Flexible Location-Based Spatial Queries

Gloria Bordogna[1], Marco Pagani[1], Gabriella Pasi[2], and Giuseppe Psaila[3]

[1] CNR-IDPA unità Dalmine, via Pasubio 5, 24044 Dalmine (BG) Italy
gloria.bordogna@idpa.cnr.it,marco.pagani@idpa.cnr.it
[2] Univ. degli Studi di Milano Bicocca, Disco, via Bicocca degli Arcimboldi 8, 20126 Milano, Italy
gabriella.pasi@itc.cnr.it
[3] Univ. degli Studi di Bergamo, Facoltà di Ingegneria, via Marconi 5, 201446 Dalmine (BG) Italy
psaila @unibg.it

Abstract. A model for representing and evaluating flexible Location-Based Spatial Queries (LBSQ) is proposed. In a LBSQ the selection condition is generally a constraint on the distance of the objects in the database (instances) from the user location. Such queries are becoming more and more useful in location-based services such as those provided by cell-phones, Wireless LAN and GPS technologies. However their usefulness is limited by the inability of current systems to represent and manage the imprecision often characterizing the knowledge of the user's and instances' locations. In this contribution we propose a fuzzy model of flexible LBSQs in which either the user location, or the instances locations or the selection condition itself or even all of them are imprecise. To define a unifying approach in all cases of imprecision we generalize the notion of the Minkowski sum within fuzzy sets and apply it to combine the (imprecise) user location with the (soft) query condition. This way we derive the actual soft constraint with respect to the user's location. The instances relevant to the query are those whose locations are included in the actual soft constraint representation to some extent.

Keywords: location-based queries, imprecise locations, Minkowski sum.

1 Introduction

During the last years the outstanding growing market for positioning technologies such as Global Positioning System, Radio Frequency Identification Systems, and Global Systems for Mobile communications has raised the research interest for more efficient and effective location-based services [16,21].

Among location-based services, the evaluation of Location-Based Spatial Queries (LBSQs) that retrieves information based on the current locations of users is a crucial task, due to the mobility of users and, in some applications, also of the objects in the database (instances) [8,18]. For example, an airplane pilot may ask for "the nearest airplane crossing his route". In order to avoid possible collisions, the answer to this query should not depend on the location of the airplanes when the pilot issued the request but on the locations where the pilot receives the query results, which are generally not precisely known at the time the query is issued. In this example, both the user and the instances in the database (the other airplanes) are moving, thus their

O. Castillo et al. (Eds.): Theor. Adv. and Appl. of Fuzzy Logic, ASC 42, pp. 36–45, 2007.
springerlink.com

locations are known with imprecision. The need to take into account the location of the user and/or of the instances at the time the query results are received by the user is necessary in many applications of robotics. Generally this is useful whenever the speed of the user and/or of the instances is so high to determine a situation that makes the query results sensible with respect to the current positions of the considered items at the time the user receives the query results.

These are not the only cases of imprecision in LBSQs. Imprecision on the user location can derive by several causes such as measurement errors or limited resolution of the device used to detect the location coordinates, or insufficient network speed. In some cases imprecision can be introduced on purpose to mask the exact user location for preserving the privacy of the user [1,13]. Finally, LBSQs can involve imprecision also in the condition specification such as in the query "find the taxi cabs that are very close". Generally the uncertainty in location data has been modeled by means of probability distributions on the spatial domain [18], and the research in this respect mainly focused on efficiency issues [8, 9, 18, 19].

In our proposal, we model in the unifying framework of fuzzy set theory the representation and evaluation mechanism of flexible LBSQs by taking ideas from the approaches to flexible querying in fuzzy databases [5,7,9,11,17,20]. In this contribution by flexible LBSQs we intend soft range queries against possibly ill-defined location information admitting degrees of satisfaction. A range query specifies a selection condition that consists in a bounding box centered at the user location. With the terms "soft range queries" we mean LBSQs specifying a vague range condition. This is expressed by a linguistic term such as *close* defined as a soft constraint on the spatial domain [3]. Imprecision may affect user location, instances location and the range condition alone or in any combination one another. Imprecision on location data is represented by means of possibility distributions [5]. The soft constraint specifying the vague range condition is defined with a membership function that decreases with the distance from the coordinates' origin. The notion of Minkowski sum is generalized to fuzzy sets and is used to generate the actual soft constraint with respect to the possibly imprecise user location. Finally, the degree of satisfaction of a soft range query is computed as the fuzzy inclusion degree of the instances' locations in the fuzzy set representing the actual soft constraint [4,7]. In the next section an overview of the literature on LBSQs is briefly introduced. In section 3 the formalization of Flexible LBSQ evaluation is defined. Finally, in the conclusion the main results are summarized.

2 Related Works

The research on LBSQs focused mainly on efficiency issues such as the investigation of new ways of indexing and caching spatial data to support the processing of LBSQs including point query, window query, nearest neighbor (NN) search, k nearest neighbor search [14,23,24,25]. Another issue was the management of LBSQs in a distributed way so as to achieve efficiency while allowing complex queries based on the use of location-dependent operators [12,15]. In this respect, another direction is the study of LBSQs involving some uncertainty in location data. This is also the focus or our proposal. Uncertainty on the user location can derive by several causes as outlined in the introduction, or can be introduced on purpose to mask the exact location for preserving the privacy of the user [1,13]. In this respect the amount of uncertainty required to meet both privacy and the requirement on the service quality has been studied [8]. Generally the location uncertainty has been modeled by means of probability distributions on the spatial domain, basically bi-dimensional Gaussian functions or uniform distributions within a window or neighborhood [18]. The research on this topic faced the evaluation of probabilistic queries, such as probabilistic range queries [22]. Probabilistic queries evaluate uncertain location information and provide plausible answers in the form of probabilities. Imprecise LBSQs have also been studied to model imprecision affecting the instances location, such as in the case of moving objects [8, 18,19]. To our knowledge no one has yet considered that both the location of the user and of the objects in the spatial database and the selection condition itself can be imprecise such as in the query "find the closest airplanes to my path ".

In our proposal we consider these three situations alone and in any combinations one another, and provide a modeling within fuzzy set theory, by taking ideas from flexible querying in fuzzy databases [5,7,11,17,20]. Imprecision on location data is represented by means of possibility distributions, which are easier to define than probability distributions since they do not need to satisfy the normalization constraint. The vague selection condition, that is a vague range condition, is represented by a soft constraint defined with respect to the coordinates' origin. The generation of the soft constraint with respect to the user's location is dynamically determined by computing the fuzzy Minkowski sum [2], that is defined in this contribution to this purpose. This way, we generate a new soft constraint that the instances' locations, possibly imprecise, must satisfy to some extent in order to be retrieved. Similarly to what happens in fuzzy databases, a matching function between fuzzy sets is defined to compute the degrees of satisfaction of the instances, which in our context is a fuzzy inclusion function between spatial distributions [4,7].

3 Flexible Location-Based Spatial Queries

In this section we classify the kinds of flexible LBSQs considered in this contribution and introduce their formal representations and evaluation functions. In Table 1 the types of LBSQs are characterized, based on the imprecision affecting their information units, i.e., the user location, the instances locations, and the selection condition. For simplicity we restrict our analysis to soft range queries and define a mechanism to compute degrees of satisfaction of the instances. We consider the linguistic expression *close* as example of specification of the vague range condition. Notice that *close* could be replaced by any other linguistic term defining a soft constraint on the distance from a position such as "*very close*", "*not too far*", etc.; thus, the vague range condition can be regarded as the specification of a nearest neighbor search condition.

3.1 Type 1: Crisp Query

This is the usual crisp range query of the kind "find instances located at a maximum distance [$\pm\Delta x$, $\pm\Delta y$] from my location". In this LBSQ both the location data and the range condition are precisely represented by their coordinates (x,y) on the spatial domain and the bounding box $[\pm\Delta x, \pm\Delta y]$ centered at the user location (x_u,y_u). Only the instances whose coordinates (x_i,y_i) fall within the limits $x_u\pm\Delta x$ and $y_u\pm\Delta y$ completely satisfy the range condition:

$$\text{Select } i \mid \{(x_i,y_i)\} \subseteq \text{box}((x_u-\Delta x,\ y_u-\Delta y),(x_u+\Delta x,\ y_u+\Delta y))$$

Table 1. Types of flexible LBSQ

Query type	User location	Instances location	Query range condition
1	(x_u,y_u)	(x_i,y_i)	$[\pm\Delta x, \pm\Delta y]$
2	*Around*(x_u,y_u)	(x_i,y_i)	$[\pm\Delta x, \pm\Delta y]$
3	(x_u,y_u)	*Around*(x_i,y_i)	$[\pm\Delta x, \pm\Delta y]$
4	*Around*(x_u,y_u)	*Around*(x_i,y_i)	$[\pm\Delta x, \pm\Delta y]$
5	(x_u,y_u)	(x_i,y_i)	*close*
6	*Around*(x_u,y_u)	(x_i,y_i)	*close*
7	(x_u,y_u)	*Around*(x_i,y_i)	*close*
8	*Around*(x_u,y_u)	*Around*(x_i,y_i)	*close*

3.2 Type 2: Crisp Query with Imprecise User Location

In this type of LBSQ, the user location is affected by imprecision *Around*(x_u,y_u) while the instances' locations and the selection conditions are precise. This is for example the case of a moving robot whose location varies in time looking for some stable resources close to his current location, that is around (x_u,y_u). Evaluating this kind of queries implies having a representation of the imprecise user location. We represent an imprecise user location *Around*(x_u,y_u) by means of a possibility distribution π_u: $X\times Y \rightarrow [0,1]$ on the bi-dimensional spatial domain. The form of π_u depends on the specific application.

Examples of definition of π_u

In the case of imprecision introduced on purpose, π_u can be defined as a uniform distribution within a box or a circle centered at (x_u,y_u), i.e., $\pi_u(x,y)=u\in[0,1]\ \forall x,y$ with $|(x,y)-(x_u,y_u)| < r$, $\pi_u(x,y)=0$ otherwise. In the case of a moving user such as a robot, π_u can be built based on a data driven approach by monitoring the robot, and by determining its speed and travel direction. Given two subsequent locations of the robot, i.e., the user, (x_0,y_0) and (x_1,y_1) at time t_0 and t_1 respectively, we can build π_u by considering the robot's speed and most possible position (x_t,y_t) at the answer time t:

$$\pi_u(x,y)=\begin{cases} e^{-\left[\frac{1}{2}k((x-x_t)^2+(y-y_t)^2)\right]} & for \quad |(x,y)-(x_t,y_t)|<|(x_1,y_1)-(x_t,y_t)| \\ & \end{cases} \quad with \quad \begin{cases} x_t=x_1+\dfrac{x_1-x_0}{t_1-t_0}(t-t_1) \\ y_t=y_1+\dfrac{y_1-y_0}{t_1-t_0}(t-t_1) \end{cases} \qquad (1)$$

In this case π_u decreases smoothly with the distance from the most possible location (x_t,y_t) and becomes null outside the circle of radius $r=\sqrt{(x_1,y_1)^2-(x_t,y_t)^2}$.

-°-

Based on the user's location representation π_u and the precise range condition, we derive a representation of the region delimiting candidate instances that satisfy the query to some extent. This is achieved by computing the Minkowski sum ⊕ of the range condition and the at least ∝-possible locations of the user with ∝>0. We represent the range condition Z=[±Δx, ±Δy] as a box centered in (0,0) and displacements ±Δx and ±Δy on the x and y axis respectively:

$$S=\textit{At least } \propto\textit{-possible}(\pi_u(x,y))=\{(x,y) \mid (\pi_u(x,y)>\propto)\}$$

The Minkowski sum ⊕ of the two polygons S and Z on the Euclidean spatial domain is defined as [2]:

$$S \oplus Z = \{ s + z \mid s \in S \text{ and } z \in Z \} \tag{2}$$

in which *s* and *z* are points on the spatial domain. The Minkowski sum is defined as the union of all the translations of Z by a point *s* located in S. For **type 2** queries the Minkowski sum can be interpreted as the union of all the range queries by considering all possible positions of the user who is located somewhere inside S. Clearly, only the instances whose location (x_i,y_i) is within the region S⊕Z satisfy the query. Then, the evaluation of this range query with user location imprecision corresponds to select the instances that satisfy the topological relationship inclusion (x_i,y_i) in S⊕Z, i.e.:

$$\text{select } i \;\Big|\; \{(x_i,y_i)\} \subseteq (\textit{At least 0-possible}(\pi_u \oplus [\pm\Delta x, \pm\Delta y])$$

Note that we can generalize the evaluation of the **type 2** query with imprecise user location considering any desired ∝-possible location of the user with 0<∝≤1. To compute degrees of satisfaction for the instances we define the generalized fuzzy Minkowski sum $\oplus_F$ that combines two fuzzy sets and determines a fuzzy set as a result.

Definition of the generalized Fuzzy Minkowski sum

Given two fuzzy sets S and Z defined on a spatial domain X, the generalized Fuzzy Minkowski sum $S\oplus_F Z$ is defined as the fuzzy union (max) of all the translations of Z by every element *s* belonging to some extent to the fuzzy set S:

$$S \oplus_F Z = \{ \mu_{S\oplus_F Z}(r) / r \mid r = s + z \text{ and } s \in S,\ z \in Z\} \tag{3}$$

where $\mu_{S\oplus_F Z}(r)=\max_{\forall s\in S, \forall z\in Z \mid s+z=r}(\min(\mu_S(s),\max(\mu_S(s+z),\mu_Z(z))))$

Example: let us consider a simple example in a one-dimensional spatial domain:

S:={0.5/2, 1./3, 1./4, 0.7/5} Z:={0.2/-2, 0.8/-1, 1./0, 0.8/1, 0.2/2}

R=S $\oplus_F$ Z={0.2/0, 0.5/1, 0.8/2, 1./3, 1./4, 0.8/5, 0.7/6, 0.2/7}

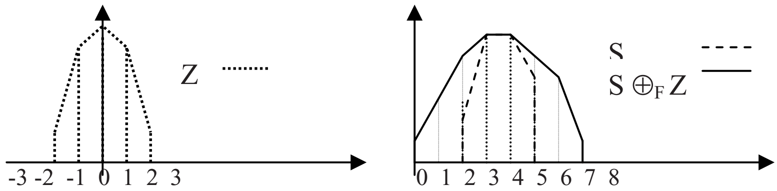

Fig. 1. Representation of the Fuzzy Minkowski sum in a one-dimensional domain

It can be proved that the fuzzy set R representing the result of Generalized Fuzzy Minkowski sum $R=S\oplus_F Z$ includes S, i.e., $\mu_{S\oplus_F Z}(x) \geq \mu_S(x)\ \forall x\in X$, and that it reduces to the crisp Minkowski sum in the case in which both S and Z are classic sets:

$$\mu_{S\oplus_F Z}(x)=1\ \forall x\in S\oplus_F Z \text{ and } \mu_{S\oplus_F Z}(x)=0 \text{ otherwise.}$$

-°-

In the context of the evaluation of **type 2** queries with imprecise user location $S=\pi_u$ and crisp range condition $Z=[\pm\Delta x, \pm\Delta y]$, we assume that $\mu_Z(z)=1$ for $-\Delta x\leq x_z\leq\Delta x$ and $-\Delta y\leq y_z\leq\Delta y$, while $\mu_Z(z)=0$ otherwise. In this case the generalized fuzzy Minkowski sum $\pi_u\oplus_F[\pm\Delta x,\pm\Delta y]$ identifies the union of all the translations of the range Z by any point $s=(x_s,y_s)$ belonging to some extent to the possible user location π_u. It can be interpreted as the fuzzy union of all the range queries by considering all possible positions of the user who is located somewhere inside S. The instances whose location (x_i,y_i) is within the support of the fuzzy set $\pi_u\oplus_F[\pm\Delta x, \pm\Delta y]$ satisfy the query. For these instances, we can compute a satisfaction degree *degree(i)* for ranking the instances to the range query based on the evaluation of the fuzzy inclusion of their precise location (x_i,y_i) in $\pi_u\oplus_F(\pm\Delta x, \pm\Delta y)$ as follows:

$$degree(i)= \text{degree}\ (\{(x_i,y_i)\}\subseteq_F (\pi_u\oplus_F[\pm\Delta x, \pm\Delta y]))= \mu_{\pi_u\oplus_F[\pm\Delta x, \pm\Delta y]}(x_i,y_i) \qquad (4)$$

3.3 Type 3: Crisp Range Query with Imprecise Instances' Locations

This type of LBSQ is dual with respect to the previous one. In this case, the instances' locations are imprecisely known while the user position is precise. For example, this is the case of moving objects, such as taxicabs with the user being located at a taxi station. To evaluate this kind of queries, first we represent the instances locations by means of possibility distributions on the spatial domain $Around(x_i,y_i)= \pi_i$. As in the previous case, we can adopt a data driven approach to generate π_i by exploiting collected information on previous positions of the instances.

To evaluate this kind of queries, we can adopt two alternative procedures.

Procedure A

With this procedure, we first build the crisp Minkowski sum $(x_u,y_u)\oplus[\pm\Delta x, \pm\Delta y]$ of the precise user location with the range condition. Then, we derive the fuzzy set $(x_u,y_u)\oplus_F[\pm\Delta x, \pm\Delta y]$ from the crisp Minkowski sum as follows:

$$\mu_{(x_u,y_u)\oplus_F[\pm\Delta x, \pm\Delta y]}(r)= 1\ \ \forall r\in (x_u,y_u)\oplus[\pm\Delta x,\pm\Delta y],\ \mu_{(x_u,y_u)\oplus_F[\pm\Delta x,\pm\Delta y]}(r)=0 \text{ otherwise}$$

Finally, to compute the degrees of satisfaction of the LBSQ by the N instances, for each instance i, we evaluate the degree of fuzzy inclusion of its imprecise location π_i in the fuzzy set $(x_u,y_u)\oplus_F[\pm\Delta x, \pm\Delta y]$ as follows:

$$degree(i) = \text{degree}\,(\pi_i \subseteq_F ((x_u, y_u)\oplus_F[\pm\Delta x, \pm\Delta y]))$$

Definition of fuzzy inclusion degree between fuzzy sets
The fuzzy inclusion between two fuzzy sets A and B on a spatial domain X can be defined based on the cardinality $\int$ of the fuzzy sets and the proportion of A included in B [3]:

$$degree(i) = \text{degree}\,(A \subseteq_F B) = \frac{\int_X (A \cap_F B)dx}{\int_A A} = \frac{\int_X min(\mu_A(x), \mu_B(x))dx}{\int_X \mu_A(x)dx} \tag{5}$$

where $\cap_F$ is the intersection of fuzzy sets and $\int \mu_A(x)dx$ is the integral (for X continuous) or sum (for X discrete) of the membership values of the fuzzy set A. Notice that formula (5) reduces to formula (4) in the particular case in which the fuzzy set A is a single point of the spatial domain as in the case of **type 2** queries.

-°-

In the case of **type 3** queries, formula (5) reduces to compute the proportion of π_i included in the support of the crisp Minkowski sum $support((x_u,y_u)\oplus_F[\pm\Delta x, \pm\Delta y])$:

$$degree(i) = \text{degree}(\pi_i \subseteq_F ((x_u,y_u)\oplus_F[\pm\Delta x,\pm\Delta y])) = \frac{\int_{support((x_u,y_u)\oplus_F[\pm\Delta x,\pm\Delta y]}\pi_i(x),dx}{\int_X \pi_i(x)dx} \tag{6}$$

This procedure computes just once the crisp Minkowski sum based on (2) and evaluates N fuzzy inclusion degrees between fuzzy sets by applying formula (6).

Procedure B
By adopting this procedure, the evaluation of the LBSQ of **type 3** corresponds to evaluate N range queries of **type 2**, in which we exchange the user location with the instances location. We first build the N fuzzy Minkowski sums $(\pi_i\oplus_F[\pm\Delta x,\pm\Delta y])$, with $i=1,..,N$ of the imprecise instance location π_i with the crisp range condition $[\pm\Delta x, \pm\Delta y]$. Then, if the precise user location falls within each N region, we retrieve the corresponding instance. Also in this case, for each instance we can compute a satisfaction degree *degree(i)* of the query based on the evaluation of the fuzzy inclusion of the precise user location (x_u,y_u) in the fuzzy Minkowski sum $(\pi_i\oplus_F[\pm\Delta x, \pm\Delta y])$ by means of formula (4):

$$degree(i) = \text{degree}\,(\{(x_u, y_u)\} \subseteq_F (\pi_i \oplus_F[\pm\Delta x, \pm\Delta y])) = \mu_{\pi_i \oplus_F [\pm\Delta x, \pm\Delta y]}(x_u,y_u)$$

With respect to ***procedure A***, in this case the computation of the fuzzy inclusion degree is much simpler than (6) since it reduces to formula (4), but we have the increased cost of computing N fuzzy Minkowski sums instead of just a crisp one. The decision on which procedure to adopt depends on efficiency reasons, i.e. the costs of computation of N fuzzy Minkowski sums (based on def. (3)) versus the cost of a crisp Minkowski sum plus N fuzzy inclusions between fuzzy sets (6).

3.4 Type 4: Crisp Range Query with Both Imprecise User's and Instances' Locations

This situation is the one in which both the user and instances locations are imprecise, such as in the airplane example of the introduction. In this case we adopt ***procedure A*** described for **type 3** queries. We have the increased complexity of computing a fuzzy Minkowski sum $\pi_u \oplus_F [\pm\Delta x, \pm\Delta y]$ of the imprecise user location, and the crisp range condition based on definition (3) instead of a simpler crisp Minkowski sum. In constrast, ***procedure B*** is much more inefficient than ***procedure A***. In fact, since we have N imprecise instances' locations π_i, by adopting ***procedure B*** we would have to compute N fuzzy Minkowski sums $\pi_i \oplus_F [\pm\Delta x, \pm\Delta y]$. Further, being also the user location imprecise, π_u, we would have also to evaluate N fuzzy inclusions of the fuzzy set $A=\pi_u$ in the fuzzy sets $B=\pi_i \oplus_F (\pm\Delta x, \pm\Delta y)$, with i=1,..$N$, by applying formula (5) so as to derive the N degrees to rank the instances:

$$degree(i) = \pi_u \subseteq_F (\pi_i \oplus_F [\pm\Delta x, \pm\Delta y]).$$

3.5 Type 5, 6, 7 and 8: Soft Range Queries with Possible Imprecise Location

All these types of LBSQs specify a vague range condition by means of a linguistic predicate such as *close*. In the context of fuzzy databases, vague conditions are defined as soft constraints on the domains of attributes [4,5,6,17,20]. We retain this representation and define a vague range condition like *close* as a soft constraint on the spatial domain $X \times Y$ with the membership function μ_{close} that decreases with the distance from the coordinate origin (0,0) (see Figure 2): $\mu_{close}(x,y) \rightarrow [0,1]$ $\forall x,y \in X \times Y$. The shape of μ_{close} can be either a box with vague boundaries or a symmetric function decreasing with the distance from the origin. This way the vague range condition is defined in an absolute way. It is during the evaluation of the query that the representation of the actual soft constraint is generated with respect to either the users' or the instances locations.

To evaluate this kind of queries according to definition (3), we first compute the fuzzy Minkowski sum of the user location $location_u$, that can be either precise (for **type 5** and **7** queries) or imprecise π_u (**type 6** and **8** queries), and the fuzzy set representing the soft range condition *close*:

$$close_u := location_u \oplus_F \mu_{close}.$$

This way we generate $close_u$ that represents the actual soft range condition defined with respect to the user location. $close_u$ must be satisfied to some extent by the location of the *i*-th instance, $location_i$, in order to retrieve the instance.

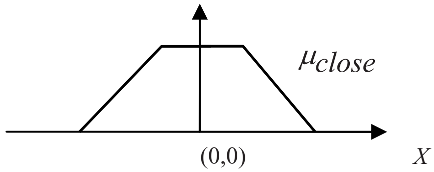

Fig. 2. Vague range condition *close* in a one dimensional doman *X*

The second step computes the degrees of the fuzzy inclusion of each instance location $location_i$ in the fuzzy set $close_u$. This step has a distinct complexity if we are evaluating **type 5 - 6** queries with respect to **type 7 – 8** queries. For **type 5 - 6** queries, since the instances' locations are precise (x_i,y_i), we just take, as satisfaction degree of an instance *i*, the value computed by formula (4):

$$degree(i) = \mu_{location_u \oplus \mu_{close}}(x_i, y_i)$$

In the case in which $location_i = \pi_i$ (imprecise instance location) as in **type 7 - 8** queries, we have to compute the fuzzy inclusion degree of the fuzzy sets $A=\pi_i$ into the fuzzy set $B=close_u$ for each instance *i* by applying definition (6):

$$degree(i) = \text{degree}(\pi_i \subseteq_F (location_u \oplus_F \mu_{close}))$$

4 Conclusions

Providing effective and efficient mechanisms to support LBSQs affected by imprecision is useful in many application fields. In this contribution, a model for evaluating flexible LBSQs with imprecise locations and vague selection condition is proposed. The model is based on the fuzzy generalization of the Minkowski sum between crisp sets. The fuzzy Minkowski sum is defined on two fuzzy subsets of the spatial domain, so as to produce a fuzzy set as a result. We also apply the notion of fuzzy inclusion of fuzzy sets to compute a degree of satisfaction of a LBSQ in the case of imprecise instances' locations. As far as we know, there is not up to date a proposal in this respect within the fuzzy context. The proposals based on probability distributions faced just some cases in which imprecision affects either user location, or instance locations but never both of them at the same time with vague range conditions. Our proposal has the advantage with respect to the probabilistic approach of formalizing all situations of imprecision in LBSQs evaluation within a unifying framework.

References

1. Beresford A., Stajano F.: Location Privacy in Pervasive Computing. IEEE Pervasive Computing, 2(1). (2003) 46-55.
2. Berg M., Kreveld M., Overmars M., Schwarzkopf O.: Computational geometry, Algorthms and Applications, Springer Verlag, (2000).

3. Bordogna G., Pagani M., Psaila, G.: SQL spatial with Customisable soft selection conditions, in “Flexible Databases supporting Imprecision and Uncertainty”, series Studies in fuzziness and soft computing, Springer Verlag,, 203, (2006).
4. Bordogna G., Pasi G.: A semantics for Soft selection conditions in Fuzzy Databases based on a Fuzzy Inclusion, in Proc. of IFSA03 (2003).
5. Bosc P., and Prade H.: An Introduction to the Fuzzy Set and Possibility Theory-based Treatment of Flexible Queries and Uncertain and Imprecise Databases, in Uncertainty Management in Information Systems: from Needs to Solutions, A. Motro and P. Smets eds, Kluwer, (1997), 285-324.
6. Bosc P., Buckles B.B., Petry F.E., Pivert O.: Fuzzy Databases, in Fuzzy Sets in Approximate Reasoning and Information Systems, J. C. Bezdek, D. Dubois and H. Prade eds., Kluwer Ac. Pub., (1999), 403-468.
7. Bosc P., Pivert O.: A Formal Framework for Representation-Based Querying of Databases Containing Ill-known Data, in Proc. of the IIA'99 SOCO'99, (1999).
8. Cheng R., Kalashnikov D., Prabhakar S.: Evaluating probabilistic queries on imprecise data, in proc. of ACM SIGMOD, (2003).
9. Cheng R., Xia Y., Prabhakar S., Shah R., Vitter J.S.: Efficient indexing methods for probabilistic threshold queries over uncertain data. In Proc. of VLDB, (2004).
10. De Caluwe R.: Fuzzy and Uncertain Object-Oriented databases, Concepts and Models, World Scientific, (1997).
11. Dubois D., Prade H.: Tolerant Fuzzy Pattern matching: an Introduction, in Fuzziness in Database Management Systems, P. Bosc and J. Kacprzyk eds., Physica Verlag, (1994).
12. Gedik B., Liu L.: Mobiyes Distributed processing of continuously moving queries on moving objects in a mobile system, in Proc. of EDBT, (2004).
13. Gruteser M. , Grunwald D.: Anonymous Usage of Location-based Services through Spatial and Temporal Cloaking. In Proc. of the 1st Int. Conf. on Mobile Systems, Applications and Services, (2003).
14. Hu H., Xu J., Wong W., Zheng B., Lee D., Lee L.: Proactive Caching for Spatial Queries in Mobile Environments, in Proc. of the 21st ICDE (2005).
15. Iliarri S., Mena E., Illarramendi A.: Location-dependent queries in mobile context: distributed processing using mobile agents. ACM TMC, 5(8) 2006.
16. Lee D., Xu J., and Zheng B.: Data management in location-dependent information services, IEEE Pervasive Computing, 1(3), (2002).
17. Petry F.E., Fuzzy Databases, Kluwer, (1996).
18. Pforser D., Jensen C.: Capturing the uncertainty of moving-objects representations. in Proc. of SSDBM, (1999).
19. Tao Y., Papadias D., Shen Q.: Continuous NN search. In Proc. of VLDB, (2002).
20. Vila M.A., Cubero J.C., Medina J.M., Pons O.: A Conceptual Approach for Dealing with Imprecision and Uncertainty in Object-Based data Models, Int. J. of Intel. Syst., 11(10), (1996), 791-806.
21. Warrior J., McHenry E., McGee K.: They know where you are, Spectrum, 40(7), (2003) 20-25.
22. Wolfson O., Sistla A., Chamberlain S., Yesha Y.: Updating and querying databases that track mobile units. Distributed and Parallel Databases, 7(3), (1999) 257-387.
23. Xu J., Zheng B., Lee W., Lee D., Energy Efficient Index for Querying Location_Dependent Data in Mobile Broadcast environment, in proc. of ICDE, (2003).
24. Xu J., Lee W., The D-Tree: An Index Structure for Planar Point Queries in Location-Based Wireless Services, IEEE Trans. on Knowledge and data Eng., 16(12), (2004), 1-16.
25. Zheng B., Xu J., Lee W., and Lee D., Energy-Conserving Air Indexes for NN Search, E. Bertino et al. Eds.: EDBT 2004, LNCS 2992, Springer-Verlag, (2004), 48–66.