ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

Efficient Computation of Range Aggregate Against Uncertain Location-Based Queries

Ahmed-ullah-khan[1], **Samra Noorin**[2]

Associate Manger, Sterlite Technology Limited[1]

M.Tech, Student, Jyothismathi Inst., of Science and Technology[2]

Abstract: Location based services have become increasingly important for many real time applications. In such applications queries are executed in a multi-dimensional space. In this paper we propose an algorithm to compute range aggregates such as count, average and sum that facilitate answering uncertain location based queries. We also developed a prototype application to test the efficiency of the proposed technique. The empirical results revealed that the application works with efficient computation of range aggregates and can be used in real world applications where location based services are required.

Index Terms: Location based services, range aggregates, uncertain location based queries, index.

INTRODUCTION

In many location based applications, there is a problem of uncertainty and query imprecision. To process location based queries in such environment is a challenging job. Many existing applications to solve such problem are not efficient due to inapplicability of certain query points and data points. There are many examples that motivate this kind of work. Two such examples are illustrated here. The first example is related to military where blast overheads are used to destroy enemies. When such overhead is executed, there are civilian points along with target point. When the distance and the location of the query is uncertain, we need an algorithm that can efficiently compute the range aggregates in order to ensure that the civilian points are not destroyed while damage is made to enemies/target. Thus it is essential to avoid damage of civilian causalities. Fig.1 illustrates this scenario well.

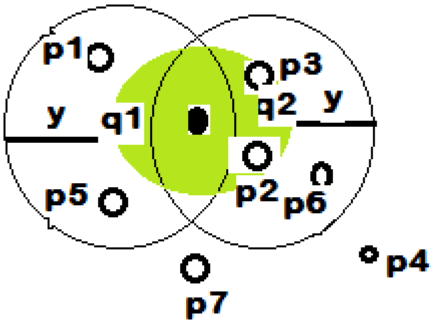

Fig. 1 – Motivating Example

As can be seen in fig. 1, there are query points q1 and q2. The points p1 to p6 are considered civilian places. Based on the range of missile it is important to compute the range aggregates in order to save civilian places. When missile is executed on q1 setting as middle, some civilian places get destroyed. In the same fashion, when missile is executed on q2 setting as middle, some other civilian places get destroyed. By adjusting the falling location the application has to ensure that less number of or no civilian causalities and still the missile range should include the target. Thus the application is able to achieve its purpose. Another motivating example is to estimate the route for police patrol vehicle effectively. Fig. 1 can also be used to illustrate this example. In Fig. 1 Q represents locations that come in patrol route. The points p1-p7 represents some spots like school, hotel, hospital etc. Based on the location of patrol vehicle, most probable route for reaching a place is computed. Thus the application helps to compute range aggregates that help in evaluating the effective route for the police patrols vehicle. The proposed algorithm is based on the standard filtering and verification approach. Falling probability of an anchor point is considered. The presence of anchor point may be anywhere with regard to Q in fig. 1. The aim of the algorithm is to filter as many points as possible so that they do not come under the effect of missile in case of first example. Thus civilian causalities can be effectively reduced.

This paper studies the problem of uncertain location based queries. It proposes a new algorithm which will efficiently compute range aggregates in order to solve the problem of uncertain location based queries in a multidimensional search space.

PRIOR WORK

Location based services have been around for many years recently. It is a challenging problem to compute range aggregates that help in processing such queries accurately. In the literature it is found that there were many researches focused on querying uncertain objects. Broad classification of probabilistic queries and evaluating such queries on uncertain data was 1presented in [1]. There are many types of aggregates. However, this paper focuses on value-based aggregates. There was much focus on probability-thresholding as the problem is significant. For given probability threshold, region, and query, the results will be all objects that are present in the given region with some likelihood. In [2] another approach is proposed by name probability threshold index (PTI) in order to process the query efficiently.

DOI 10.17148/IJARCCE.2015.4673

However, it works only one- dimensional uncertain objects. Of later Agarwal et al. [3] proposed an indexing technique which support range query on one-dimensional uncertain objects. Tao et al. [4] proposed probabilistic constrained region (PCR) for supporting range queries in multidimensional space on uncertain objects. They used probability density function as an intermediary function. They also proposed a pruning technique in order to validate the objects and ensure only necessary objects is processed. Then they improved the technique further in [5] besides another range query problem where objects and locations of query are not certain. This kind of problem was also studied by Chen and Cheng [6] where they applied pruning technique PCR for validation. Range aggregate query processing was explored in [7] over uncertain data. They proposed two approaches for the actual estimation of aggregate values required by range queries. Probabilistic range query was investigated by Dai et al. [8] on uncertain data. Range queries are also studied in [9] and [10] with a constraint. The constraint is that the objects considered in the dataset should follow Gaussian distribution. Then the results are presented according to the rank that satisfied queries. This kind of work which was done recently is in [11] whichfocuson solving the problem of solving indexing uncertain data which is hidimensional in nature.

Apart from range queries, many conventional queries were studied for the purpose of clustering uncertain data as described in [12], [13]. In the same fashion it is done for similarity join [14], [15] and other technologies like skyline query [16], [17], nearest neighbor (NN) queries [18], and top-k queries [19], [20]. However in [21] the experiments are done on uncertain objects which are arranged in the form of a tree named U-tree. For this reason these solutions can't be directly adopted to the problem specified in this paper. However, the problem of this paper is similar to the problem studied in [22]. One difference is that in [22] search region is rectangle in nature which is not impressive for solving such queries. We also studied computation of range aggregates which is quite different from that of [22] and [23]. We also experimented and understood that the PCR technique can also be applied to our problem by modifying it. Ishikawa et al. [24] also studied range queries where the location of query is not precise. However they assumed Gaussian distribution for possible locations. Therefore it can't be generalized and not used in this paper.

PROPOSED SYSTEM TO COMPUTE RANGE AGGREGATES

This section provides information about the problem definition, proposed algorithm to solve the problem, and filters used to prune the search space.

Problem Definition

This paper considers a set of points in a d-dimensional search space. Distance metrics can be used to find the distance between any two objects. In this paper, Euclidian Distance (ED) is used for computing distance between objects. The equation used for distance calculation is as given below.

$$\delta_{min}(r_1, r_2) = \begin{cases} 0 & \text{if } r_1 \cap r_2 \neq \emptyset \\ \min_{\forall x \in r_1, y \in r_2} \delta(x, y) & \text{otherwise.} \end{cases}$$

In many real time applications, with respect to query and distance, users are interested in probable falling points that exceed given threshold. The falling probability is computed as:

$$P_{fall}(Q, p, \gamma) = \int_{x \in Q.region \wedge \delta(x,p) \leq \gamma} Q.pdf(x)dx.$$

Proposed Algorithm for Computing Range Aggregates

The proposed algorithm is based on the concept of filtering and verification and it assumesa set of points which are organized as R-Tree as proposed in [22]. This tree has many entries. Each entry may be an item or an intermediate entry. The former is nothing but a data entry while the latter is a set of points again. While processing the entries, the intermediate entries are to be sent back to the queue again. The algorithm makes use of a threshold for processing uncertain location based query and it should know the falling probability of a specific point with respect to query and distance. The notations used in the algorithm are presented in table 1.

NOTATION	DEFINITION
Q	Uncertain location based query
S	A set of points
q	instance of an uncertain query Q
d	dimensionality
Pq	The probability of the q to appear
Q and Ɣ	probabilistic threshold and query distance
Pfall(Q,p,Ɣ)	the falling probability of p regarding Q and γ
Qθ,γ(S)	{p|p∈ S ∧Pfall(Q, p, γ) ≥ θ}
p, x, y, b(S)	point (a set of data points)
e	R tree entry
Cp,r	a circle(sphere) centred at p with radius r
δ(x, y)	the distance between x and y
δmax(min)(r1, r2)	the maximal(minimal) distancebetween two rectangular regions
gQ	mean of Q
ηQ	weighted average distance of Q
σQ	variance of Q
€	small positive constant value
a	anchor point
nap	number of anchor points
LPfall(p, γ)	lower bound of the Pfall(p, γ)
UPfall(p, γ)	upper bound of the Pfall(p, γ)
nd	the number of different distances pre-computed for each anchor point
Da	a set of distance values used by anchor point a

Table 1 –Notation Summary

As can be seen in table 1, the notations used in the algorithm and also their meaning are presented in table. 1. The algorithm is presented in listing 1.

```
Algorithm 1 Filtering-and-Verification(RS, Q, F, γ, θ)
Input: RS : an aggregate R tree on data set S,
Q : uncertain query, F : Filter, γ : query distance,
θ : probabilistic threshold.
Output: |Qθ,γ(S)|
Description:
Queue := ∅; cn := 0; C := ∅;
Insert root of RS into Queue;
while Queue = ∅ do
    e ← dequeue from the Queue;
    if e is validated by the filter F then
        cn := cn+|e|;
    else
        if e is not pruned by the filter F then
            if e is data entry then
                C := C ∪ p where p is the data point e
                represented;
            else
                put all child entries of e into Queue;
            end if
        end if
    end if
end while
for each point p ∈ C do
    if Pfall(Q, p, γ) ≥ θ then
        cn := cn+ 1;
    end if
end for
Return cn
```

Listing 1 –Algorithm based on Filtering and Verification

As can be seen in listing 1, the proposed algorithm takes R-Tree that contains set of points, uncertain query, filtering technique, query distance and probability threshold. Then it computes range aggregates and finally returns count of validated points. Thus it ensures that more accurate solution can be used which avoids destruction of civilian places as illustrated in the first motivating example. For pruning or validating the filtering technique used is presented in listing 2.

```
Algorithm 2 Quick-Filtering(e, a, γ, θ)
Input: e : entry of RS, a : anchor point
γ : query distance, θ : probabilistic threshold
Output: status : { validated, pruned, unknown}
Description:
Compute δ
max(a, embb);
if γ −δ
max(a, embb) ≥ UD(θ) then
    if δ
    min(a, embb) +γ +>UD(1−θ) then
        Return pruned;
    else
        Return unknown;
    end if
end if
if δ
min(a, embb)−γ −≤ LD(0) then
    if δ
    max(a, embb) < LD(θ)−γ then
        Return pruned;
    else
        Return unknown;
    end if
end if
if UPfall(e
mbb, γ) < θ then
    Return pruned;
else
    Return unknown;
end if
```

Listing 2 –Quick Filtering Algorithm

As can be seen in listing 1, the quick filtering algorithm is used to filter data points. It takes an entry in R-Tree, anchor point, query distance, and probabilistic threshold. After making filtering task it returns the status of the entry or data point. The possible status value is validated or pruned or unknown.

EXPERIMENTS AND RESULTS

We have done experiments with a prototype application. The application is built in Java. The environment used is a PC with 4GB of RAM and Core 2 Dual processor. The source code is developed using NetBeans IDE. The system parameters used in the application are as shown in table 3.

NOTATION	DEFINITION
qr	the radius of the uncertain region (600)
σ	standard deviation for Normal distribution (300)
nap	the number of anchor points in APF (30)
nd	the size of Da for each anchor point (30)
γ	query distance (1200)
θ	probabilistic threshold (∈ [0,1])
η	the number of data points in P (1m)

Table 3 –System Parameters

As seen in table 3, system parameters are used and the experiments are made for many times. The results are presented below in a series of graphs.

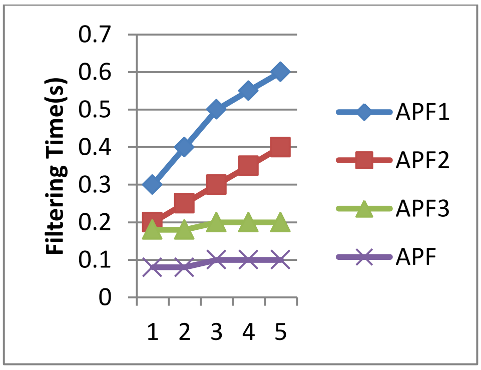

Fig. 2 – Filtering performance with respect to filtering time

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

As can be seen in fig. 2 the horizontal axis represents number of anchor points while the vertical axis represents filtering time. The results reveal that the APF outperforms all others. It proves the fact that the number of anchor points involved does not influence much on the filtering time.

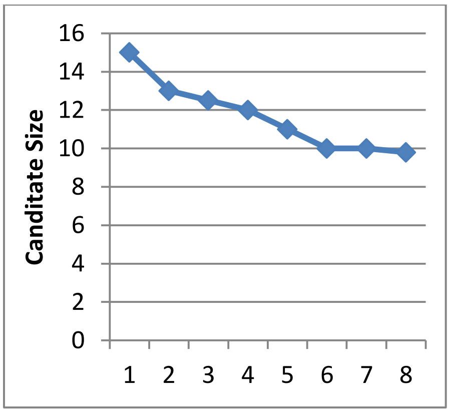

Fig. 3 – Filtering performance with respect to candidate size

As can be seen in fig. 3 the horizontal axis represents a set of distance values used by anchor points while the vertical axis represents candidate size. The results reveal that the candidate size decreases as the number of distance values used by anchor points increases.

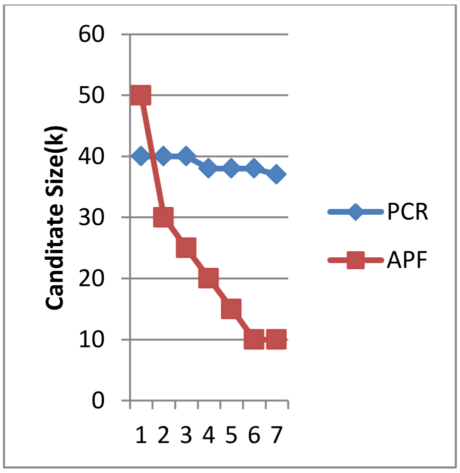

Fig. 4 –Filter performance vs. space usage with respect to candidate size

As can be seen in fig. 4 the horizontal axis represents number of anchor points while the vertical axis represents candidate size. The results reveal that the APF outperforms the other approach. It proves the fact that the performance of APF increases as number of anchor points increases.

As can be seen in fig. 5 the horizontal axis represents number of anchor points while the vertical axis represents filtering time. The results reveal that the APF cost is more as number of anchor points increases.

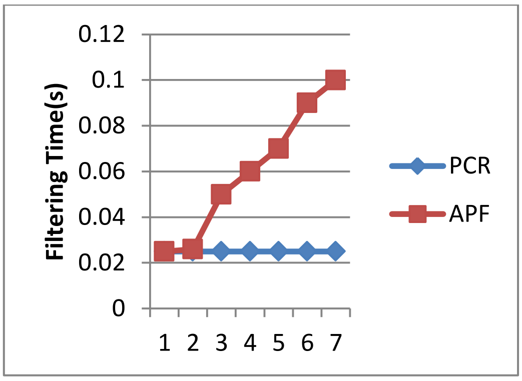

Fig. 5 – Filter performance vs. space usage with respect to filtering time

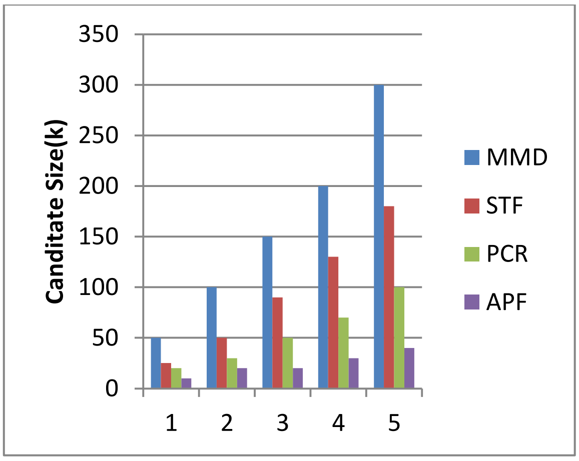

Fig. 6 –Candidate size vs. query distance with “us” dataset

As can be seen in fig. 6 the horizontal axis represents query distance while the vertical axis represents candidate size. When compared with other techniques, the APF performance is better as the number of anchor points grow. The experiment is done with “us” dataset.

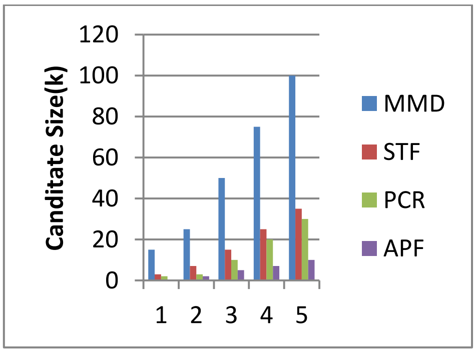

Fig. 7 - Candidate size vs. query distance with “3d uniform points” dataset

As can be seen in fig. 7 the horizontal axis represents query distance while the vertical axis represents candidate size. When compared with other techniques, the APF performance is better as the number of anchor points grow. The experiment is done with “3d uniform points” dataset.

DOI 10.17148/IJARCCE.2015.4673

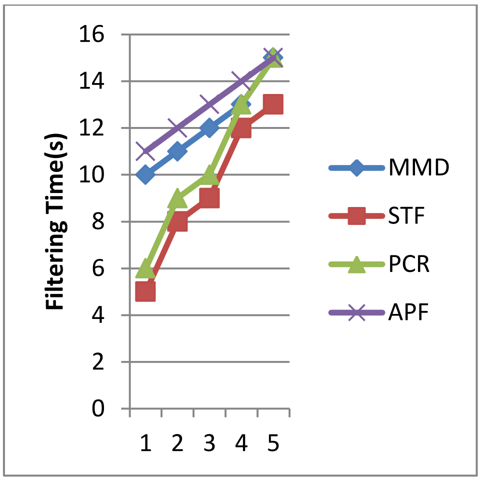

Fig. 8 – Filtering vs. query distance with "us" dataset

As can be seen in fig. 8, the filtering time of the four techniques is presented. The results reveal that as the query distance increases, the response time increases.

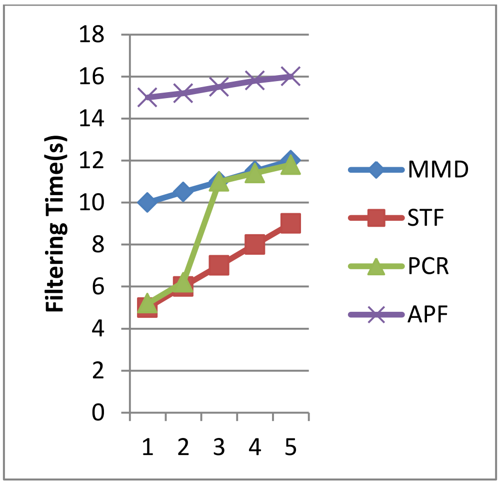

Fig. 9 – Filtering time vs. query distance with "3d uniform points" dataset

As can be seen in fig. 9, the filtering time of the four techniques is presented. The results reveal that as the query distance increases, the response time increases.

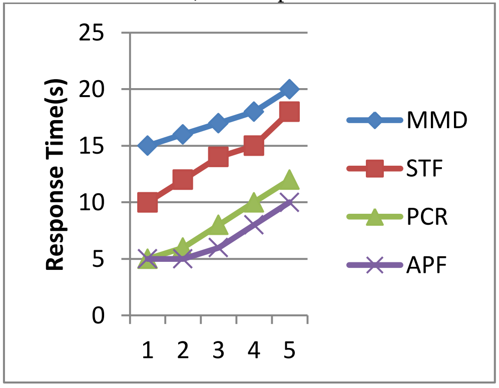

Fig. 10 –Query response time vs. distance with "us" dataset

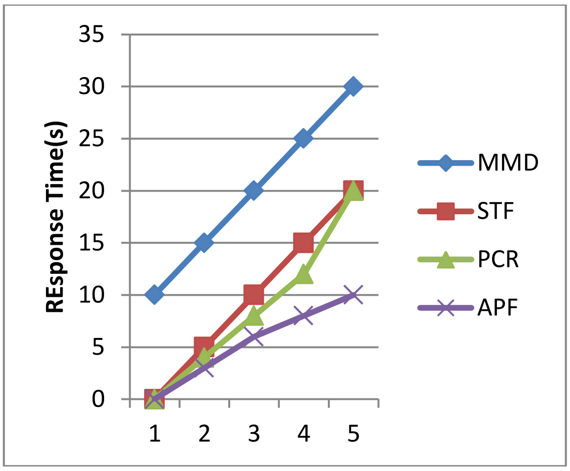

Fig. 11 – Query response time vs. distance with "3d uniform points" dataset

As can be seen in fig. 11, the response time of the four techniques is presented. The results reveal that with respect to response time, the APF technique outperforms all others.

CONCLUSION

This paper formally states the problem of uncertain location based queries in a multidimensional search space. It proposed a new algorithm to compute range aggregates like, sum, max, count efficiently in order to serve uncertain location based queries. We also developed a prototype application to test the efficiency of the proposed algorithm. The experimental results revealed that the proposed algorithm is computationally efficient and can be used in the real world applications.

REFERENCES

[1] R. Cheng, D.V. Kalashnikov, and S. Prabhakar, "Evaluating Probabilistic Queries over Imprecise Data," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2003.

[2] R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J.S. Vitter, "Effcient Indexing Methods for Probabilistic Threshold Queries over Uncertain Data," Proc. Int'l Conf. Very Large Data Bases (VLDB),2004.

[3] C. Bohm, M. Gruber, P. Kunath, A. Pryakhin, and M. Schubert, "Prover: Probabilistic Video Retrieval using the Gauss-Tree," Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE), 2007.

[4] Y. Tao, R. Cheng, X. Xiao, W.K. Ngai, B. Kao, and S. Prabhakar, "Indexing Multi-Dimensional Uncertain Data with Arbitrary Probability Density Functions," Proc. Int'l Conf. Very Large Data Bases (VLDB), 2005

[5] Y. Tao, X. Xiao, and R. Cheng, "Range Search on Multidimensional Uncertain Data," ACM Trans. Database Systems, vol. 32, no. 3, pp. 1-54, 2007.

[6] P.K. Agarwal, S.-W. Cheng, Y. Tao, and K. Yi, "Indexing Uncertain Data," Proc. Symp. Principles of Database Systems (PODS), 2009.

[7] S. Yang, W. Zhang, Y. Zhang, and X. Lin, "Probabilistic Threshold Range Aggregate Query Processing over Uncertain Data," Proc. Joint Int'l Conf. Advances in Data and Web Management (APWeb/WAIM), 2009.

[8] X. Dai, M. Yiu, N. Mamoulis, Y. Tao, and M. Vaitis, "Probabilistic Spatial Queries on Existentially Uncertain Data," Proc. Int'l Symp. Large Spatio-Temporal Databases (SSTD), 2005.

[9] P.K. Agarwal, S.-W. Cheng, Y. Tao, and K. Yi, "Indexing Uncertain Data," Proc. Symp. Principles of Database Systems (PODS), 2009.

DOI 10.17148/IJARCCE.2015.4673

[10] C. Bohm, A. Pryakhin, and M. Schubert, “Probabilistic Ranking Queries on Gaussians,” Proc. 18th Int’l Conf. Scientific and Statistical Database Management (SSDBM), 2006.

[11] C. Aggarwal and P. Yu, “On High Dimensional Indexing of Uncertain Data,” Proc. IEEE 24th Int’l Conf. Data Eng. (ICDE),008.

[12] H.P. Kriegel and M. Pfeifle, “Density-Based Clustering of Uncertain Data,” Proc. 11th ACM SIGKDD Int’l Conf. KnowledgeDiscovery in Data Mining (KDD), 2005.

[13] W.K. Ngai, B. Kao, C.K. Chui, R. Cheng, M. Chau, and K.Y. Yip, “Efficient Clustering of Uncertain Data,” Proc. Int’l Conf. Data Mining (ICDM), 2006.

.[14] H.-P. Kriegel, P. Kunath, M. Pfeifle, and M. Renz, “Probabilistic Similarity Join on Uncertain Data,” Proc. Int’l Conf. Database Systems for Advanced Applications (DASFAA), 2006.

[15] R. Meester, A Natural Introduction to Probability Theory. Addison Wesley, 2004.

[16] J. Pei, B. Jiang, X. Lin, and Y. Yuan, “Probabilistic Skyline on Uncertain Data,” Proc. Int’l Conf. Very Large Data Bases (VLDB),2007.

[17] X. Lian and L. Chen, “Monochromatic and Bichromatic Reverse Skyline Search over Uncertain Databases,” Proc. ACM SIGMOD Int’l Conf. Management of Data, 2008.

[18] R. Cheng, J. Chen, M.F. Mokbel, and C.-Y. Chow, “Probabilistic Verifiers: Evaluating Constrained Nearest-neighbor Queries over Uncertain Data,” Proc. IEEE Int’l Conf. Data Eng. (ICDE), 2008.

[19] M. Hua, J. Pei, W. Zhang, and X. Lin, “Ranking Queries on Uncertain Data: A Probabilistic Threshold Approach,” Proc. ACM SIGMOD Int’l Conf. Management of Data, 2008.

[20] M.A. Soliman, I.F. Ilyas, and K.C. Chang, “Top-k Query Processing in Uncertain Databases,” Proc. Int’l Conf. Data Eng. (ICDE), 2007.

[21] J. Ni, C.V. Ravishankar, and B. Bhanu, “Probabilistic Spatial Database Operations,” Proc. Int’l Symp. Large Spatio-Temporal Databases (SSTD), 2003.

[22] J. Chen and R. Cheng, “Efficient Evaluation of Imprecise Location-Dependent Queries,” Proc. IEEE 23rd Int’l Conf. Data Eng. ICDE), 2007.

[23] V. Bryant, Metric Spaces: Iteration and Application. Cambridge Univ.Press, 1996.

[24] Y. Ishikawa, Y. Iijima, and J.X. Yu, “Spatial Range Querying for Gaussian-Based Imprecise Query Objects,” Proc. IEEE 25th Int’l Conf. Data Eng. (ICDE), 2009.

DOI 10.17148/IJARCCE.2015.4673