Reporting Skyline on Uncertain Dimension with Query Interval

Nurul Husna Mohd Saad[1], Hamidah Ibrahim[1], Fatimah Sidi[1], Razali Yaakob[1], Ali Amer Alwan[2]
[1]*Faculty of Computer Science and Information Technology,Universiti Putra Malaysia.*
[2]*Kulliyyah of Information and Communication Technology, International Islamic University Malaysia.*
nhusna.saad@gmail.com

***Abstract*—Naturally, users sometimes specify their preference in an imprecise way (i.e. query with an interval/range). To report results that satisfy the imprecise query as well as interesting would be easy on dataset with atomic values. The challenge is when the dataset being queried consists of both atomic values as well as continuous range of values. For a set of objects with *uncertain dimension* and given a query interval $[q_i, q_i']$ on that *uncertain* dimension, a skyline query on that interval returns the objects which are not dominated by any other objects in the query interval. A method is proposed to help determine objects that intersect with the query interval and answer skyline query that satisfy the query interval. The correctness of the method is proven through comparisons between two methods that strictly reject and loosely accept objects from/into the query interval.**

***Index Terms*—Dataset; Skyline; Uncertain Dimension; Query Interval.**

I. INTRODUCTION

A constrained skyline query is reporting skyline objects that are within a specified query interval [8]. To compute skyline objects in a query interval, it is preferable to find objects that are not dominated by any other objects. In principle, an object v is preferable if, in the query interval, v is better than another object w on at least one dimension and v is not worse than w on every dimension. It is quite straightforward to report skyline that is within a query interval when dealing with a set of objects that are all points. Only points that fall within the query interval will be considered for skyline computations. Consider the apartments example in Figure 1, where a tenant desired to rent apartments within the rent range 5-9. Hence, the skyline in this scenario would be points e, k, and l, as they are the most desirable apartments in the specified rent range.

Now, let's consider the following scenario: the rental database for houses and apartments that supports *uncertain dimensions* [12] contains listing on thousands of houses/apartments for rent. Each apartment is associated with its rental price (can be fixed or within some range), the commute length, the size (sq. ft.), the number of bedrooms and bathrooms, etc. A potential tenant who would like to limit his search within a rent budget may ask, "*With rent ranged between $250 and $450, find me apartments that are as cheap as possible and as close as possible to the workplace.*" Thus, can skyline be efficiently reported straightforwardly using existing skyline algorithms when dealing with a set of objects with *uncertain dimension*, which can be points or line segments, and the objects with line segments intersect the query interval? It is very obvious that for every object that does not lie within the query interval, they can definitely be filtered out, and every object that clearly falls within the query interval will be accepted for further skyline computation. Nevertheless, how does one determine to accept or reject objects that intersect with the boundary of a query interval? Figure 2 illustrates the above discussion.

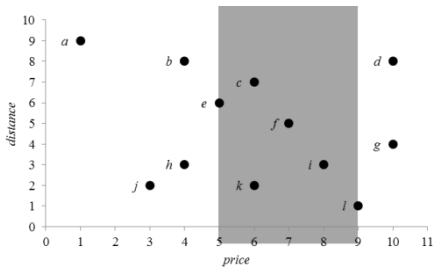


Figure 1: Example of query interval

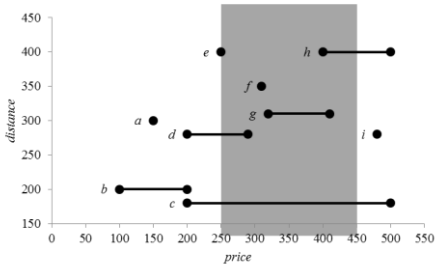


Figure 2: Example of query interval on uncertain dimension

Throughout the paper, the focus of this work is on how to determine objects that satisfy a given query interval in order to be considered in skyline reporting when objects in *uncertain dimension* intersect with the query interval yet they do not fully lie inside it. To the best of our knowledge, this is the first work that tackles the problem of skyline with query interval on *uncertain dimension*. Note that maintaining a data structure in the algorithm is not the main concern of this paper. Various works exist on implementing the most suitable data structure (i.e., radix priority search tree

[4], range tree [11], grid index [13], etc.) and maintaining them, and is easily applicable into this work later on.

The rest of the paper is organized as follows. Section II reviewed the related works of skyline and query intervals on *uncertain dimension*. Then, the preliminaries of this paper and the problem of query interval are formerly defined in Section III. The proposed approach is discussed in Section III as well, followed by an empirical study in Section IV. Finally, Section V concludes the paper.

II. Related Works

The evolution of skylines in the context of databases can be seen from the first work by Borzsonyi *et al.* [1] where they have developed the *block-nested-loop* (*BNL*) and the *divide-and-conquer* (*D&C*) algorithms, as well as proposing SQL syntax for skyline queries. Then, Chomicki *et al*. [2] introduced presorting into *BNL* to build a more effective algorithm, which is called *sort-filter-skyline* (*SFS*) algorithm, in which then Godfrey *et al*. [3] have further improved it. Kossmann *et al*. [6] then introduced an algorithm based on the nearest neighbor search using R-tree. The algorithm recursively searches the nearest neighbor, which cost more time and space, even though the skyline query speeds up with the algorithm. To overcome this problem, Papadias *et al*. [8] have proposed the *branch-and-bound skyline* (*BBS*) algorithm that is based on the sorted R-tree. To this day, it is known as one of the best skyline query methods on centralized databases.

In recent years, skyline computations have been extended when the importance of uncertainty in databases has been recognized. Pei *et al*. [9] pioneered the concept of probabilistic skyline on uncertain data, in which each object is represented by a set of instances and is part of the skyline answer with a certain probability. Inspired by this work, Qi and Atallah [10] then focused on the problem of computing skyline probabilities for all instances by motivating instance-level probabilistic skylines without threshold. Meanwhile, Lian and Chen [7] proposed the *probabilistic reverse skyline* algorithm over uncertain data for both monochromatic and bichromatic cases. Then, Zhang *et al*. [14] investigated the problem on how to efficiently compute skyline queries against sliding windows over uncertain data streams. On the other hand, Khalefa *et al*. [5] introduced the concept of probabilistic skyline on uncertain data, where the objects are of continuous range instead of having multiple instances. While skyline query in [5] deals with all objects having continuous range in a dimension, the work by Saad *et al*. [12] introduced the concept of *uncertain dimensions* and computes skyline probabilities when objects in a dimension can be points as well as continuous range (i.e. line segments).

Later on, various studies have started incorporating range queries (e.g. queries with interval, hyper-rectangle region, arbitrary area, etc.) into skyline algorithms in order to refine the skyline answer sets when users described their preferences in an imprecise way (i.e. issuing range queries). Papadias *et al*. [8] coined the term constrained skyline query where the query would returned the most interesting points in the data space defined by the constraints. They illustrated how the *BBS* algorithm can be implemented for constrained skyline query processing. Jiang and Pei [4] tackle the problem of computing interval skyline queries on time series data, where each timestamp is considered as a dimension and each time series is considered as a point in the space. Working on the same context, Rahul and Janardan [11] proposed an algorithm to first filter out points that do not fall within the *range space* based on their *range attributes* and then compute skyline points in the *range space* with respect to the *feature space*. Wang *et al*. [13] then proposed a new issue on dynamic skyline computation involving range queries, in which to satisfy the range query, if no data points exist within the range query, then dynamic skyline is performed.

Despite having a plethora of studies focusing on skyline query and its variant, until now there is no work accomplishes on skyline with query interval regarding *uncertain dimension*. The works in [8, 4, 11, 13] focused on implementing a suitable data structure to efficiently search and report skyline points that lie within the range query, and the datasets involved are considered as points only. Following [12], this work focuses on datasets with *uncertain dimensions*, while investigating on how to choose skyline sets that are within the query interval when there are objects that intersect with the query interval.

III. Preliminaries And Proposed Method

In this section, the concept of *uncertain dimensions* is briefly described, followed by the introduction on the issue of reporting skyline set with query interval for *uncertain dimensions*.

A. Definition 1 (Uncertain Dimension)

Given a dataset of $\boldsymbol{n}$-dimensional space $\boldsymbol{D} = (\boldsymbol{D_1}, \boldsymbol{D_2}, \dots, \boldsymbol{D_n})$. A dimension is said to be an *uncertain dimension*, denoted U′$(\boldsymbol{D_i})$ where $\boldsymbol{1 \le i \le n}$, if there exists two objects in $\boldsymbol{D}$ with different forms in that dimension.

Let $\boldsymbol{v} = (\boldsymbol{v.D_1}, \boldsymbol{v.D_2}, \dots, \boldsymbol{v.D_n})$ and $\boldsymbol{w} = ([\boldsymbol{w.D_1}: \boldsymbol{w.D_1'}], \boldsymbol{w.D_2}, \dots, \boldsymbol{w.D_n})$, where $[\boldsymbol{w.D_1}: \boldsymbol{w.D_1'}]$ is an interval representing continuous range in dimension $\boldsymbol{D_1}$ and $\boldsymbol{w.D_1 < w.D_1'}$, be two objects in $\boldsymbol{D}$, such that $\{\boldsymbol{v}, \boldsymbol{w}\} \in \boldsymbol{D}$. Here, the uncertain dimension would be U′$(\boldsymbol{D_1})$ since both objects $\boldsymbol{v}$ and $\boldsymbol{w}$ are represented in different forms in $\boldsymbol{D_1}$.

For ease of description and without loss of generality, this work assumes that the dataset has only one *uncertain dimension*, namely the first dimension, and as such the dataset $\boldsymbol{D}$ has the form (U′$(\boldsymbol{D_1}), \boldsymbol{D_2}, \dots, \boldsymbol{D_n})$. Given the nature of the *uncertain dimension*, the results of skyline query executed on this kind of dataset are bound to be probabilistic, since each object with continuous range is now associated with a probability value of it being a query answer. This issue has been solved in [12]. Subsequently, when query interval is introduced into skyline query, the aim is to determine objects that lie inside the query interval.

B. Definition 2 (Query Interval)

A query interval $[\boldsymbol{q_j}: \boldsymbol{q_j'}]$ indicates a range that is being queried on $\boldsymbol{j^{th}}$ dimension, where $\boldsymbol{q_j < q_j'}$. We say $\boldsymbol{v.D_j} \in [\boldsymbol{q_j}: \boldsymbol{q_j'}]$ if $\boldsymbol{q_j \le v.D_j \le q_j'}$.

Let $D = (\text{U}'(D_1), D_2, \dots, D_n)$ be a dataset in n-dimensional space. Since $\text{U}'(D_1)$ is an *uncertain dimension*, the dataset D is bound to have two sets of objects, $\{A, I\} \in D$, such that A represents a set of objects with atomic value in D_1, while I represents a set of objects with interval in D_1. Having a set of objects that does not involve any *uncertain dimension* would be straightforward to report skyline objects that lie within the query interval. An object $v \in A$ is a skyline object only if there does not exist an object $k \in A$ that dominates v. Hence, v is said to dominate object k with respect to the query interval, denoted by $v \prec_{[q_j:q_j']} k$, if $\exists D_j, v.D_j \in [q_j:q_j'] < k.D_j \in [q_j:q_j']$ and $\forall D_{i,i\neq j}, v.D_i \leq k.D_i$.

On the contrary, having interval query on *uncertain dimension* would be challenging as there will be objects with continuous range, denoted by $[\boldsymbol{w.D_j}:\boldsymbol{w.D_j}']$ where $\boldsymbol{w} \in \boldsymbol{I}$, that can intersect the query interval, and therefore, the concept of $\boldsymbol{w.D_j} \in [\boldsymbol{q_j}:\boldsymbol{q_j}']$ as previously defined is not applicable in this case.

C. Problem definition

Let $\boldsymbol{S}$ be a set of objects in $\boldsymbol{D}$, where $\boldsymbol{D} = (\text{U}'(\boldsymbol{D_1}), \boldsymbol{D_2}, \dots, \boldsymbol{D_n})$, and an interval $[\boldsymbol{q_1}:\boldsymbol{q_1}']$ queried on the *uncertain dimension* $\text{U}'(\boldsymbol{D_1})$. Report skyline on $\boldsymbol{S}$ with respect to the query interval $[\boldsymbol{q_1}:\boldsymbol{q_1}']$ in such a way that the skyline objects satisfy the query interval $[\boldsymbol{q_1}:\boldsymbol{q_1}']$.

For the purpose of this paper, this work assumes that the query interval posed by a user is on a single *uncertain dimension*.

D. Skyline with Query Interval on Uncertain Dimension

Let S be a set of objects in a d-dimensional space with *uncertain dimension*, $D = (\text{U}'(D_1), D_2, \dots, D_n)$. To answer query interval on *uncertain dimension*, an algorithm that filters S is needed, in such a way that all objects reported satisfy the query interval $[q_1:q_1']$. There are several cases where two sets of objects A and I, such that $A \cup I = S$, can lie within or intersect with the query interval $[q_1:q_1']$. The easiest and simplest case is when object $v \in A$ lies entirely inside the query interval $[q_1:q_1']$ (as illustrated in Figure 3), such that $q_1 \leq v.D_1 \leq q_1'$, and it can definitely be reported as object that satisfies the query interval $[q_1:q_1']$. The same can be said for object $w \in I$ that lies entirely inside the query interval $[q_1:q_1']$ (as illustrated in Figure 3), in such a way that $q_1 \leq w.D_1 < w.D_1' \leq q_1'$. The next case is when object w has one endpoint inside the query interval $[q_1:q_1']$ (as illustrated in Figure 4), where $w.D_1 < q_1 < w.D_1' \leq q_1'$ or $q_1 \leq w.D_1 < q_1' < w.D_1'$. And lastly, when object w intersects the query interval $[q_1:q_1']$ but does not have an endpoint inside the query interval $[q_1:q_1']$ (as illustrated in Figure 5), in which $w.D_1 < q_1 < q_1' < w.D_1'$. In the latter two cases, it remains to decide whether those objects should be reported as objects that satisfy the query interval $[q_1:q_1']$ and be included in skyline computation at a later stage.

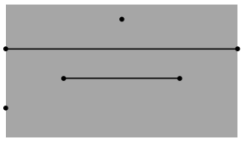


Figure 3: Objects that definitely satisfy the query interval

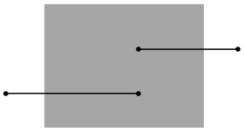


Figure 4: Intersecting objects have one endpoint within the query interval

Figure 5: Both endpoints of an intersecting object lie outside of the query interval

Since object w in D_1 is a continuous range modeled as a uniform probability density function $pdf\ f(x)$ defined on the real range $[w.D_1:w.D_1']$, then $P(w.D_1 < D_1(w) < w.D_1') = \int_{w.D_1}^{w.D_1'} f(x)dx = 1$. Note that $D_1(w)$ refers to object w in D_1. This fact can be used to find object w that satisfy the query interval $[q_1:q_1']$ by having the probability of $D_1(w)$ being between the query interval $[q_1:q_1']$ above a threshold t value.

$$P(q_1 < D_1(w) < q_1') \geq t\ . \tag{1}$$

For example, if $[w.D_1:w.D_1']$ intersects with the query interval $[q_1:q_1']$ and has an endpoint within the query interval $[q_1:q_1']$, such that $w.D_1 < q_1 < w.D_1' \leq q_1'$, to determine if w will be reported as object that satisfies the query interval $[q_1:q_1']$, then the probability of $D_1(w)$, $\int_{q_1}^{w.D_1'} f(x)dx$, shall be more than t. When $[w.D_1:w.D_1']$ intersects with the query interval $[q_1:q_1']$ yet with both its endpoints being outside of the query interval $[q_1:q_1']$, then the probability of $D_1(w)$ being within the query interval $[q_1:q_1']$ is computed as $\int_{q_1}^{q_1'} f(x)dx$.

The threshold t value is important as it is impossible to determine that for objects with continuous range, they will always satisfy the query interval $[q_1:q_1']$ and the same t value will be used when computing skyline on *uncertain dimension* [12]. Once a set of objects S' that satisfies the query interval $[q_1:q_1']$ is obtained, skyline query is then performed on S'. The skyline computation follows the work proposed in [12].

Due to limited space, detailed discussion on computing skyline on S', which comprise an *uncertain dimension*, is omitted in this paper, though the extensive study on it can be found in [12]. Algorithm 1 gives a method to retrieve objects that satisfy the query interval $[q_1:q_1']$ with a probability above a given threshold value. The computation of skyline in query interval $[q_1:q_1']$ follows the computation in the work proposed in [12].

Algorithm 1

Input: dataset S with U(D_1), query interval $[q_1 : q_1']$, threshold t;
Output: the skyline on $[q_1 : q_1']$;
Description:

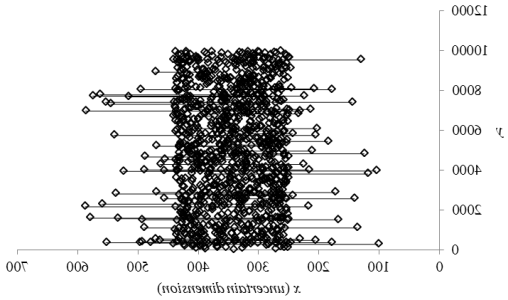
```
1:  Initialize S';
2:   for each object v ∈ S do
3:    if q1 ≤ v.D1 ≤ q1' then
4:    add v to S';
5:    end if
6:     if v.D1 < q1 < v.D1' ≤ q1' or
       q1 ≤ v.D1 < q1' < v.D1' or
       v.D1 < q1 < q1' < v.D1' then
7:     compute P(D1(v)) w.r.t [q1: q1'];
8:        if P(D1(v)) ≥ t then
9:        add v to S';
10:       end if
11:    end if
12:  end for
13:  Sky = compute skyline on S' as applied in [12];
14:  return Sky;
```

IV. EMPIRICAL STUDY

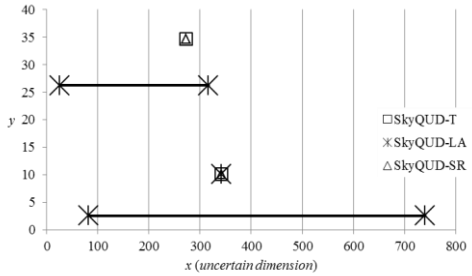
To study the correctness of the proposed method, denoted as SkyQUD-*T*, which is adopted from SkyQUD algorithm [12], a comparison on the set of skyline objects reported is conducted. Due to the lack of previous work, the following two naïve methods are considered as a basis of comparison: the SkyQUD algorithm, yet instead (1) utilizing the concept of *strictly rejecting* any object that intersects the boundary of query interval, denoted as SkyQUD-*SR*, and (2) utilizing the concept of *loosely accepting* any object that intersects the boundary of query interval, denoted as SkyQUD-*LA*.

The comparison is performed on synthetic dataset that has been generated for 100,000 objects on two dimensions. Each dimension represents a uniform random variable from 1 to 10,000, and the first dimension is set as *uncertain dimension*. The query interval $[q_1 : q_1']$ is set as [250:440] and [2000: 5000] and the threshold t is set to 50%. For the purpose of this paper, it is assumed that the given query interval is on the *uncertain dimension*.

Figure 6(a) and Figure 7(a) exhibit objects for all three methods that satisfy the boundaries set by a given query interval, while Figure 6(b) and Figure 7(b) demonstrate that given a query interval $[q_1 : q_1']$, skyline objects reported with respect to the query interval by SkyQUD-*T* are reported by SkyQUD-*SR* and SkyQUD-*LA* as well. In Figure 6(b), the cases discussed in previous section are illustrated, where continuous range objects intersect the query interval and their endpoints either lie within or outside of the query boundaries. SkyQUD-*LA* reports both objects with such cases, even when clearly it can be seen that such objects have small probability of them being within the query interval. This is due to with SkyQUD-*LA*, the method will accept all objects that intersect with the query interval without taking into consideration the probability of these objects being within the given query interval. By applying threshold, SkyQUD-*T* has taken into consideration of the probability of a continuous range object that will satisfy the query interval while still reporting a correct set of skyline objects, as the threshold is applied when computing the probability of objects being within the query interval as well as the probability of objects being a skyline object with respect to the query interval.

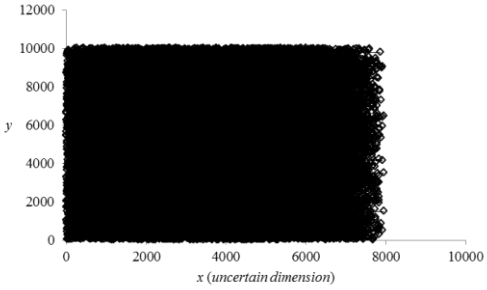


(a) Objects that satisfy the query interval

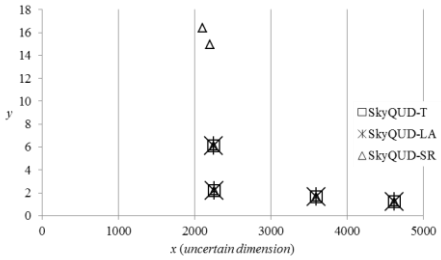


(b) Skyline on the query interval.

Figure 6: Skyline objects on *uncertain dimension* with respect to the query interval [250: 440]



(a) Objects that satisfy the query interval



(b) Skyline on the query interval

Figure 7: Skyline objects on *uncertain dimension* with respect to the query interval [2000: 5000]

V. CONCLUSION

The issue of computing skyline on *uncertain* dimension within a given query interval is investigated in this paper. A method that incorporated a threshold value is proposed in order to filter out objects that intersect with the query interval

yet having a probability of them being within the query interval less than the threshold value. To demonstrate the correctness of the proposed method, two naïve methods are proposed that implemented a simple concept of strictly rejecting and loosely accepting objects that intersect with the given query interval before computing skyline on them. The skyline objects reported by these two methods are than compared to the skyline objects reported by the proposed method. As future work, it is interesting and challenging to utilize a suitable data structure in the computation of skyline query on *uncertain dimension* when given a query interval as it helps to speed up the searching/filtering process.

REFERENCES

[1] Borzsonyi, S., Kossmann, D., Stocker, K.: The Skyline Operator. In Proceedings of International Conference on Data Engineering (ICDE'01), Heidelberg, Germany. (2001) 421-430.

[2] Chomicki, J., Godfrey, P., Gryz, J., Liang, D.: Skyline with Presorting. In Proceedings of International Conference Data Engineering (ICDE'03), Bangalore, India. (2003) 717-719.

[3] Godfrey, P., Shipley, R., Gryz, J.: Maximal Vector Computation in Large Data Sets. In Proceedings of the 31th International Conference on Very Large Data Bases (VLDB'05), Trondheim, Norway. (2005) 229-240.

[4] Jiang, B., Pei, J.: Online Interval Skyline Queries on Time Series. In Proceedings of the 2009 IEEE International Conference on Data Engineering. (2009) 1036-1047.

[5] Khalefa, M.E., Mokbel, M.F., Levandoski, J.J.: Skyline Query Processing for Uncertain Data. In Proceedings of the Conference on Information and Knowledge Management. (2010) 1293-1296.

[6] Kossmann, D., Ramsak, F., Rost, S.: Shooting Stars in the Sky: An Online Algorithm for Skyline Queries. In Proceedings of International Conference on Very Large Data Bases (VLDB'02), Hong Kong, China. (2002) 275-286.

[7] Lian, X., Chen, L.: Monochromatic and Bichromatic Reverse Skyline Search over Uncertain Databases. In Proceedings of the ACM SIGMOD International Conference on Management of Data. (2008) 213-226.

[8] Papadias, D., Tao, Y., Fu, G., Seeger, B.: Progressive Skyline Computation in Database Systems. ACM Transactions on Database Systems (TODS). Vol. 30 (2005) 41-82.

[9] Pei, J., B. Jiang, B., Lin, X., Yuan, Y.: Probabilistic Skylines on Uncertain Data. In Proceedings of the International Conference on Very Large Database. (2007) 15-26.

[10] Qi, Y., Atallah, M.J.: Identifying Interesting Instances for Probabilistic Skylines. In Proceedings of the International Conference on Database and Expert Systems Applications. (2010) 300-314.

[11] Rahul, S., Janardan, R.: Algorithms for Range-Skyline Queries. In: Cruz, I.F., Knoblock, C.A., Kröger, P., Tanin, E., Widmayer, P. (eds.) SIGSPATIAL/GIS, (2012) 526–529.

[12] Saad, N.H.M., Ibrahim, H., Alwan, A.A., Sidi, F., Yaakob, R.: A Framework for Evaluating Skyline Query over Uncertain Autonomous Databases. Elsevier, Vol. 29 (2014) 1546–1556.

[13] Wang, W.-C., Wang, E.T., Chen, A.L.P.: Dynamic Skylines Considering Range Queries. In: Yu, J.X., Kim, M.H., Unland, R. (eds.) DASFAA 2011, Part II. LNCS, Vol. 6588, Springer, Heidelberg (2011) 235–250.

[14] Zhang, W., Lin, X., Zhang, Y., Wang, W., Yu, J.: Probabilistic Skyline Operator over Sliding Windows. In Proceedings of the International Conference on Data Engineering. (2009) 1060-1071.