, denoted by *NzNN*. That means that when an uncertain point p is reported, there is a choice of points (called a *placement*) exactly one instance from each uncertainty region such that the instance of p is the nearest instance to q among all instances. Note that it is possible that q is also an uncertain point. Thus in this case, there is a placement of p and an instance of q like q' such that the instance of p is the nearest instance to q' among all instances (see Figure 1).

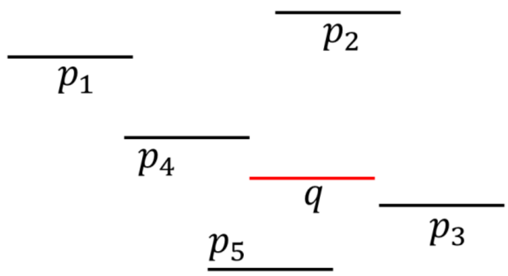

Figure 1. $\{p_3, p_4, p_5\}$ **are non-zero NN of uncertain query** q.

1.2. Previous work

Under the assumption that the data is precise, a simple and efficient method can be used to find that NN is a decomposing workspace using the Voronoi diagram of the data points in the $O(n \log n)$ time. Thus in the query phase for a given query point q, it is sufficient to report NN in the $O(\log n)$ time by locating q in the Voronoi regions and reporting the corresponding data [6]. For uncertain points and certain query, the original Voronoi diagram has been extended to the *non-zero probabilistic Voronoi diagram* (*PVD*) [7, 8]. Each cell in PVD contains the points that have non-zero probability to be the NN of the corresponding data. Sember et al. [7] have shown the worst case complexity of PVD for uncertain points modeled by disks is $O(n^4)$ but they did not compute any lower bound for its complexity. Agarwal et al. [8] have shown that if the query is certain and the points are uncertain regions modeled by disks, PVD can be built in the $\theta(n^3)$ time. Hence, it is possible to report NzNN in the $O(\log n + k)$ time, where k is the number of possible non-zero probability points. Also by applying the expected distance between the uncertain points, the problem can be solved in the $O(\log n)$ time using the $O(n)$ space and the $O(n \log n + nm)$ preprocessing time, where m is the number of possible values in the data [9]. Cheng et al. [10] have introduced a method based on branch and prune on the *R*-tree. Further, Zhang et al. [11] have shown that in *d*-dimensional, there is no polynomial algorithm to compute PVD, and they combined PVD and *R*-tree to propose a heuristic method to report NzNN. However, the method did not guarantee a proper performance. Emrich et al. [12,13] have presented an effective criterion for detecting NzNN, and proposed a heuristic method to report NzNN but their method did not guarantee any performance in the worst case.

Beside the region-based models used for modelling uncertainty, other models have been proposed as well. Davoodi et al. [14] have introduced a generalization of the region-based models —called the *λ-geometry* model— for handling a dynamic form of imprecision that allows the precision changes in the input data of the geometric problems. They have also studied the problems of proximity, bounding box, and orthogonal range searching under this model [14, 15]. Meyers et al. [16] have introduced a new model called the linear parametric geometric uncertainty model (LPGUM), and have proposed algorithms to find the closest and farthest pairs and range searching under LPGUM [17].

In some real applications, it is useful to report NN with the highest probability. Yuen et al. [18] have studied the superseding NN search on uncertain spatial databases, i.e. finding the data with the highest probability to be NN of the query. They have shown that sometimes no object is able to supersede NN, and proposed an $O(n^2)$ time algorithm to find the superseding set. Beskales [19] has considered finding the top k probable

NN, and has presented I/O efficient algorithms to retrieve them and extended algorithms to support the threshold queries. Cheema [20] has formalized the *probabilistic reverse NN*, and has proposed an efficient branch and prune algorithm, and retrieved uncertain data that has a probability more than a given threshold. In the reverse NN problem, the goal is to find all the data points whose NNs are a given query point. Xiang [21] has focused on another important query-based problem, namely, *probabilistic group nearest neighbor* (PGNN) query. The goal is specifically given a set Q of query points; a PGNN query retrieves data objects that minimize the aggregate distance (e.g. sum, min, and max) to Q. He has proposed effective pruning methods to reduce the PGNN search space, and has considered extensive experiments to demonstrate the efficiency of the method.

In this work, we studied NN searching for uncertain query and uncertain data, and proposed efficient algorithms to find NzNNs. In section two, we propose an algorithm for certain data and uncertain query when they are modeled by distinct parallel unit segments. Our algorithm works under Manhattan metric in the $O(\log n + k)$ query time with the $O(n \log n)$ pre-processing time and space, where k is the output size. Wang et al. [13] have shown that if the data is certain and the query has m possible locations, the k-NNs can be reported in the $O(m \log m + (k+m)\log^2 n)$ time using the $O(n \log n \log\log n)$ space. For $m=1$, our algorithm outperforms in both the pre-processing space and the query time. In section three, we propose an $O(\log n + k)$ query time algorithm with $O(n^2\alpha(n^2))$ pre-possessing time and space for uncertain data and uncertain query, where $\alpha(.)$ is the inverse of the Ackermann's function. Our algorithm guarantees the performance in the worst case, and if the query is exact, it uses less space than the PVD method that uses a $\theta(n^3)$ space. In section four, for uncertain data and uncertain query, we propose an approximation algorithm with the maximum error of the query length. Finally, we draw conclusions, and suggest future works in this area.

2. NN searching for certain data and uncertain query

Let $p = \{p_1, ..., p_n\}$ be a set of n points in the plane. For a given uncertain query point q modeled by an axis-aligned parallel segment, the goal is to report NzNN under Manhattan metric. This means that if point p is reported, there exists an instance of the query like q' whose NN is p. A popular method —called the *Minmax method* [18]— reports such a nearest data by computing the minimum distance among all the farthest instances of any data. In other words, p_i is NzNN of a certain query point q if:

$$\forall p_j \in P : d_{min}(p_i, q) \le d_{max}(p_j, q), \qquad (1)$$

where, $d_{\min}(.,.)$ and $d_{\max}(.,.)$ denote, respectively, the minimum and maximum possible distances between two objects. If q is an uncertain query point (e.g. modeled by a segment), the *Minmax method* may report incorrect nearest data because different instances of q can be selected. Figure 2 shows an example of two data points a and b and an uncertain query point q. The bisector of a and b is shown by B_{ab}. It is easy to see that all points above B_{ab} (including all instances of q, especially its end-points q' and q'') are closer to a than to b. Thus b does not have any chance to be NN of q. However, if we use the *Minmax method*, b will be reported as an NzNN. Hence, we define the following definition for reporting NzNN. Point p_i is NzNN of q if and only if

$\exists$ an instance q' of q such that

$$\forall p_j \in P : d(p_i, q') \le d(p_j, q') \qquad (2)$$

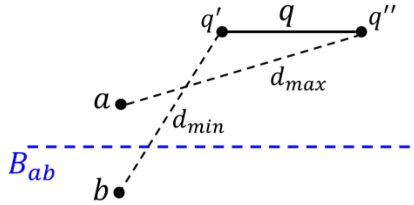

Figure 2. *b* is not NzNN of *q*, although it is reported in case that *Minmax* method is applied.

Therefore, to handle the problem, we construct the Manhattan Voronoi diagram in the pre-processing phase. It consists of four different line slopes (horizontal, vertical, and the lines with slopes $\pi/4$ and $3\pi/4$). A set of lines or segments are said to be *c-oriented* if all of them are parallel to at most c possible orientations. The edges of the Manhattan Voronoi diagram are 4-oriented. We use the following theorem to detect NzNN.

Theorem 1 Point $p_i \in P$ is NzNN of a given query point q if and only if q intersects the Voronoi cell of p_i.

Proof. Suppose that point p_i is NzNN of q. Thus there exists an instance of q like q' such that it is nearest to p_i among all points of P. This includes q' lies in the Voronoi cell of p_i, and consequently, q intersects the Voronoi cell of p_i. Conversely, let q' be an instance of q located in the Voronoi cell of p_i. By the definition of the Voronoi cell, we have:

$d(p_i, q') \le d(p_j, q')$, for $j = 1, \ldots, n., i \ne j$.

Thus based on Eq. (2), point p_i is NzNN of q.

■

Therefore, we can conclude this section by the following theorem.

Theorem 2 Let $P = \{p_1, \ldots, p_n\}$ be a set of n points in the plane, and the query is an uncertain point modeled by an axis-aligned parallel segment. Then NzNN of q can be reported in $O(\log n + k)$ time using the $O(n \log n)$ pre-processing time, where k is the size of the output.

Proof. Based on theorem 1, for finding NzNNs, it is sufficient to find the Voronoi cell(s) containing q. Therefore, using the two end-points of q, we can perform two standard point locations over the Voronoi cells in the $O(\log n)$ time, and report the two cells containing the endpoints —called vc_1 and vc_2. (In the case where both end-points of q lie on the same cell, the problem is easily solved because the Voronoi cells are convex.) In addition, since q is a segment, it intersects the cells between vc_1 and vc_2, and should report all of them. To this end, we have 4-oriented segment intersection searching because edges of the Manhattan Voronoi diagram are 4-oriented, and we can report NzNN in the $O(\log n + k)$ time using the $O(n \log n)$ pre-processing time [22, 23].

3. NN searching for uncertain data and uncertain query

In this section, we consider the case where both the query and the data are uncertain, and we propose an efficient algorithm to find NzNNs. Let $P = \{p_1, \ldots, p_n\}$ be a set of n uncertain points modeled by unit segments parallel to the x-axis (called *x-parallel*) whose projections onto the x-axis do not intersect each other. Using Eq. (2) for an uncertain query point q modeled by a unit x-parallel segment, we first present a method to detect NzNNs.

3.1. Detecting non-zero NNs

Let $\{q_1, q_2\}$ be the end-points of a given uncertain query point q, and $\{e_i, e_i'\}$ be the end-points of $p_i \in P$ for $i = 1, 2, \ldots, n$. The maximum distance between p_i and a point $p \in R^2$ occurs on the end-points, and can be computed using the following equation (see Figure 3).

$$d_{max}(p_i, p) = \max\left(d(e_i, p), d(e_i', p)\right). \quad (3)$$

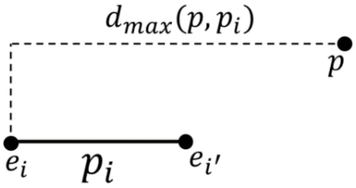

Figure 3. Maximum distance between point *p* and uncertain point p_i.

In order to detect NzNNs, we use the following method. Consider q_1 as a certain query point, and apply the *Minmax method* [18]. Let $M_1 = \{d_1, \ldots, d_n\}$ be a set of maximum distances between q_1 and $p_i \in P$ for $i = 1, \ldots, n$ (e.g. $d_{max}(q_1, p_i)$). Set $m_1 = min_{1 \le i \le n} d_i$, and let $mindata_1$ be some uncertain point in which $d_{max}(mindata_1, q_1) = m_1$.

We assume a diamond (a disk under Manhattan metric) centered at q_1 with radius m_1. We denote such a diamond by Mq_1. Similarly, we assume Mq_2 using $mindata_2$ and q_2 (see Figure 4). From the geometric viewpoint, these diamonds correspond to the *Minmax* method when the query lies on q_1 or q_2 [8].

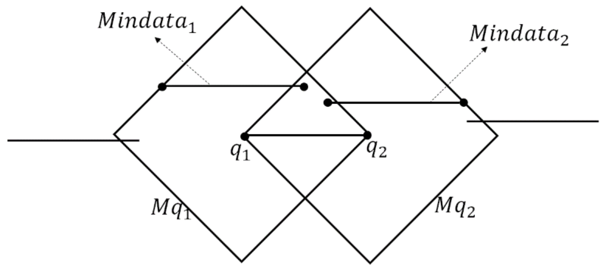

Figure 4. Diamonds Mq_1 and Mq_2.

Observation 1 There is no uncertain point that lies completely in Mq_1 (Mq_2), except $Mindata_1$ ($Mindata_2$).

Indeed, existing of an uncertain point that lies completely in Mq_1 (Mq_2) contradicts with the definition for Mq_1 (Mq_2). The following lemma states that any uncertain point intersecting Mq_1(Mq_2) should be reported as NNzN.

Lemma 1 For an uncertain query point q with end-points $\{q_1, q_2\}$, every uncertain point $p \in P$ that intersects Mq_1 (Mq_2) is a NzNN of q.

Proof. Let e_s be the end-point of p that lies in Mq_1. We show that there is an instance q' of q where $d_{min}(p,q') \le d_{max}(p_i,q)$ for all $p_i \in P, i = 1,\ldots,n$ and $p_i \ne p$. To this end, we construct a placement such that p is a NzNN of q. For any segment (an uncertain point) that intersects Mq_1, we choose the end-point that lies outside Mq_1 for $mindata_1$, we choose the end-point that lies on the boundary of Mq_1 and finally, for p and q, we choose e_s and q_1 as the instances (see Figure 5). By the definition for Mq_1, we have the following equation:

$$d(e_s,q_1) \le d_{max}(p_i,q_1) \quad \text{for } i = 1,\ldots,n \qquad (4)$$

Thus based on *Eq.* (2), it can be concluded that p is a NzNN of q.

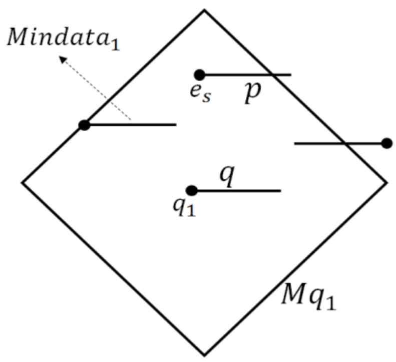

Figure 5. Suitable placement of uncertain points mentioned in proof of lemma 1.

In order to obtain all NNzNs, we should consider all instances of the query that lie between q_1 and q_2. By lemma 1, it is clear that the uncertain points intersecting Mq_1 or Mq_2 are NzNN of q. Furthermore, we need to take into account the points that do not have any intersection with Mq_1 (and Mq_2).

We say that $p_i \in P$ *prunes* $p_j \in P$ with respect to an uncertain query q, if $d_{max}(p_i,q') \le d_{min}(p_j,q')$ for all instances q' of q. We consider all uncertain points that lie outside Mq_1 and Mq_1 that $mindata_1$ and $mindata_2$ cannot prune them, and define the critical regions C to remove such uncertain points (see Figure 6).

$$C = \{p \in R^2 \,|\, d_{min}(p,q') \le d_{max}(mindata_1,q') \;\&\; d_{min}(p,q') \le d_{max}(mindata_2,q') : \forall q' \text{ of } q \qquad (5)$$

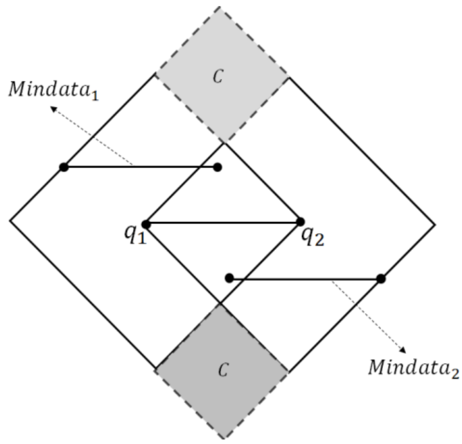

Figure 5. Critical regions with respect to $mindata_1$ and $mindata_2$.

In order to compute the critical regions C, we consider four lines passing through the edges of the diamonds Mq_1 and Mq_2 (see Figure 7). We can construct C by extending the edges for Mq_1 and Mq_2 and finding the intersection points. For example, as shown in figure 7, the critical region that lies above q can be computed by the intersection of lines Lu and Ru of Mq_1 and Mq_2.

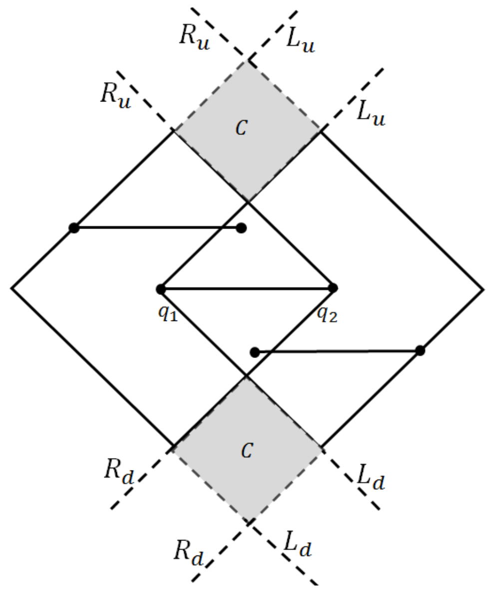

Figure 6. Definition of critical regions *C* by lines passing through edges of a diamond.

Since, in this section, we assume that the segments are unit, Mq_1 and Mq_2 overlap, thus we have two similar critical regions above and below q. Considering the above one, let v_u be the top-most point of the critical region, and v_d be the intersection point of Mq_1 and Mq_2. Assume two horizontal lines h_u and h_d passing through

v_u and v_d, respectively. Figure 8 shows these notations.

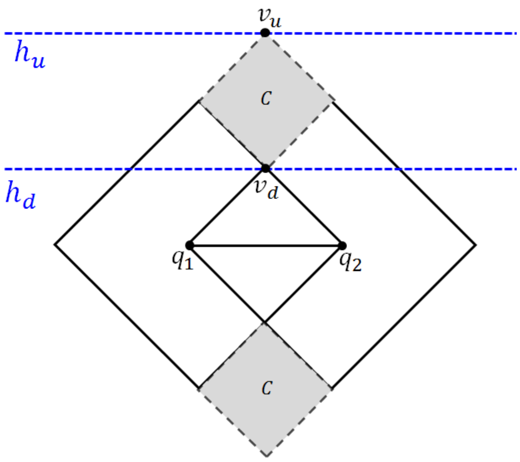

Figure 7. Definitions for h_u and h_d .

Lemma 2 The distance between h_u and h_d (introduced above) is at most L, where L is the query length.

Proof. Without a loss of generality, we assume that Mq_1 is smaller than Mq_2. Let δ be the maximum distance of $mindata_1$ from q_2. We construct a diamond centered at q_2 with radius δ, and denote it by $Maxq_1$. The distance between Mq_1 and $Maxq_1$ is L, which is the sum of the two segments a and b ($a+b=L$) (see Figure 9). The two gray triangles are similar because their angles are equal. It is clear that $e_2 \le e_1$, and by similarity of the triangles, it can be concluded that $c \le a$. Therefore, $b+c \le L$, and the proof is complete.

Corollary 2 If Mq_1 and Mq_2 are disjoint, the distance between h_u and the line passing through q is at most L, where L is the length of q.

Considering figure 10, the proof of the corollary is similar to the proof of lemma 2.

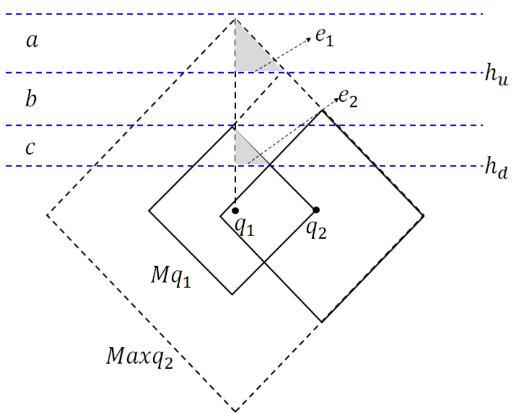

Figure 8. Definition of $Maxq_1$, *a*, *b*, and *c* used in proof of lemma 2.

Theorem 6 Let P be a set of disjoint unit segments as uncertain data. For a given uncertain query point q modeled by an x-parallel segment with length L, the segments intersecting the critical regions (see Eq. (5)) are NzNN of q.

Proof. Suppose that the critical regions above and below q are denoted by c_1 and c_2, and $p_i, p_j \in P$ intersect c_1 and c_2, respectively. Since all the segments are unit, there is no uncertain point that lies completely on c_1 or c_2. Thus we need to show that p_j does not prune p_i. Suppose that both p_i and p_j intersect c_1. We choose the end-point of p_i that lies in c_1 as its instance. Let p be the intersection of the segment perpendicular to q from the instance. We have the following equations under Manhattan metric (see Figure 11).

$$d_{max}(p_j, p) = h_j + v_j,$$

$$d_{min}(p_i, p) = v_i.$$

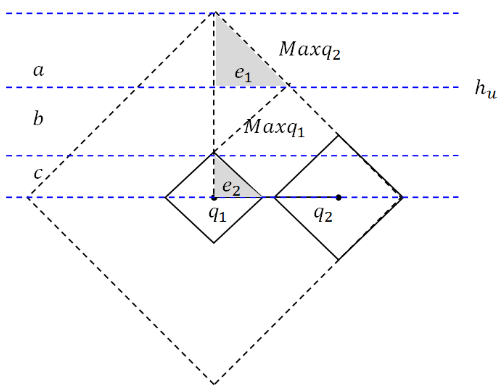

Figure 9. Distance between h_u and query.

Assume, to the contrary, that p_j prunes p_i. Since p_i lies above p_j, $v_j \le v_i$ and $v_i = v_j + v$ for some $v \ge 0$. According to lemma 2, we know that $v \le L$ and $h_j \ge L$ (note that members of p do not overlap). Thus we have:

$$h_j = L + h, \text{ for some } h > 0.$$

If p_i is pruned by p_j, we have the following equation:

$$h_j + v_j < v_i \rightarrow L + h < v,$$

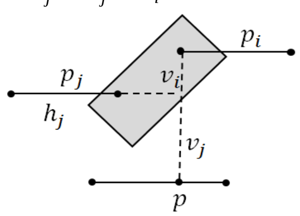

Figure 10 Minimum and maximum distances of p_i and p_j from p.

which is a contradiction to $v \le L$. If p_i intersects c_1, and p_j intersects c_2, we can get symmetry of p_i with respect to *q*, and similarly, prove that p_j cannot prune p_i (see Figure 12).

3.2. Reporting non-zero NNs

By the argument in Section 3.1 we must report all the uncertain points that intersect Mq_1, Mq_2 or the critical regions as NzNN, so we need to compute diamonds Mq_1 and Mq_2 to find the critical regions.

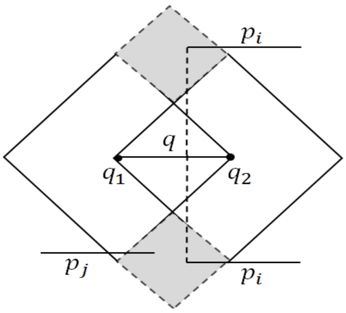

Figure 11. Symmetry of p_i with respect to query point *q*.

It is clear that $m_1 = min_{1 \le i \le n} d_i$ is a lower envelope of $M_1 = \{d_1, \ldots, d_n\}$, where d_i is the maximum distance between q_1 and $p_i \in P$. The projection of d_i onto the *xy*-plane is the farthest point Voronoi diagram of p_i. Therefore, the *xy*-projection of the graph of the function m_1 is a planar sub-division with $O\left(n^2\alpha\left(n^2\right)\right)$ vertices, and it can be computed in the $O(n^2 \log n)$ pre-processing time, where $\alpha(.)$ is the extremely slow growing functional inverse of the Ackermann's function [8,24]. Thus by pre-processing the projection of m_1 (m_2) onto the point location queries, we can perform two standard point locations for q_1 and q_2, and compute Mq_1 and Mq_2 in the $O(\log n)$ time [6]. By theorem 6 and lemma 1, we need to report all the segments that intersect Mq_1, Mq_2 or the critical regions. Since Mq_1, Mq_2, and the critical regions construct a *4-oriented set*, we can report such segments in the $O(\log n + k)$ time using the $O(n \log n)$ space, where *k* is the output size [22,23]. Therefore, we can conclude this section by the following theorem.

Theorem 3. Let $P = \{p_1, \ldots, p_n\}$ be a set of *n* uncertain points modeled by unit *x*-parallel segments that do not intersect each other. Then for any uncertain query modeled by an *x*-parallel segment, we can report NzNNs in the $O(\log n + k)$ time using the $O\left(n^2\alpha\left(n^2\right)\right)$ space, where *k* is the output size.

4. Approximation algorithm for NN searching for uncertain data and uncertain query

Let $P = \{p_1, \ldots, p_n\}$ be a set of *n* uncertain points modeled by arbitrary *x*-parallel segments. Then the segments that intersect the defined critical region *C* in *Eq*. (5) may prune each other. In this case, we report all the segments intersecting Mq_1, Mq_2, and *C*. For such NN reporting, we claim that the maximum error is *L*, where *L* is the query length. This error means that if p_j prunes p_i and we move p_i towards *q* at most *L* (under Manhattan metric), there is no segment that prunes p_i any more.

Lemma 3. An uncertain point $p_i \in P$ intersecting the critical region *C* is a NzNN of query *q* with maximum error *L*, where *L* is the length of *q*.

Proof. Assume that p_i is pruned by some points in *C*. The goal is to show that if we move p_i towards *q* at most *L* (under Manhattan metric), there is no segment that prunes p_i. If Mq_1 and Mq_2 overlap, according to lemma 2, the maximum distance between p_i and Mq_1 (or Mq_2) is *L*, and by moving p_i towards *q* at most *L*, it intersects Mq_1 (or Mq_2), and the goal is achieved. If Mq_1 and Mq_2 are disjoint, by corollary 2, the maximum distance between p_i and *q* is *L*, and by moving p_i towards *q* at most *L*, p_i intersects with *q*, and the proof is complete. The approximation factor *L* for the mentioned approach is tight. When sizes of Mq_1 and Mq_2 are equal, the amount of error is exactly *L*. See figure 12 as such a tight example.

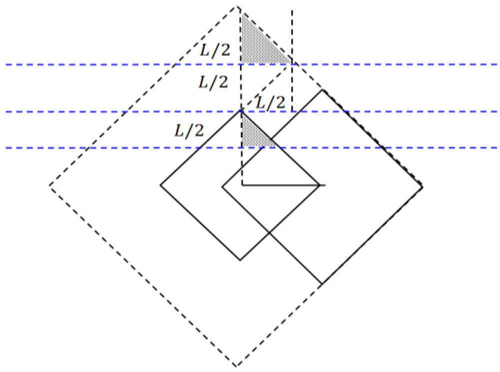

Figure 12. Maximum length for approximation factor.

5. Conclusions and future work

In this paper, we considered the nearest neighbor (NN) searching problem under uncertainty, and proposed algorithms for its variations under Manhattan metric. For the uncertain query and certain data points, we proposed an efficient algorithm that reported non-zero NNs in the $O(n\log n)$ space and $O(\log n+k)$ time, where k is the output size. For the uncertain query and uncertain points modeled by unit segments, we proposed an efficient algorithm that reported non-zero NNs in the $O\left(n^2\alpha\left(n^2\right)\right)$ space and the $O(\log n+k)$ time. For the uncertain query and uncertain points modeled by segments with arbitrarily length, we proposed an approximation algorithm that reported non-zero NNs with the maximum error L in the $O\left(n^2\alpha\left(n^2\right)\right)$ space and the $O(\log n+k)$ time, where L is the query length. As a future work, if the data and query are uncertain and the goal is to report non-zero NNs under Euclidean metric, instead of the constructed diamonds explained in section three, we should compute disks and the critical regions whose boundaries are defined by some algebraic equations. Thus we need to design an efficient algorithm for detecting intersection of geometric objects with algebraic equations.

References

[1] Löffler, M. & Snoeyink, J. (2010). Delaunay triangulation of imprecise points in linear time after preprocessing. Computional. Geometry Theory and Application, vol. 43, no. 3, pp. 234–242.

[2] Ostrovsky-Berman, Y. & Joskowicz, L. (2005). Tolerance envelopes of planar mechanical parts with parametric tolerances. Computer Aided Design, vol. 37, no 5: pp. 531–544.

[3] Khoshdel, V. & Akbarzadeh, A. R (2016). Application of statistical techniques and artificial neural network to estimate force from sEMG signals. Journal of Artificial Intelligence & Data Mining. vol. 4, no. 2, pp. 135-141.

[4] Löffler, M. (2009). Data imprecision in computational geometry, PhD Thesis. Department of computer science. University Utrecht.

[5] Khanban, A. A. (2005). Basic algorithms of computational geometry with imprecise input. PhD Thesis. Department of computing imperial college, University of London.

[6] de Berg, M., Cheong, O., Van Kreveld, M. & Overmars, M. (2008). Computational geometry algorithms and applications, third edition, Berlin Heidelberg: Springer-Verlag.

[7] Sember, J. & Evans, W. (2008). Guaranteed voronoi diagrams of uncertain sites. 20th Canadian Conference on Computational Geometry, Montreal, Canada, 2008.

[8] P.K. Agarwal, et al. (2013). Nearest neighbor searching under uncertainty II. 32th symposium on Principles of database systems. New York, USA, 2013.

[9] Zhang, W. (2012). Nearest neighbor searching under uncertainty, PhD Thesis. Duke University.

[10] Cheng, R., Kalashnikov, D. & Prabhakar, S. (2004). Querying imprecise data in moving object environments. IEEE Transactions on Knowledge and Data Engineering, vol. 16, pp.1112 – 1127.

[11] Zhang, P., Cheng, R., Mamoulis, N., Renz, M., Zufile, A., Tang, Y. & Emrich, T. (2013). Voronoi-based nearest neighbor search for multi-dimensional uncertain databases. 29th International Conference on Data Engineering , Brisbane, Australia, 2013.

[12] Emrich, T., et al. (2010). Boosting spatial pruning: on optimal pruning of MBRs. 40th International Conference on Management of data, Indiana, USA ,2010.

[13] Wang, H. & Zhang, W. (2014). L1 Top-k Nearest Neighbor Searching with Uncertain Queries. Proceedings of the VLDB Endowment, vol. 8, no. 1, pp. 13-24.

[14] Davoodi, M., Modades, A., Sheikhi, F. & Khanteimouri, P. (2015). Data imprecision under **λ**-geometry model. Information Sciences, vol. 295, pp. 126–144.

[15] Davoodi, M. & Mohades, A. (2013). Data Imprecision under λ-geometry: range searching problem. Scientia Iranica, vol. 20 , no.3, pp. 663–669.

[16] Myers, Y. & Joskowicz, L. (2010). Point distance and orthogonal range problems with dependent geometric uncertainties. 14th Symposium on Solid and Physical Modeling, New York, USA, 2010.

[17] Joskowicz, L., Ostrovsky-Berman, Y. & Myers, Y. (2010). Efficient representation and computation of geometric uncertainty: the linear parametric model. Precision Engineering, vol. 34, no. 1, pp. 2–6.

[18] Yuen, S., et al. (2010). Superseding nearest neighbor search on uncertain spatial databases. IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 7, pp.1041-1055.

[19] Beskales, G., Soliman, M. & Ilyas, I. (2008).Efficient search for the top-k probable nearest neighbors in uncertain databases. Proceedings of the VLDB Endowment, vol. 1, no. 1, pp. 326-339.

[20] Cheema, M., et al. (2010). Probabilistic reverse nearest neighbor queries on uncertain data. IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 4, pp. 550-564.

[21] Xiang, L. & Chen, L. (2008). Probabilistic group nearest neighbor queries in uncertain databases. IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 6, pp. 809-824.

[22] Tan, X., Hirata, T. & Inagaki, Y. (1991). The intersection searching problem for c-oriented polygons. Information Processing Letters, vol. 37, no.4, pp. 201–204.

[23] Güting, R. H. (1984). Dynamic c-oriented polygonal intersection searching. Information and control, vol. 63, no. 3, pp.143–163.

[24] Huttenlocher, D. P., Kedem, K., & Sharir, M. (1993). Discrete & Computational Geometry, vol. 9, no.3, pp. 267-291.

نشریه هوش مصنوعی و داده کاوی

نزدیکترین همسایه غیرصفر احتمالی

امیر مصری‌خانی* و منصور داودی منفرد

دانشکده علوم رایانه و فناوری اطلاعات، دانشگاه تحصیلات تکمیلی علوم پایه-زنجان، ایران.

ارسال ۲۰۱۵/۰۴/۲۷؛ پذیرش ۲۰۱۶/۰۹/۱۲

چکیده:

جستجوی نزدیکترین همسایه (NN)، یکی از مسائل مهم در مدیریت داده‌ها است که در داده‌کاوی، تشخیص الگو و هندسه محاسباتی به صورت گسترده مورد مطالعه قرارگرفته است. هدف در مسئله NN، گزارش نزدیکترین داده نسبت به پرس‌وجو داده شده به صورت کارا است. در بسیاری از پژوهش‌های صورت گرفته داده‌ها و پرس‌وجو دقیق فرض شده است، در حالی که در بسیاری از کاربردهای واقعی مانند ردیابی، مکان‌یابی، GIS و داده‌کاوی ممکن است داده‌ها و پرس‌وجو نادقیق باشند. بنابراین در چنین موقعیتی یکی از راه‌حل‌ها، گزارش داده‌ای-نزدیکترین همسایه غیرصفر- است که با احتمال بزرگتر از صفر، نزدیکترین همسایه پرس‌وجو داده شده باشد. فرض کنید P شامل n نقطه غیرقطعی باشد که به صورت نواحی هندسی مدل شده‌اند. در این مقاله ما حالت‌های مختلفی از مسئله را بررسی کرده‌ایم. اگر داده‌ها دقیق و پرس‌وجو نادقیق باشد که به صورت پاره‌خط‌های موازی محورها مدل شده باشد، ما الگوریتمی کارایی ارائه کرده‌ایم که با زمان پیش‌پردازش $\mathrm{O}(\mathrm{n}\log\mathrm{n})$، در زمان $\mathrm{O}(\log\mathrm{n}+\mathrm{k})$ نزدیکترین همسایه‌های غیرصفر را گزارش می‌کند که k اندازه خروجی است. اگر داده‌ها و پرس‌وجو نادقیق باشند که به صورت پاره‌خط‌های واحد موازی محور x ها مدل شده‌اند، الگوریتمی کارایی ارائه شده است که تحت متر منهتن و زمان پیش‌پردازش $O(n^2\alpha(n^2))$، در زمان $O(\log n+k)$ نزدکترین همسایه غیرصفر را گزارش می‌کند که $\alpha(.)$ تابع معکوس آکرمن است و سرعت رشد فوق‌العاده پایینی دارد. در نهایت برای پاره‌خط‌های با اندازه دلخواه موازی محور x ها، ما الگوریتمی تقریبی ارائه کرده‌ایم که نزدیکترین همسایه غیرصفر را با زمان پیش‌پردازش $O(n^2\alpha(n^2))$ و در زمان $O(log\, n+k)$ نزدیکترین همسایه‌های غیرصفر را با خطای L گزارش می‌کند که L اندازه پرس‌وجو است.

کلمات کلیدی: جستجوی نزدیکترین همسایه، عدم‌قطعیت، عدم دقت، احتمال غیرصفر.