Survey on Skyline Queries with Its Algorithms and Operators

R.Prema Steffi [1], S.Sundaramoorthy [2]

1 PG Scholar, 2 Assistant Professor

Angel College of Engineering and Technology, Tamil Nadu, India

Abstract*—Spatial database uses the space related data based on their attributes. In spatial data mining it makes use of spatial databases for mining the efficient result and thus provide the Location Based Services(LBS).With spatial data mining, the subsequent skyline queries are processed by performing skyline query processing and thus finally it makes extracting those data which give out effective result for the users. For processing the skyline queries it also makes use of spatial and skyline data sets for its efficiency. Here we are going to take survey based on both skyline queries and their skyline operators. Also we have extended our survey mainly focusing on the algorithms of skyline queries.

***Index Terms*— Spatial Mining, Skyline Queries, Skyline Operator.**

I. INTRODUCTION

In a database, a Skyline is a set of tuples of information (points) which stand out among the others because are of special interest to us. In location-based services (LBS), users with mobile devices are able to make queries about their surroundings anywhere and at any time. Location-based services are booming as a major application of mobile geospatial technologies. This is processed by skyline queries based upon the spatial queries. More formally, a skyline is defined as those points which are not dominated by any other point. A point will dominate another point if it is as good or better in all dimensions and better in at least one dimension. For example, a dataset containing information about hotels; the distance to the beach and the price for each data point is recorded. Consider a two dimensional plot of the dataset, where the distance and price are assigned as X, Y axis for plotting.

The goal of the search is to find a hotel whose distance to the beach and the price are both minimum (not restricted to minimum, any other function max, join, group-by clause could be used.) [2]. The preference function in our example is "least price and bare least distance". Here the dataset may not only have single data point that satisfies both these desirable properties. With these data points in the skyline the interesting points are founded out which satisfies the dominance relation with the other points in the skyline. With these skyline based queries it can able to focus in the range based skyline queries also by performing the queries dased on some range in the LBS in the system.

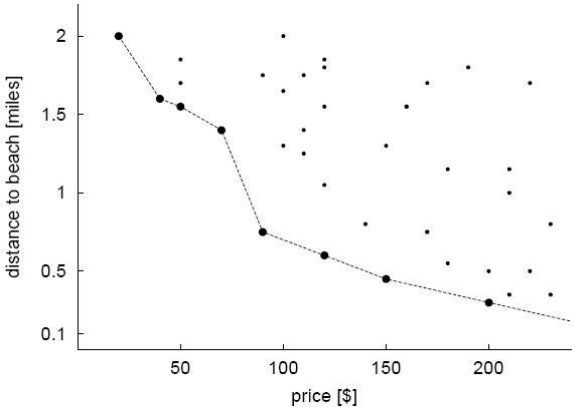

Fig1: Example of Skyline Query to find cheapest hotel and nearest to the beach

The user is accessible with a set of interesting points that partly satisfy the imposed constraints. In figure 1, the interesting points are shown. If those points dominate other in that it will be better in any of those dimensions. The interesting data-points in the dataset are considered by the least distance from the beach. With this user can able to view the Location based services with the help of skyline data points which shows the interesting point for the user to analysis the result.

II. DEFINITIONS

A. Skyline Query

Definition: Given a set of points, the skyline query proceeds a set of points (referred to as the skyline points), such that any point is not dominated by any other point in the dataset.

B. Skyline Query Processing

Definition: Given a set of points, the skyline query is examined for the interesting point in the datasets is referred as Skyline Query Processing.

C. Range Based Skyline Query

Definition: Given a set of points, the skyline query based on the range is examined for the interesting point in the datasets is referred as Range Based Skyline Query Processing.

III. SKYLINE QUERY PROCESSING

In Skyline Query Processing [1], a number of algorithms have been proposed already. In this it mainly categorized as

two: based on index and not based on index. In based on index, it will have representatives as B+Trees, to speed up the query processing and R Trees, extension to B+Tree. In not based on index, it will have the representatives as Block Nested Loop (BNL), Divide & Conquer (D&C). These algorithms are discussed in the section 4. It also has algorithms as Branch and Bound Skyline (BBS) discussed in the section 4 which gives us effective results.

IV. RANGE-BASED SKYLINE QUERY PROCESSING

Range-based Skyline Query (RSQ) Processing [1] has recently received notable attention as the location privacy issue is becoming progressively important. For privacy reasons, mobile clients incline to blur their exact locations into an uncertain range so that the service provider cannot find where they are exactly located in the region.

The service provider thus will return a superset of candidate results for every possible query point in the range. With that finally, the clients filter these results and obtain the true result by their exact locations and provide the services for the users.

The existing range-based query algorithms [1] studied only range-based kNN (RkNN) query. With this Range based Skyline Query Processing the interesting points in the skyline data sets are founded with the Location based services for the users. With this Range based Skyline Query it can thus provide the LBS for the user and thus for enabling it will process those query with privacy concerned basics.

V. SKYLINE ALGORITHMS

A. Block Nested Loop (BNL)

In the Block Nested Loop, it scans through a list of point and test each point for dominance criteria. i.e. It makes the scan as sequentially with its list of points.[2],[3]. Active list of potential skyline points seen thus far are maintained, each visited point is compared with all elements in the list. The list is suitably updated. Method does not require a precomputed index. Total work done depends on the order in which points were encountered. Method performs laid off work, that is no provision for early termination.

BNL works particularly well if the Skyline is small. The advantage of BNL is its wide applicable, since it can be used for any dimensionality without indexing or sorting the data sets. Its main troubles are the dependence on main memory (a small memory may lead to numerous iterations) and its inadequacy for progressive processing (it has to read the entire data file before it returns the first skyline point).

The sort first skyline (SFS) variation of BNL lessens these problems by first sorting the entire dataset according to a (monotone) liking function. Candidate points are inserted into the list in ascending order of their scores, as points with lower scores are possible to dominate a large number of points, thus exposé the pruning more effective. SFS exhibits progressive behavior because the presorting ensures that a point p dominating another p' must be visited before p'; hence we can immediately output the points inserted to the list as skyline points. Nevertheless, SFS has to scan the entire data file to return a complete skyline, because even a skyline point may have a very large score and thus appear at the end of the sorted list.

B. Divide and Conquer (D&C)

In Divide and Conquer, it will get median value and then divide tuples into 2 partitions. Compute skyline of each partition with the skyline points. And then merge those partitions to find out the interesting points with them. i.e It will makes the partition on the space as the subspaces, and then solve the problem in the subspaces and thus synthesize the solution in the whole space.In divide and conquer, it recursively breaks up large datasets into smaller partition.[5] It will continue till each smaller partition of the dataset fits in the main memory. With that it will compute the partial skyline for each partition using any in-memory approach and later combine these partial skyline points to form the final skyline query and to find the dominance points with those data points as shown in fig 2.

C. Nearest Neighbor Search

In Nearest Neighbor Search, assumes that a spatial index structure on the data points is available for use. It will identify skyline points by repeated application of a nearest neighbor search technique on the data points, using a suitably defined distance norm. The nearest neighbor in the data-point is as it is closest to the origin when a distance measure is assumed. i.e uses an R-tree index and performs a sequence of nearest-neighbor queries until all Skyline objects have been found. The R-tree can be viewed as an extension of the B+-tree to handle multiple dimensions [5]. B+-tree is used to classify numeric data in one dimension only. R-trees have been extensively used in spatial databases to organize points and rectangles. [5] Show excellent performance in processing interesting queries such as: Range query: return the points that are contained in a precise region. K-nearest-neighbor: given a point *p* and an integer *k* return the *k* objects closer to *p*.

D. Branch and Bound Skyline (BBS)

The Branch and Bound Skyline is mainly based on the best first nearest neighbour search [1] , this accesses the node only that has the skyline points. With this BBS will thus achieve the high I/O access for the skyline points. Any Branch-and-Bound method requires two resolutions [5]:

- How to branch: which part of the space needs to be examined next?
- How to bound: which parts of the search space can be safely removed.

In Branch and Bound skyline, [5] assume that all points are indexed in an R-tree. The mindist (MBR) = the L1 distance between its lower-left corner and the origin. Each heap entry keeps the mindist of the MBR. It will insert top level of the hierarchy. Upon visiting a node, the mindist of its entries is calculated and entries are inserted into the priority queue. If the top of the queue is a data point, it is experienced if it is dominated by any point in *S*. If yes it will be rejected, otherwise it is inserted into *S* shown in Algorithm 1[3].

Algorithm BBS (R-tree *R*)
1. *S*=∅ //*list of skyline points*
2. Insert all entries of the root *R* in the heap
3. While heap not empty

4. Remove top entry *e*
5. If *e* is dominated by some point in *S* discard *e*
6. Else // *e is not dominated*
7. If *e* is an intermediate entry
8. for each child *ei* of *e*
9. If *ei* is not dominated by some point in *S* insert *ei* into heap
10. Else // *e is a data point*
11. Insert *ei* into *S*
12. End while

End BBS

Algorithm 1: BBS algorithm

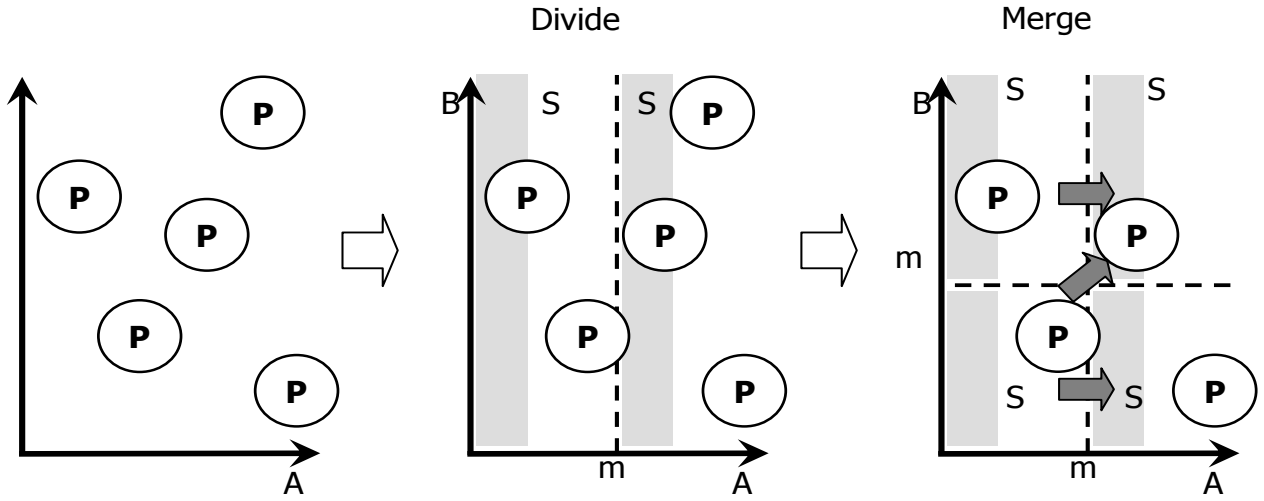

Fig 2: Example for divide & conquer in skyline

Dominance check condition: Before expanding or inserting, compare to current skylines. Before inserting an object, also check for internal objects [4].In this Branch and Bound Algorithm it makes the analysis on the basics of the minimum lower bounded rectangle for identifying the interesting points in the skyline data points. It thus will makes analysis for finding the dominance points in the skyline points.

VI. SKYLINE OPERATOR

A. SQL Extensions

In order to specify Skyline queries, [3] proposed to extend SQL's SELECT statement by an optional SKYLINE[2] of clause as follows with the extension of the SQL and thus represented for making the processing for the skyline.

```
SELECT ... FROM ... WHERE...
GROUP BY ... HAVING...
SKYLINE OF [DISTINCT] d1 [MIN | MAX | DIFF], ..., dm [MIN | MAX | DIFF]
ORDER BY ...
```

d1; :::; dm denotes the dimensions of the Skyline; e.g., price, distance to the beach, or rating [2]. MIN, MAX, and DIFF specify whether the value in that dimension should be minimized, maximized, or simply be dissimilar.

For example, the price of a hotel should be minimized (MIN annotation) whereas the rating should be maximized (MAX annotation). The optional DISTINCT specify how to deal with duplicates [2].

The specification is given below indicates the skyline data points with some criteria:

```
SELECT *
FROM Hotels
WHERE city = 'Nassau'
SKYLINE OF price MIN, distance MIN;
```

(a)

```
SELECT *
FROM Buildings
WHERE city = 'New York'
SKYLINE OF distance MIN, height MAX,
x DIFF;
```

(b)

Examples of skyline query

B. Queries Without Skyline Clause

The subsequent standard SQL queries is corresponding to previous example but without using the Skyline operator [2] and their clauses are specified as the follows and thus shows how the skyline points are viewed without its skyline clauses:

```
SELECT *
FROM Hotels h
WHERE h.city = 'Hawaii' AND NOT EXISTS(
SELECT *
FROM Hotels h1
WHERE h1.city = 'Hawaii'
AND h1.distance ≤h.distance
AND h1.price ≤h.price
AND (h1.distance < h.distance OR h1.price < h.price));
```

C. Queries With Skyline Clause

The subsequent standard SQL query is corresponding to previous example using the Skyline operator [2] and their clauses are specified as follows and thus show how the skyline points are viwed with the skyline clauses:

```
SELECT *
FROM Hotels
WHERE city = 'Hawaii'
SKYLINE OF price MIN, distance MIN;
```

With this queries with the skyline clause it can able to focus the skyline points more efficiently and thus provides the interesting points with those skyline data sets. This provides the skyline result more quickly and thus can able to view more frequently with those skyline data points.

VII. EXPERIMENTAL RESULT

In this we have done our experimental setup based on the rapidminer tool to make the analysis for the skyline queries and to find the interesting points in the skyline data sets. Here we have taken the data sets as the car parking for the scenario to find the interesting points in the skyline. In this car parking it has the 30 objects within the data sets.

With this we have taken the operating system as windows 7 and with rapidminer5 tool wee have generated the skyline points along with their interesting points for making analysis over those data points.

In this car parking data set we have presented the data for the making the analysis in the rapidminer5 and thus we have imported those data sets with the help of read excel and thus implemented with that. With this it have worked by just importing those data sets. Whiling running the process it makes the views as data view, meta data view, plot view, advanced chart view, annotation. In this it can able to view the design side and also the perspective side. While in result side it can able to view the perspective side within it.

In the data view it makes the overall view of the data sets with in it. In meta data view it displays the description of data about data within it. In plot view it makes the views by choosing the plotter within it. In this we have chosen the plotter as scatter and bar. With the scatter it makes the analysis of those data sets and thus makes the plot view as the interesting points in the skyline. With the bar it makes the analysis of those data sets and thus makes the plot view as the interesting point in the skyline as the bar representatives.

Here we have presented this as the analysis in the rapidminer with the above discussed views which are shown as the below.

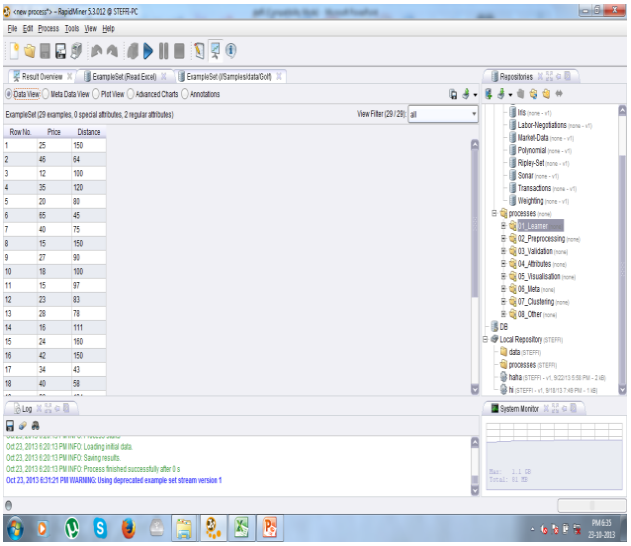

Fig 3: Data View of Skyline Points

In the figure 3, it shows the result of data view in rapidminer5 for the given car parking data sets and thus can view the data points that are used in the car parking data. Within this it can able to view the meta data view of the data sets in the car parking which thus shows the data about the data within the data sets of the skyline points.

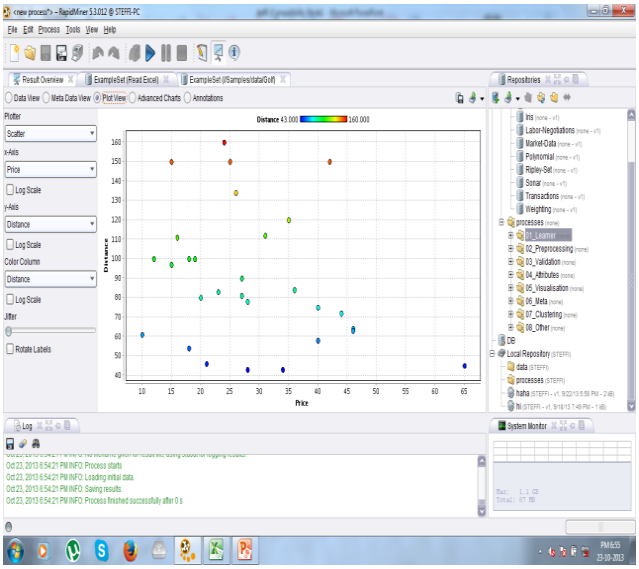

Fig 4: Plot View of Skline Points with Scatter as its Plotter

In the figure 4, it shows the result of plot view in the rapidminer5 for the given car parking data sets and thus can view the data points in the skyline with the help of plotter as scatter and their interesting points in the skyline are shown. In this it has taken the price as their x-axis and distance as their y-axis for the car parking scenario. With this it considered the interesting points as nearer distance and with their price as the lowest in the skyline.

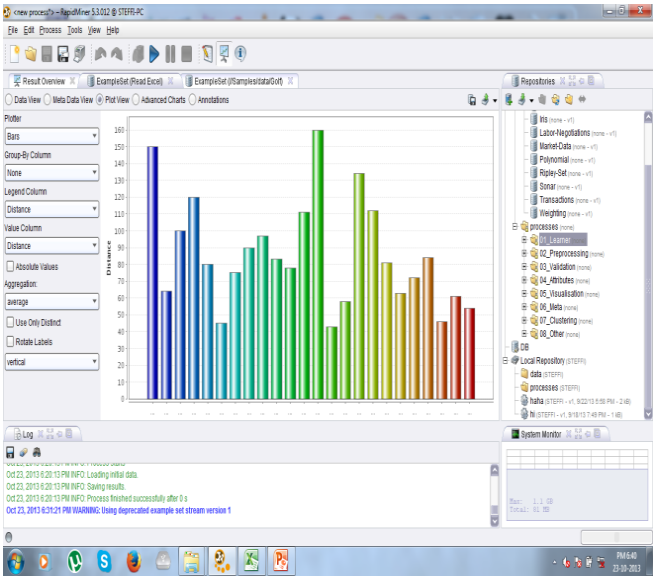

Fig 5: Plot View of Skline Points with Bar as its Plotter

In the figure 5, it shows the result of plot view in the rapidminer5 for the given car parking data sets and thus can

view the data points in the skyline with the help of plotter as bar and their interesting points in the skyline are shown. In this it makes the price as their interesting points for this bar plotter. Here also it considers the x-axis as the price and the y-axis as their distance for the car parking scenario.

VIII. CONCLUSION

As we discussed above the skyline queries are used to find out the interesting points without dominating the other points in the skyline sets. The dominance points are indicated ass the best in the skyline data points. These skyline queries are processed through the above mentioned skyline algorithms and thus been implemented by using the skyline operators. And also discussed about range based skyline query processing. We have discussed these and presented as a survey and thus focused our future work to monitor dynamic and continuous queries skyline in the mobile environment and to process those queries in efficient manner.

REFERENCES

[1]S.Borzonyi, D.Kossmann, and K. Stocker, "Range based skyline queries in mobile environment" Proc. Int'l Conf. Data Eng., pp. 421-430, 2013.

[2]S.Borzonyi, D.Kossmann, and K. Stocker, "The Skyline Operator," Proc. Int'l Conf. Data Eng., pp. 421-430, 2001.

[3]D. Papadias, Y. Tao, G. Fu, B. Seeger. "Progressive Skyline Computation in Database Systems", 2005 ACM Transactions on Database Systems.

[4]D. Papadias, Y. Tao, G. Fu, and B. Seeger, "An Optimal and Progressive Algorithm for Skyline Queries," 2003 ACM SIGMOD.

[5]"Top-k and Skyline Computation" Department of Informatics Aristotle University of Thessaloniki Fall 2009

[6]J. Xu, X. Tang, H. Hu, and J. Du, "Privacy-Conscious Location- Based Queries in Mobile Environments," *IEEE Trans. Parallel and Distributed Systems,* vol. 21, no. 3, pp. 313-326, Mar. 2010.

[7]B. Zheng, C.K. Lee, and W.-C. Lee, "Location-Dependent Skyline Query," *Proc. Int'l Conf. Mobile Data Management,* 2008.

AUTHORS

R.Prema Steffi pursuing M.E Computer Science and Engineering in Angel College of Engineering and Technology in Tirupur under Anna University Chennai. She did her B.E Computer Science and Engineering in Dr.Sivanthi Aditanar College of Engineering in Tiruchendur under Anna University Chennai. She is a member of IAENG, CSI. Her area of interest is Data Mining. Specialized area is Spatial Data Mining and Web Mining. Presented paper on "Arc Based Skyline Query" in a National Conference.

S.Sundaramoorthy working as Assistant Professor in Angel College of Engineering and Technology in Tirupur and had 2.5 year of experience. He completed his M.TECH Computer Science and Engineering in SNS College of Technology in Coimbatore under Anna University and completed his M.B.A in Bharathiyar University and B.E Computer Science in K.S.Rangasamy College of Technology under Anna University. He has published four papers in International journal and member in ISTE,IAENG. His Area of interest is Data Warehousing and Data Mining.