How to cite this paper:

Ma'aruf Mohammed Lawal, Hamidah Ibrahim, Fazlida Mohd Sani, & Razali Yaakob. (2017). Skyline computation of uncertain database: A survey in Zulikha, J. & N. H. Zakaria (Eds.), Proceedings of the 6th International Conference of Computing & Informatics (pp 84-90). Sintok: School of Computing.

SKYLINE COMPUTATION OF UNCERTAIN DATABASE: A SURVEY

Ma'aruf Mohammed Lawal[1,2], Hamidah Ibrahim[2], Fazlida Mohd Sani[2], Razali Yaakob[2]

[1]*Ahmadu Bello University, Nigeria, mmlawal80@gmail.com*
[2] *Faculty of Computer Science and Information Technology*
Universiti Putra Malaysia
{hamidah.ibrahim, fazlida, razaliy}@upm.edu.my

ABSTRACT. Conducting advance skyline analysis over certain and uncertain databases is still an evolving research area in the field of database, despite several research works that have been conducted in this area. This paper conducts a survey on research issues on computing skyline for uncertain databases, with the view of providing interested researchers with an overview of the most recent research directions in this area. It further suggests possible research direction on skyline processing for uncertain databases. Taxonomy of the existing approaches is also presented.

Keywords: skyline query, uncertain databases, uncertain dimension.

INTRODUCTION

Recently, skyline analyses of databases have been largely populated due to the huge benefit that can be derived from it. Skyline data analysis can support the development of a multi-criteria decision tool for retrieving non-trivial, interesting information. These databases may contain discrete, range values or combination of both values, captured either by modern electronic/computing devices or as a result of integrating several similar databases into a single computing platform. Uncertain data found among captured data has made traditional skyline query processing no longer applicable, thus making it practically difficult and almost impossible in retrieving non-trivial information that can support multi-criteria decision-making from these databases (Börzsönyi et al., 2001, Chuan-Ming & Syuan-Wei, 2015). For this reason, it has become imperative to survey relevant research works that support skyline query processing of uncertain databases, in order to explore evolving issues regarding skyline queries over databases, provide readers with an overview of the state-of-the-art, and also suggest future research direction in this area.

The rest of this paper is organized as follows: the next section presents concepts that are related to skyline query processing which is then followed by taxonomy and related works section, and finally the last section of this article concludes this paper.

PRELIMINARY

This section provides the definitions and concepts that are related to skyline query processing on uncertain databases.

Definition 1 (***Dominance***) Assume that minimum value is preferred. Given two objects, p and q with the same number of dimensions; object p is said to dominate object q, formally written as $p \prec q$, if and only if object p is as good as object q in all dimensions and better than object q in at least one dimension.

Definition 2 (*Skyline Query*) Given a set of objects O, an object p of O is said to be a *skyline object* if and only if there does not exist any other objects k in O which dominates p. Then the skyline on O is the set of all skyline objects in O. Applying the skyline definition on objects presented in Fig. 1, with the assumption that minimum value is preferred for both dimensions, then the set of skyline objects is {a, b, c}.

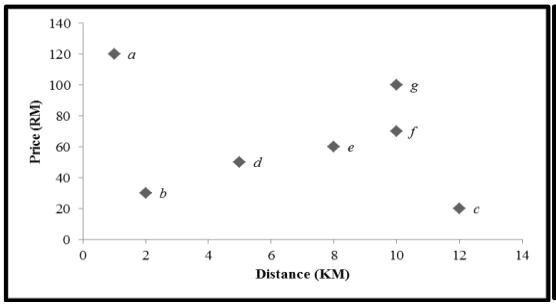

Figure 1: Skyline query

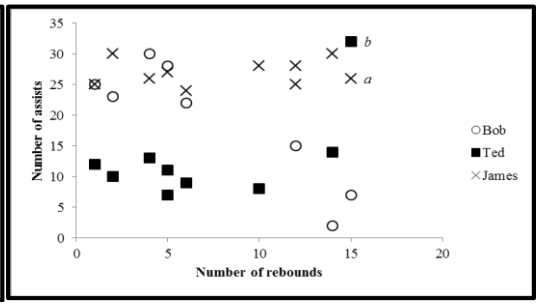

Figure 2: Uncertain database at the object level

Definition 3 (***Uncertain Database***) Let D be an n-dimensional database. The database D can either contain uncertainty at the object level if each object of D has several instances or at the dimension level if any of its objects can have different forms of data values for a dimension. Fig. 2 is an example of a database with uncertainty at the object level, while Fig. 3 and Fig. 4 depict examples of uncertain database at the dimension level.

Definition 4 (*Uncertain Dimension*) Given a database D with n-dimensions, a dimension A_j is said to be an *uncertain dimension* if it contains different data value representations (i.e. points and range values).

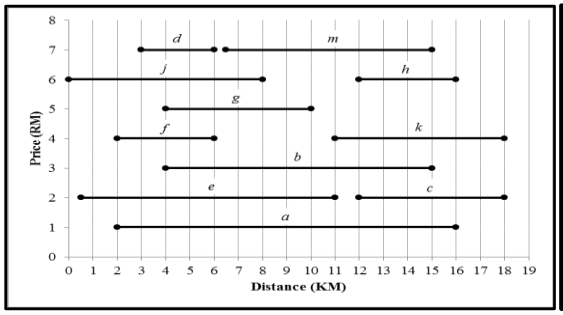

Figure 3: Uncertain database with homogeneous dimension

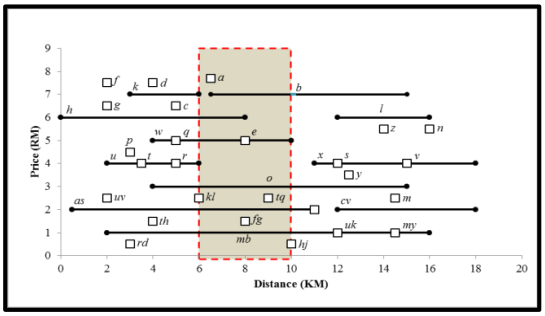

Figure 4: Uncertain database with heterogeneous dimension

If the entire data values of any dimension in D are of the same form (i.e. either all are points or all are range values), such database is said to be an uncertain database with homogeneous dimensions; otherwise, if it contains different data value representations (i.e. points and range values) the database is referred to as uncertain database with heterogeneous dimensions. An example of uncertain database with homogeneous and heterogeneous dimension is depicted in Figure 3 and Figure 4, respectively.

RELATED WORKS

Not until recently, a lot of works have been conducted on skyline processing of certain databases which have been very useful in retrieving non-trivial information from certain databases. Uncertain databases are emanated either as a result of populating databases with uncertain data captured by modern computing devices or the representation of the data value in the database. With uncertain database, it becomes very difficult in realizing valuable information using the traditional skyline query (Saad et al., 2016). Several skyline computation works on uncertain databases have been inspired due to the benefits that can be achieved if applications for retrieving non-trivial information are developed (Saad et al., (2016), Khalefa et al., (2010), Agarwal et al., (2011), Papadias et al., (2003), and Liu et al., (2013)). The skyline objects for uncertain database are derived using a probabilistic skyline query in finding the dominance between the objects involved. Fig.1 shows the taxonomy of skyline techniques and its categories based on the uncertain database and the level at which the uncertainty of the database is established.

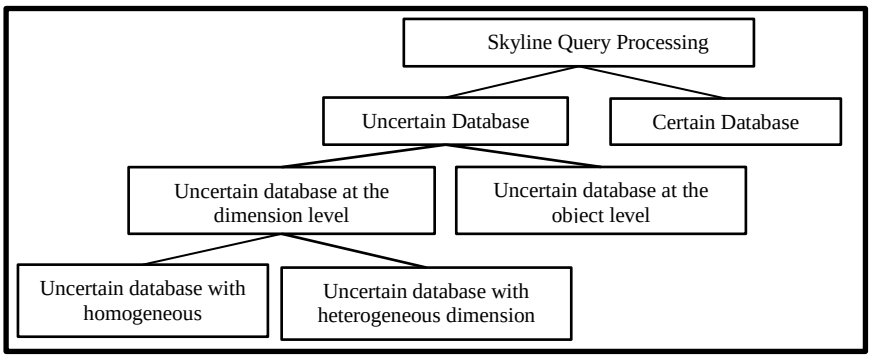

Figure 5: Taxonomy of skyline query processing approaches

Even though, some skyline query processing works over certain databases are discussed here, the essence is to highlight how data structures are utilized to drastically speed up skyline query processing.

Generally, skyline query processing can be categorized into two distinct groups based on the approach followed in processing the skyline query, namely: index-based and non-index approach. The non-index based algorithm requires no preprocessing operations or data structures unlike the index-based algorithms; instead the operations to be executed are performed directly on the database. Among the non-index based algorithm is the Divide-and-Conquer (D&C) algorithm (Han et al., 2013). The D&C divides the data space into two equal halves, along its certain dimensions. This process is repeated for each of these partitions to get their respective skylines. The final skyline objects are returned by computing the binary predicates for these two sets of results. The Block-Nested-Loop (BNL) algorithm is a naïve, generic algorithm for computing skyline objects. This approach scans the database sequentially to maintain a set of candidate skylines in the memory buffer (Börzsönyi et al., 2001). Each tuple is compared to the candidate skyline in the buffer, if the tuple dominates other tuples in the buffer, these tuples are replaced by the dominating tuple; otherwise it is discarded. Similarly, if the scanned tuple is incomparable with the candidate skyline and there is no space for it to be inserted into the memory buffer then it is written into a temporary file in a disk. The content of the buffer before the insertion into the temporary file constitutes part of the skyline result, and the remaining candidate skylines are compared to the tuples stored in the temporary file.

The index-based algorithm requires a pre-constructed data structure that supports the evaluation of skyline in an efficient manner. The pre-constructed data structures ensure that not all the entire database is scanned each time a query is to be processed. This category of algorithm typically uses a *B*-Tree which is a two ordered indices for computing the skyline, a *R*-Tree which is a spatial index tree, usually used for skyline query that considers all dimensions of the object or a *Bitmap* for pre-constructing the data structure to be used for processing the query. Example of such algorithm includes Nearest-neighbor (NN) proposed by Kossmann et al. (2002). It utilizes the existing methods of nearest neighbor search to recursively split the data space. The Branch and Bound Skyline (BBS), an NN-based algorithm proposed by Papadias et al. (2003) for evaluating skyline objects using the *R*-tree structure (an index structure with multiple dimensions). Another index-based skyline algorithm is the Bitmap, which is used to represent the data in bit representation. Each tuple is encoded as *m*-bit vector, where *m* is the cardinality of the dimensions in the database (Tan et al., 2001). This computed vector is used to decide which object is a skyline member by encoding and storing the coded database as a bit transposed file such that if more than a single one-bit is returned in the bitwise this implies that the *nth* point dominates *x* (represents distinct value for a dimension) or otherwise.

Pie et al. (2007) proposed *p*-skyline, a probability-based skyline query approach for evaluating the skyline of uncertain databases. It considers uncertain database at the object level, where object's uncertainty is represented as a probability distribution over discrete databases. To achieve *p*-Skyline, first it derives the probability of each object to be a skyline and consequently uses a probability threshold value *p* to return the skyline by pruning off objects whose chances of being skyline objects are very slim (with probability < *p*). The only set back arises when the probability threshold value specified by the user is low or high. Specifying the permissible probability threshold value that will yield correct result remains an issue. Just like Pie et al. (2007), Agarwal (2011) also considers uncertain database at the object level. While Pie et al. (2007) focus on how to devise several standard procedures for computing *p*-Skyline (probability skyline) efficiently, Agarwal (2011) proposed a simple, near linear time approximation algorithm for evaluating the skyline membership of each point (a point belongs to the skyline membership, if its ε-approximate lies in the skyline). The primary aim of his work lies on improving the existing method for computing ε-approximate skyline solution. This improvement is achievable by reducing the problem to a rectangle-stabbing problem, and then to orthogonal searching in group model. A rectangle is constructed for each uncertain point by projecting in opposite direction for the coordinates of each point. This algorithm can extend to higher dimensions at the cost of increasing the running time.

Liu et al. (2013) proposed the *U*-skyline algorithm, a skyline query algorithm for returning tuples with the highest probability threshold value as the skyline answers. The introduction of *U*-skyline algorithm is due to the inefficiency associated with *p*-skyline algorithm. In *p*-skyline, there are skyline tuples undesirably dominating each other, and this will not allow the algorithm to yield any optimal skyline result. Thus, the number of tuples returned as skyline objects is affected by the probability threshold value specified. In the *U*-skyline algorithm a number of optimization techniques are adopted for processing query efficiently. This includes computational simplification of *U*-skyline probability by combining probability computation for multiple objects that share the same skyline sets; pruning off unqualified candidate skylines and early termination of query processing; reduction of input database by obtaining the subset of the whole database after all dominated tuples are discarded, such that the *U*-skyline for this subset is the same as the *U*-skyline of the whole databases; and partition and conquest of the reduced database through D&C technique to split the reduced database into

disjointed subsets. These disjointed subsets are processed independently. Subsequently, the individual skylines are merged to form the final *U*-skyline result.

Works on retrieving skyline objects for uncertain databases at the object level include (Han et al., 2013) who proposed a novel and efficient skyline algorithm named SSPL. This algorithm utilizes a pre-constructed data structure (sorted positional index list), which requires a low space overhead to reduce I/O cost significantly, by pruning off any positional index whose corresponding object is not a candidate skyline object. The obtained candidate skyline objects are then processed to get the skyline result by scanning the pre-constructed data structure in a selective and sequential manner. The sorted positional index list is pre-constructed from the attributes of a database such that each entry of the constructed sorted positional index list is referenced by a pair $[PI, A_j]$, where *PI* is the index of the tuple in the database corresponding to that attribute entry and A_j is the attribute corresponding to the same entry, as such the sorted positional index list consists of all sorted entries of the database, with each entry still maintaining its index $[PI, A_j]$. The SSPL utilizes a much smaller and feasible data structure to achieve a comparable performance to a tree-based algorithm that requires exponential number of indexes before the required skyline criteria can be covered. Although SSPL algorithm returns skyline result efficiently, but it seems not feasible and practically almost impossible and applicable to the real world scenario when the size of skyline criteria increases. A large database will definitely not only increase the number of sorted positional index list to be constructed but also increase the effort and time required for processing the query, as the size of the criteria becomes larger.

Chuan-Ming and Syuan-Wei (2015) proposed an Effective Probabilistic Skyline query process on Uncertain data (EPSU) algorithm, with the primary objective of having a time efficient approach for deriving the skyline probabilities of all the data objects in the current sliding window. EPSU uses the *U*-skyline probability threshold value approach, a compliment of *p*-skyline. This algorithm returns a valid skyline with maximum probability threshold value, for the processing of uncertain databases with continuous distribution on the instances, which are generated and will be invalid after a period of time interval. It requires minimal computational time to return correct results by leveraging on *SW*-tree structure that provides and supports an effective data update and better data management. The *SW*-tree is an augmented *R*-tree data structure, use for indexing data objects in the sliding window. By indexing the data objects it further helps in managing the memory buffer registers efficiently. However, the EPSU algorithm cannot support skyline processing for a large volume of data.

So far, research works on skyline computation either uses the index-based or the non-index based techniques to compute the skyline of the database. The index-based technique comprises of sorting, categorizing, and the use of data structure to organize the input databases. Index-based techniques have been seen to be quite useful in query optimization instances, by playing a major role in computational time required for processing a query. The data structure is meant to significantly speed up skyline computation by pruning off nodes which do not satisfy the query when computing the skyline objects. Morse et al. (2007) proposed *LookOut*, a time conscious algorithm for evaluating skylines over data stream. This work focused on the certainty aspect of the database, and also examined the effect of the underlying quadtree data structure when computing current skyline objects in a continuous environment. It contributed to the significant performance of *LookOut* over previous works that use R^*-tree index structures, by using continuous time-interval skyline operator as against the general sliding window approach and also using a quadtree data structure for supporting skyline computation of large database as against the R^*-tree structure in (Kossmann et al., 2002).

Processing of skyline for uncertain databases at the dimension level in continuous domain has started receiving attention. Khalefa et al. (2010) proposed a probabilistic skyline query approach for evaluating skyline objects over uncertain databases with homogeneous dimensions. Each object is associated with a probability value while users will have to specify a threshold value to determine the skyline objects. Specified probability value decides which among the objects is a skyline. This work evaluates the skyline by utilizing the uncertainty reduction, pairwise comparison, and bound tightening method. Though this work has provided an efficient framework for answering skyline query over uncertain databases with homogeneous dimension, but user specification of probability threshold value can affect the overall skyline result. If the threshold is too low, it may result into too many answers or if it is too high, it may return small set of candidate skyline objects. Though this work has successfully presented a probabilistic skyline query algorithm, but can only be found applicable to only uncertain database with homogeneous dimensions.

To the best of our knowledge, SkyQUD algorithm proposed by Saad et al. (2014) is the only skyline algorithm for answering skyline query over uncertain database with heterogeneous dimensions. Unlike the probabilistic approach proposed by Khalefa et al. (2010), SkyQUD can retrieve skyline objects from either uncertain database with homogeneous or heterogeneous dimensions. In this approach, the uncertain database is grouped into two sets, each of which is treated differently to get their candidate skylines. One set contains objects with atomic values while the other contains objects with range values. The traditional skyline query is used to process the first set, while objects with range values are processed with a probabilistic interval skyline query algorithm by Khalefa et al. (2010), to get the set of candidate skyline objects. The results obtained from these sets are finally combined and treated with the probabilistic skyline query to obtain the final skyline objects.

Table 1: A summary of skyline query processing approaches

***HM- homogeneous dimension * HT- heterogeneous dimension *OL (Uncertainty at the object level) * DD (Discrete Domain)**

No.	Author	Data structure used	Characteristics of the uncertain database					Type of Database
			Uncertainty model		Level at which uncertainty exist			
			DD	CD	OL	DL		
						HM	HT	
1.	Han et al., 2013	Sorted positional index-list	-	-	-	-	-	Certain & Static
2.	Börzsönyi, et al., 2001	-	-	-	-	-	-	Certain & Static
3.	Kossmann et al., 2002	R^*-tree	-	-	-	-	-	Certain & Dynamic
4.	Papadias et al., 2003	*R*-tree	-	-	-	-	-	Certain & Static
5.	Tan et al., 2001	Bitmap	-	-	-	-	-	Certain & Static
6.	Pei et al., 2007	*R*-tree	✓	-	✓	-	-	Uncertain & Static
7.	Agarwal et al., 2009	Kd-tree	✓	-	✓	-	-	Uncertain & Static
8.	Liu et al., 2013	-	✓	-	✓	-	-	Uncertain & Dynamic
9.	Chuan-Ming & Syuan-Wei, 2015	*SW*-tree	-	✓	-	✓	-	Uncertain & Dynamic
10.	Morse et al., 2007	Quadtree	-	✓	-	✓	-	Uncertain & Dynamic
11.	Khalefa et al., 2010	-	-	✓	-	✓	-	Uncertain & Static
12.	Saad et al., 2014	-	-	✓	-	-	✓	Uncertain & Static

*** CD (Continuous Domain) *DL (Uncertainty at the dimension level) * OL (Object Level) * DL (Dimension Level)**

CONCLUSION

Today, we are faced with the challenges of realizing efficient applications that can support multi-criteria decision-making based on multi criteria data analysis of uncertain databases. To support interested researcher in realizing this object, we provide a taxonomy of the skyline processing approaches based on the method used by the existing skyline works.

REFERENCES

Agarwal, P.K., Afshani, P., Arge, L., Green, L.K., & Phillips, M.J. (2011). (Approximate) Uncertain Skylines. *Proceedings of the International Conference on Database Theory*, pp. 186-196. doi: 10.1145/1938551.1938576

Börzsönyi, S., Kossmann, D., & Stocker, K. (2001). The Skyline Operator. *Proceedings of the International Conference on Data Engineering*, pp. 421-430. doi: 10.1109/ICDE.2001.914855

Chuan-Ming, L. & Syuan-Wei, C. (2015). An Effective Probabilistic Skyline Query Process on Uncertain Data Streams. *Proceedings of the International Conference on Emerging Ubiquitous Systems and Pervasive Networks*, pp. 40-47. doi: 10.1016/j.procs.2015.08.310

Han, H., Li, J., Yang, D. & Wang, J. (2013). Efficient Skyline Computation on Big Data. *IEEE Journal: Transactions on Knowledge and Data Engineering*, 25(11), pp. 2521-2535. doi:10.1109/TKDE.2012.203

Khalefa, M.E., Mokbel, M.F., & Levandoski, J.J. (2010). Skyline Query Processing for Uncertain Data. *Proceedings of the International Conference on Information and Knowledge* Management, pp. 1293-1296. doi: 10.1145/1871437.1871604

Kossmann, D., Ramsak, F., & Rost, S. (2002). Shooting Stars in the Sky: An Online Algorithm for Skyline Queries. *Proceedings of the International Conference on Very Large Database*, pp. 275-286.

Liu, X., Yang, D., Ye, M., & Lee, W. (2013). U-Skyline: A New Skyline Query for Uncertain Databases. *IEEE Journal: Transactions of Knowledge and Data Engineering*, 25(4), pp. 945-960. doi:10.1109/TKDE.2012.33

Morse, M., Patel, J.M., & Gosky, W.I. (2007). Efficient Continuous Skyline Computation. *Elsevier Journal: Information Science*, 177(17), pp. 3411–3437. doi:10.1016/j.ins.2007.02.033

Papadias, D., Tao, Y., Fu, G, & Seeger, B. (2003). Progressive Skyline Computation in Database Systems, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 41-82. doi: 10.1145/1061318.1061320

Pei, J., Jiang, B., Lin, X. & Yuan, Y. (2007). Probabilistic Skyline on Uncertain Data. *Proceedings of the International Conference on Very Large Database*, pp. 15-26.

Saad, N.H.M., Ibrahim, H., Alwan, A.A., Sidi, F., & Yakoob, R. (2014). A Framework for Evaluating Skyline Query on Uncertain Autonomous Database. *Proceedings of the International Conference on Computational Sciences*, pp. 1546-1556. doi: 10.1016/j.procs.2014.05.140

Saad, N.H.M., Ibrahim, H., Alwan, A.A., Sidi, F., & Yakoob, R. (2016). Computing Range Skyline Query on Uncertain Dimension. In S. Hartmann, & H. Ma (Eds.), *Lecture Note in Computer Science: Vol.* 9828. *Database and Expert System Applications* pp. 377-388. Berlin, Germany: Springer-Verlag. doi: 10.1007/978-3-319-44406-2_31

Tan, K.-L., Eng, P.K., & Ooi, B.C. (2001). Efficient Progressive Skyline Computation. *Proceedings of the International Conference on Very Large Database*, pp. 301-310.