A Skyline Query Processing Approach over Interval Uncertain Data Stream with K-Means Clustering Technique

Zarina Dzolkhifli, Hamidah Ibrahim, Fatimah Sidi,
Lilly Suriani Affendey, Siti Nurulain Mohd Rum
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
Malaysia
e-mail: {zarinadzol@gmail.com, hamidah.ibrahim, fatimah, lilly, snurulain}@upm.edu.my

Ali Amer Alwan
Kulliyyah of Information and Communication Technology,
International Islamic University Malaysia
Malaysia
e-mail: aliamer@iium.edu

***Abstract*—Skyline query processing which extracts a set of interesting objects from a potentially large multidimensional dataset has attracted significant research attention in many emerging important applications. Although skyline computation has been studied extensively for data streams, there has been relatively less work on uncertain data stream. Only recently, a few methods have been proposed to process uncertain data stream, however data uncertainty in these works is restricted to objects having many instances. In contrast, there is no work that has considered uncertainty due to objects having interval values wherein the exact values of the objects are not known at the point of processing. Hence, in this paper a skyline query processing approach utilising the K-Means clustering technique is proposed to efficiently compute skyline over interval uncertain data stream.**

Keywords-skyline query processing; uncertain data; data stream.

I. INTRODUCTION

Nowadays, real-time data streams processing technologies play an important role in enabling time-critical decision making in many applications. Handling streaming data is particularly challenging since it is continuously generated by an array of sources and devices and is delivered in a wide variety of formats. The abundance of data streams has led to new algorithmic paradigms for processing them. Processing data streams is intricate due to several reasons: (i) the objects in the streams arrive online, (ii) the system has no control over the order in which objects arrive to be processed, either within a data stream or across data streams, (iii) data streams are potentially unbounded in size, and (iv) once an object from a data stream has been processed it is discarded or archived, it cannot be retrieved easily unless it is explicitly stored in memory, which typically is small relative to the size of the data streams [1].

For the past last decade, skyline query processing over data streams has attracted significant research attention in many emerging important applications. Although skyline computation for data streams has been studied extensively [5][7][12][15][16][17][22][23], there has been relatively less work on uncertain data stream. Uncertain data are defined as data which are inaccurate, imprecise, untrusted, and unknown. In fact, there is no work that focuses on uncertainty due to objects having interval values. The fast flowing of continuously generated data with uncertainty by an array of sources and devices complicates the query process and the amount of computations for processing the uncertain data stream is generally huge. It becomes more complicated when the values of the objects are nondeterministic, i.e., objects having interval values wherein the exact values of the objects are not known at the point of processing. For example, the prices of objects *a*, *b*, *d*, *e*, and *g* shown in Table 1 are in interval form. Here, one cannot derive the exact skyline but can only compute the probability of an object being a skyline member. In addition, identifying the domination between objects is not straightforward especially when the interval values of the objects intersect. For instance, one cannot state that object *a* dominates object *b* or object *b* dominates object *a* as the values of their prices intersect. Thus, identifying an efficient approach that is capable of computing skylines before the objects become obsolete to meet the time-critical expectancy of the applications is vital. It is also important to ensure that the approach can avoid the re-computation of probabilities of objects being skylines.

TABLE 1: A SNAPSHOT OF SAMPLES OF DATA STREAM

Object	Price	Rating	Distance	…	Arrival Time (ms)
a	200 – 600	5	1.5		1
b	300 – 450	3	2.5		2
c	500	4	4.0		4
d	100 – 200	2	5.5		6
e	700 – 800	5	2.0		7
f	900	5	1.0		12
g	400 - 500	3	3.5		13

Hence, this paper attempts to tackle the issues of efficiently computing skyline over interval uncertain data stream. An approach that can handle uncertain data stream is proposed with the aim to reduce the cost of skyline computation while ensuring that the time-critical expectancy of applications is met. Two main tasks are identified, namely: clustering and skyline processing. Clustering

 ISBN: 978-1-61208-715-3

technique is utilised to group objects that are similar into the same cluster. This will assist in identifying objects (clusters) that are dominated by other objects (clusters). The skyline processing is then employed to select the most dominant objects from each cluster and between clusters.

This paper is organised as follows: Section II presents the works related to the study. In Section III, definitions and notations that are used in the rest of the paper are set out. Our proposed approach is elaborated in Section IV. We have performed two analyses to evaluate the performance of our proposed approach. This is presented in Section V. Conclusion and future works are presented in the final section of this paper, Section VI.

II. RELATED WORK

Skyline query processing has been studied extensively for the last past decade. The earliest works focus on finding algorithm to expedite the process of identifying skylines for static dataset. These algorithms, which are based on non-indexing method include *Divide & Conquer* (*D&C*) [2], *Block Nested Loop* (*BNL*) [2], *Sort-Filter-Skyline* (*SFS*) [3], and *LESS* [6]. Then, algorithms using precompute indexes were proposed. These include *NN* [9], *Branch-and-Bound Skyline* (*BBS*) [17], and *ZSearch* [10]. There are also works that focus on uncertain data, such as *p-skyline*, which is designed for probabilistic skyline queries over static uncertain databases [18], *Iskyline* which supports skyline query on data that are represented as continuous ranges [8], and *SkyQUAD* a probabilistic skyline processing on interval values with threshold approach [19][20].

In the last decade, skyline query processing over data streams has attracted significant research attention in many emerging important applications, such as internet search logs, network traffic, sensor networks, and scientific data streams (such as in astronomic, genomics, physical simulations, etc.). In such applications, the challenge mainly lies in the huge volume of data, as well as its fast arrival rate. Moreover, it is impossible to reserve all the streaming items in memory, thus one-pass algorithms should be devised to adapt to the streaming data. Applying the existing methods of processing queries on this huge fast flowing data streams can be costly, time consuming, and impractical [1]. Several algorithms have been proposed for continuously monitoring skyline changes over ing data, which include *Lazy* and *Eager* algorithms [17], *LookOut* algorithm [16], and *FAST* algorithm [11]. On the other hand, the work by [12] focuses on skyline query of *n-of-N* data streams model in sliding window.

Recently, works have focused on processing skyline queries over uncertain data streams. This includes the work by [21] where skylines are identified based on objects having many varying instances with time. Works such as [24] and [4] have proposed efficient techniques in finding probabilistic skyline objects based on sliding windows on possible semantic. The work by [13] proposed the Effective Probability Skyline Update (*EPSU*) method by defining the interesting probabilistic skyline objects to return to the users and efficiently finding these objects without enumerating all possible objects. A sliding window partitioning strategy is proposed in [14] in order to reduce the processing time of the probability skyline computation. However, most of these works focus on objects having many instances. On the other hand, there is no work that identifies skyline over uncertain data stream, where uncertainty is due to objects having interval values.

III. PRELIMINARIES

In this section, we provide the definitions and notations that are related to skyline queries over uncertain data stream, which are necessary to clarify our proposed approach. Our approach has been developed in the context of multidimensional data stream, D, which consists of a set of objects, $D = \{o_1, o_2, o_3, \ldots\}$. An object of the database D is denoted by $o_i(d_1, d_2, \ldots, d_m)$ where o_i is the ith object with m-arity and $d = \{d_1, d_2, \ldots, d_m\}$ is the set of dimensions. In the following, we first give the general definitions that are related to skyline queries (Definitions 1 to Definition 5). Then, we extend these definitions to suit with uncertain data stream (Definitions 6 to Definition 9).

Definition 1 (Skyline): The set of skylines, S, is defined as those objects that are not dominated by any other objects in the dataset.

Definition 2 (Dominate): Given two objects o_i and $o_j \in D$ dataset with d dimensions, o_i dominates o_j (the lesser the better) (denoted by $o_i \prec o_j$) if and only if the following condition holds: $\forall\, d_k \in d,\ o_i.d_k \leq o_j.d_k \wedge \exists d_l \in d,\ o_i.d_l < o_j.d_l$.

Definition 3 (Skyline Queries): Select an object o_i from the set of objects D if and only if o_i is as good as o_j (where $i \neq j$) in all dimensions and *strictly* in at least one dimension. We use S to denote the set of skyline objects, $S = \{o_i \mid \forall\, o_i, o_j \in D, o_i \prec o_j\}$.

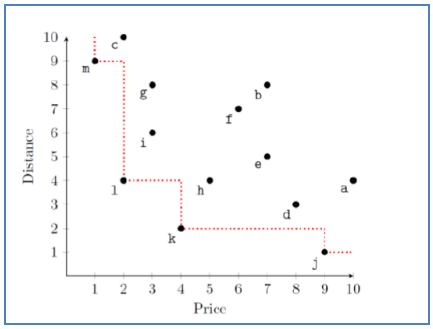
There are various forms of uncertain data. In this work, we focus on uncertain data where the object is expressed in an imprecise way, i.e., the exact value of the object is not known at the point of processing. This form of data is continuously generated especially in data stream.

Definition 4 (Uncertain Data): An object $o_i(d_1, d_2, \ldots, d_m)$ is said to contain uncertain data if at least one of its dimensions, d_j, contains value in the form of interval, i.e., $o_i[d_j] = [l, u]$ where l is the lower bound value and u is the upper bound value.

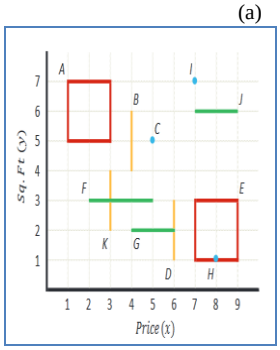
Definition 5 (Skyline over Uncertain Data) [19]: An object $o_i \in D$ with uncertain data is a skyline object if it has

a probability of not being dominated by other object $o_j \in D$ more than a threshold value, H.

Figure 1(a) shows objects with exact values for dimensions price and distance. If we assumed minimum values are preferred in both dimensions, then the set of skyline objects returned is {*m*, *l*, *k*, *j*}. Meanwhile, Figure 1(b) shows examples of objects with interval values. In this example, we cannot state that object *A* definitely dominates object *B*, and vice versa, or that object *F* dominates object *K* with 100% probability.



(a)



(b)

AptID	Price	Sq. Ft
A	1 - 3	5 - 7
B	4	4 - 6
C	5	5
D	6	1 - 3
E	7 - 9	1 - 3
F	2 - 5	3
G	4 - 6	2
H	8	1
I	7	7
J	7 - 9	6
K	3	2 - 4

Figure 1. (a) Skyline Example (b) Example of Uncertain Data [19]

Since a data stream is often unbounded, a query over a data stream is generally specified with a sliding window. The sliding-window model works based on recent objects of the data stream, whereas older objects are not taken into account, as they are considered obsolete.

Definition 6 (Data Stream): A data stream D contains a set of objects, $D = \{o_1, o_2, o_3, \ldots\}$ that arrive in sequence where each object is associated with a timestamp that indicates the arrival time of the object. We use the notation t_{oi} to indicate the arrival time of object o_i.

Definition 7 (Sliding Window) [11]: A sliding window, w_i, represents equally sized time intervals that are defined based on the parameters RANGE and SLIDE where RANGE specifies the length of the window extent and SLIDE specifies the step by which the window extent moves. Based on these parameters, the size of the sliding window can be easily determined.

Consider the hotel reservation systems where hotels continuously advertise their competitive deals to the system. The system contains streaming of millions of hotels for booking. Each hotel is associated with rating, distance from the city center, price etc. (see Table 1). The price advertised by the hotels might be in the form of exact value or within some price range. A potential user may ask the most preferable deals advertised during the recent 5 hours (*w*3 in Figure 2) and wants to update the results every 1 hour. In this example, RANGE = 5 and SLIDE = 1.

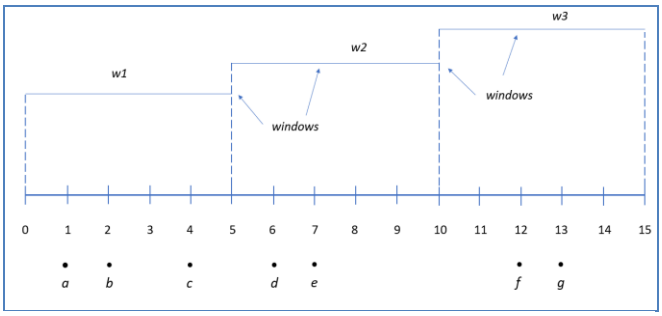


Figure 2. Sliding Windows [11]

Definition 8 (Skyline of a Sliding Window): The set of skylines of a window w_i, S_{wi}, is defined as those objects that are not dominated by any other objects in the window w_i.

Definition 9 (Skyline over Uncertain Data of a Sliding Window): An object of a given window, $o_i \in w_i$, with uncertain data is a skyline object if it has a probability of not being dominated by other object $o_j \in w_i$ more than a threshold value, H.

IV. THE PROPOSED APPROACH

Figure 3 presents our proposed approach in processing skyline queries over interval uncertain data stream. The proposed approach consists of four main stages, as explained below:

A. Identifying the Sliding Windows of a Given Skyline Query

Given a skyline query, SQ_q, the sliding window of the query is identified utilising the values of RANGE and SLIDE parameters. For each window, w_i, the objects that fall within the window are analysed. This is depicted in Figure 3(a). In this example, the second substream (window) contains 20 objects that are *A*, *B*, *C*, …, *T*. Objects like *B*, *E*, *J*, *M*, *N*, *O*, *P*, *R*, *S*, and *T* contain interval values in the first dimension and are considered as uncertain.

B. Grouping the Objects of a Sliding Window

In this stage, objects are grouped based on the type of data they contain. Objects with exact values are grouped

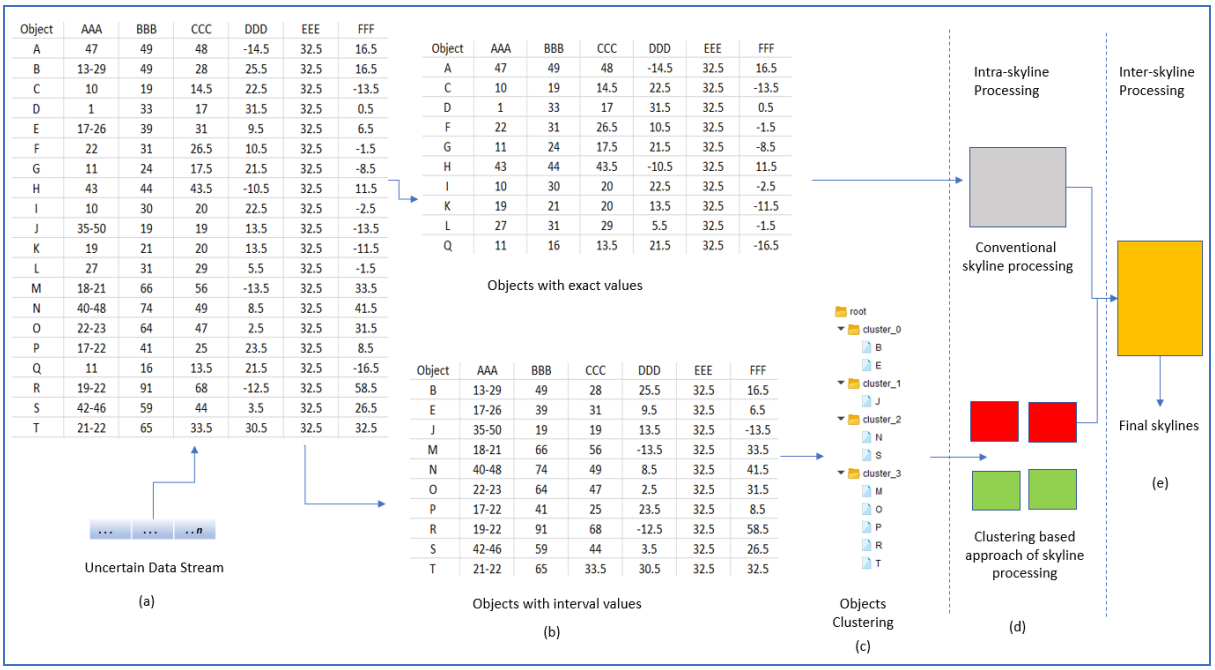


Figure 3. The Proposed Approach of Skyline Query Processing over Interval Uncertain Data Stream

together (G_A) while objects with interval values are put together in another group (G_R). This is shown in Figure 3(b). Based on the example, G_A = {*A*, *C*, *D*, *F*, *G*, *H*, *I*, *K*, *L*, *Q*} and G_R = {*B*, *E*, *J*, *M*, *N*, *O*, *P*, *R*, *S*, *T*}.

C. Clustering the Objects

Objects in the G_R group are then clustered. In this work, the K-Means clustering technique is utilised. The objects are clustered only based on the dimension having interval values. Since the clustering technique requires deterministic values as input, thus the interval value of an object, for instance, $o_i[d_j] = [l, u]$, is represented by its mean value. We have performed several analyses, in which the interval value of an object is represented with its min, max, as well as mean value. Based on the analyses, representing the interval value with its mean value produces better set of clusters. However, due to limited space, the results of these analyses are not presented here. The result of this stage is a set of *k*-clusters denoted as $C = \{C_1, C_2, C_3, \ldots, C_k\}$. Figure 3(c) presents samples of clusters formed through this stage. There are four clusters for this example, labelled as *cluster_0*, *cluster_1*, *cluster_2*, and *cluster_3*.

D. Identifying the Skylines

In this final stage, the skylines are determined. Two levels of skyline processing are performed, namely: intra-skyline processing and inter-skyline processing.

Intra-skyline Processing – At this level, the candidate skylines are identified by comparing the objects within the group/cluster. The conventional skyline processing is utilised for the group of objects with exact values, i.e., those in the G_A group as shown in Figure 3(d). Here, the dominance comparison is performed at the object level. Based on the G_A derived from the second stage, object *C* dominates object *I* while object *Q* dominates object *G*. Both objects *I* and *G* are removed from further processing, while the other objects are considered as the candidate skylines of group G_A.

While those objects in the G_R group, the dominance comparison is performed at the cluster level instead of objects. Since the mean value is used to represent the interval value, thus the minimum (C_i-min) and maximum (C_i-max) mean values of a cluster are used to identify cluster domination, which is defined below:

Definition 10 (Cluster Domination): Given two clusters C_i and $C_j \in C$, C_i dominates C_j (denoted by $C_i \prec C_j$) if and only if the following condition hold: the C_i-max of C_i < C_j-min of C_j. C_i is called non-dominated cluster, while C_j is called dominated cluster.

Obviously, if a cluster, C_i, dominates a cluster C_j, this implies that the objects of cluster C_i dominate every object in cluster C_j. The objects of C_i are the candidate skylines. For example, comparing *cluster_0* = {*B*, *E*} and *cluster_2* = {*N*, *S*}, the *cluster_0-max* = 21.5, which is less than the *cluster_2-min* = 44. This means that both objects *B* and *E* dominate objects *N* and *S*. A great number of probability computations can be avoided especially if the clusters contain a huge number of objects. However, since the cluster domination is only based on the dimension with interval values, the dominated clusters still have chances to be candidate skylines based on the other dimensions.

Nevertheless, it will not involve any probability computations.

Inter-skyline Processing – Here, the final skylines are identified by performing dominance comparison between the candidate skylines produced by conventional skyline processing and also those produced through the clustering domination. This means domination comparison is performed between the candidate skylines of G_A, non-dominated clusters, and dominated clusters. This is as shown by Figure 3(e).

V. EVALUATION

We have performed two simple analyses to get initial findings on the performance of our proposed approach. These analyses are conducted using RapidMiner [25] as a tool to cluster the objects. The first analysis aims to prove that at least one non-dominated cluster is identified. Having the non-dominated cluster implies that objects in the cluster can be omitted from further processing. The second analysis aims to prove that our proposed approach utilising the clustering technique can improve skyline processing by reducing the number of pairwise comparisons. For both analyses, we have varied the number of objects from 50 to 1000 objects. Every object has only a single dimension with interval values, which are generated randomly. Every interval value is within 20% of the range of possible values.

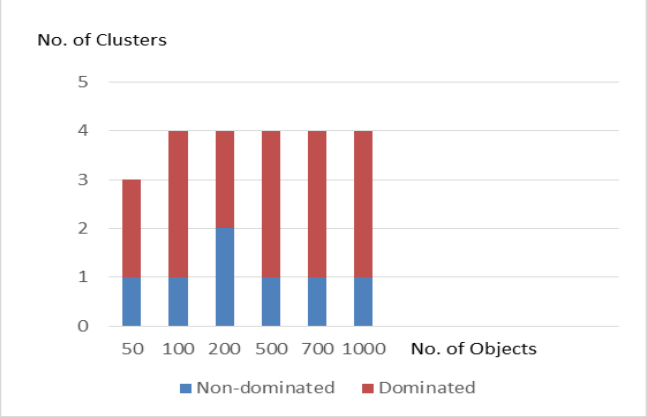


Figure 4. The Number of Non-dominated and Dominated Clusters

Figure 4 presents the results of the first analysis, which shows the number of non-dominated and dominated clusters formed when the number of objects varies from 50 to 1000. From this figure, the following can be observed:

(i) At least one non-dominated cluster is identified, which implies that the objects of the non-dominated cluster dominate the objects of the other clusters (dominated cluster).

(ii) The number of non-dominated clusters formed is not being affected by the number of objects. This can be clearly seen when the number of objects increased from 50 to 1000, the number of non-dominated cluster is always 1 (except for 200 objects).

(iii) Although the number of non-dominated and dominated clusters formed is almost the same when the number of objects varies from 50 to 1000, the number of objects in the clusters is not the same, i.e., the size of each cluster when the number of objects is 1000 is larger as compared to when the number of objects is 50.

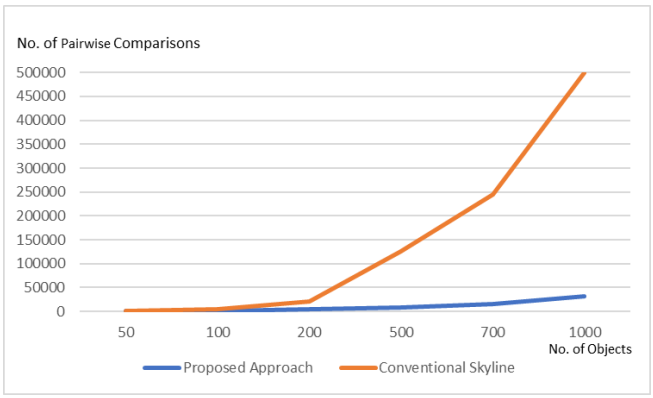


Figure 5. Number of Pairwise Comparisons

Figure 5 presents the results of the second analysis, which shows the number of pairwise comparisons performed by our proposed approach as compared to the conventional skyline processing approach. From the figure, the followings are observed:

(i) When the number of objects increases, the number of comparisons performed by our proposed approach increases steadily. While the number of comparisons performed by the conventional skyline approach shows a sudden increment as the number of objects increases.

(ii) On average the percentage of reduction with respect to number of pairwise comparisons gained by our proposed approach as compared to conventional skyline approach is 89.11%.

VI. CONCLUSION

This paper addresses the issues of processing skyline queries over interval uncertain data stream. An approach utilising the K-Means clustering technique has been proposed with the aim to reduce the number of pairwise comparisons. Two analyses have been conducted to get initial findings on the performance of the proposed approach. Results show that the approach is able to significantly reduce the number of pairwise comparisons. We will attempt to perform detailed analyses in the future to examine the proposed approach with regards to other aspects, such as scalability, distribution of interval values, and huge data size.

REFERENCES

[1] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems,"

Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 1 – 16, 2002.

[2] S. Borzsonyi, D. Kossmann, and K. Stocker, "The skyline operator," Proceedings of the 17th. International Conference on Data Engineering, pp. 421 – 430, 2001.

[3] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, "Skyline with presorting: theory and optimizations," Proceedings of the Intelligent Information Processing and Web Mining, pp. 595 – 604, 2005.

[4] X. Ding, X. Lian, L. Chen, and H. Jin, "Continuous monitoring of skylines over uncertain data streams," Journal of Information Sciences, vol. 184(1), Feb. 2012, pp. 196 – 214, doi:10.1016/j.ins.2011.09.007.

[5] L. Dong, G. Liu, X. Cui, and T. Li, "Finding group-based skyline over a data stream in the sensor network," Journal of Information, vol. 9(2), Feb. 2018, pp. 1 – 22, doi:10.3390/info9020033.

[6] P. Godfrey, R. Shipley, and J. Gryz, "Maximal vector computation in large data sets," Proceedings of the 31st. International Conference on Very large Data Bases, pp. 229 – 240, 2005.

[7] X. Guo, H. Li, A. Wulamu, Y. Xie, and Y. Fu, "Efficient processing of skyline group queries over a data stream," Journal of Tsinghua Science Technology, vol. 21(1), Feb. 2016, pp. 29 – 39, doi: 0.1109/TST.2016.7399281.

[8] M. E. Khalefa, M. F. Mokbel, and J. J. Levandoski, "Skyline query processing for uncertain data," Proceedings of the 19th. ACM International Conference on Information and Knowledge Management, pp. 1293 – 1296, 2010.

[9] D. Kossmann, F. Ramsak, and S. Rost, "Shooting stars in the sky: An online algorithm for skyline queries," Proceedings of the 28th. International Conference on Very Large Data Bases, pp. 275 – 286, 2002.

[10] K. C. Lee, W. C. Lee, B. Zheng, H. Li, and Y. Tian, "Z-SKY: An efficient skyline query processing framework based on z-order," Journal of Very Large Data Bases, vol. 19(3), June 2010, pp. 333 – 362, doi:10.1007/s00778-009-0166-x.

[11] Y. W. Lee, K. Y. Lee, and M. H. Kim, "Efficient processing of multiple continuous skyline queries over a data stream," Journal of Information Sciences, vol. 221, Feb. 2013, pp. 316 – 337, doi: 10.1016/j.ins.2012.09.040.

[12] X. Lin, Y. Yuan, W. Wang, and H. Lu, "Stabbing the sky: efficient skyline computation over sliding windows," Proceedings of the 21st. International Conference on Data Engineering (ICDE '05), pp. 502 – 513, 2005.

[13] C. Liu and S. Tang, "An effective probabilistic skyline query process on uncertain data streams," Proceedings of the 6th. International Conference on Emerging Ubiquitous Systems and Pervasive Networks, pp. 40 – 47, 2015.

[14] J. Liu, X. Li, K. Ren, J. Song, and Z. Zhang, "Parallel n-of-N skyline queries over uncertain data streams," Proceedings of the International Conference on Database and Expert Systems Applications, pp. 176 – 184, 2018.

[15] H. Lu, Y. Zhou, and J. Haustad, "Efficient and scalable continuous skyline monitoring in two-tier streaming settings," Journal of Information Systems, vol. 38(1), Mac 2013, 68 –81, doi: 10.1016/j.is.2012.05.005.

[16] M. Morse, J. M. Patel, and W. I. Grosky, "Efficient continuous skyline computation," Journal of Information Sciences, vol. 177(17), Sept. 2007, pp. 3411 – 3437, doi: 10.1016/j.ins.2007.02.033.

[17] D. Papadias, Y. Tao, G. Fu, and B. Seeger, "Progressive skyline computation in database systems," Journal of ACM Transactions on Database Systems (TODS), vol. 30(1), Mar. 2005, pp. 41 – 82, doi: 10.1145/1061318.1061320.

[18] J. Pei, B. Jiang, X. Lin, and Y. Yuan, "Probabilistic skylines on uncertain data," Proceedings of the 33rd. International Conference on Very Large Data Bases, pp. 15 –26, 2007.

[19] N. H. M. Saad, H. Ibrahim, A. A. Alwan, F. Sidi, and R. Yakoob, "A framework for evaluating skyline query over uncertain autonomous databases," Proceedings of the International Conference of Computational Science (ICCS 2014), pp. 1546-1556, 2014.

[20] N. H. M. Saad, H. Ibrahim, A. A. Alwan, F. Sidi, and R. Yakoob, "Computing range skyline query on uncertain dimension," Proceedings of the Database and Expert System Applications (DEXA), pp. 377 – 388, 2016.

[21] H. Z. Su, E. T. Wang, and A. L. Chen, "Continuous probabilistic skyline queries over uncertain data streams," Proceedings of the International Conference on Database and Expert Systems Applications, pp. 105 – 121, 2010.

[22] Z. Wang, J. Xin, L. Ding, J. Ba, and X. Gao, "ρ-Dominant skyline computation on data stream," Journal of IEEE Access, vol. 6, Sept. 2018, pp. 53201 – 53213, doi:10.1109/ACCESS.2018.2871254.

[23] J. Xin, G. Wang, L. Chen, X. Zhang, and Z. Wang, "Continuously maintaining sliding window skylines in a sensor network," Proceedings of the International Conference on Database Systems for Advanced Applications, pp. 509 – 521, 2007.

[24] W. Zhang, A. Li, M. A. Cheema, Y. Zhang, and L. Chang, "Probabilistic n-of-N skyline computation over uncertain data streams," Journal of World Wide Web, vol. 18(5), Sept. 2015, pp. 1331 – 1350, doi:10.1007/s11280-014-0292-2.

[25] https://rapidminer.com/. Last accessed 21 February 2019.

 ISBN: 978-1-61208-715-3