ISSN 23499842(Online), Volume 1, Special Issue 2(ICITET 15), March 2015
International Journal of Innovative Trends and Emerging Technologies

A SKYLINE QUERY BASED EFFECTIVE ASSESS OVER INCOMPLETE INDEPENDENT DATABASES

MRS.J.SRIDEVI[1], MR.A.SAIRAM[2], DR.C.SURESH GNANA DHAS[3]
[1]M.E .CSE (Final year), [2]Assistant Professor, [3]Professor/HOD
[2]Department of CSE, AMACE, Vadamavandal- 604410
[3]Department of CSE, VCEW, Elayampalayam-637205

ABSTRACT : Skyline computations are used to find the exact results on uncertain data sets. The purpose of skyline query is to find a set of objects that should not be dominated by any other objects and is much preferred in all dimensions. While this theory is easily applicable on certain and complete database, however, when it comes to data integration of databases where each has different representation of data in a same dimension, it would be difficult to determine the dominance relation between the underlying data. In the existing system, they proposed a framework that computes the skyline probability of datasets in uncertain dimensions using probabilistic skyline queries. A novel efficient framework was proposed to optimize the performance of skyline for large datasets in autonomous databases using two-phase skyline query algorithms IIDSkyline and also gets feedback from the authorised customers to improve quality of service & display the true data.

KEYWORDS: Skyline query, IID, Uncertain dimensions, incomplete data

1.INTRODUCION

Skyline queries introduce the notion of dominated objects under Pareto optimality. In a database of multi-dimensional objects, the skyline queries performed on the underlying database would return a set of objects that is the best trade-offs between different dimensions. For example, consider a database that contains information on hotel's room rate per night and distance to the beach. Figure 1 shows each record of the database that is represented as a point in a data space consisting of those two dimensions. A user may enter a query such as "find me hotels that are as cheap as possible and as close as possible to the beach". Now, the query itself can be understood differently depending on the user, the user might have wanted the cheapest price a hotel can offer with the hotel being not really close to the beach, or the user would have been satisfied with paying extra cost for the hotel as long as it is as near as possible to the beach [5].

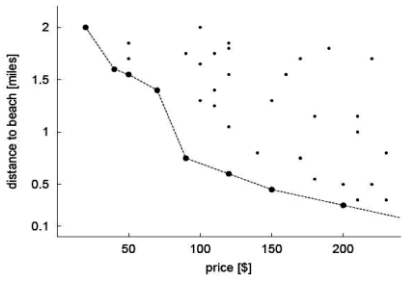

Fig 1:Skyline record of a database

Several existing works of skyline queries have been introduced in the database field. The skyline technique identifies a set of objects, in such a way that the other objects in the dataset do not dominate them. There has been no previous work that addresses the case of skyline query on uncertain data in a dimension. Existing approach to avoid the uncertainty is to use an automated translator or perform digital curation to reformat data from one representation to another. This approach has such an obvious drawback, in that the transformation of data does not guarantee that the combined, transformed data are meaningful, and thus, performing skyline queries on the underlying data would as well incur inaccurate result of skyline objects. Although there has been research done on handling uncertainty in skyline query processing, much of it hold onto the assumption that the occurrence of uncertainty in a dimension would mean all values under that dimension is represented as a continuous range.

The work by Khalefa [3][4] has also introduced an interesting concept of probabilistic skyline but in a different context than the previous works. This work was influenced by existing works at the time where these works are only limited to the simpler case of nearest-neighbor queries. Although this work is focusing

on uncertainty in continuous domains where the uncertain data is represented as continuous range of values, however, their work is restricted to uncertain dimensions with the underlying dimensions having all values in the form of real intervals.

Properties and its Implications

The skyline dominance relation is following a practical and flexible preference model as a strict partial order, which formalizes and generalizes the intuitive concept of an ordering, sequencing, or arrangement of the elements of a set. Strict partial order is not reflexive, transitive, and therefore asymmetric binary relation. More good properties and scalability on preference aggregations have theoretically derived from such powerful concepts in artificial intelligence and database communities. There is an increasing number of interactive applications that can utilize skyline queries, such as the Hotel examples described above. In such applications, e.g., a hotel search in the limited bandwidth client, the objective is not to return all skyline results but to progressively show users a big picture. This is challenging and the skyline processing algorithms should meet some of the following criteria. They are efficiency, correctness, completeness, fairness, scalability, dynamic, progressiveness etc.,

All above aspects pose major challenges for a novel skyline algorithm. That is why they are main themes of the skyline query in the database research community and Pareto optimization in both theoretical and experimental senses.

Furthermore, it is almost impossible to answer the query of how much cheaper and nearer a hotel should be. With skyline query, it will retrieve all hotels that are either cheap or near to the beach (or both, if possible) without there being other hotels that are cheaper and nearer to the beach. However, performing skyline queries on autonomous databases is not as straightforward as previously discussed. As more and more data becomes accessible via web servers, information has to be efficiently retrieved from these autonomous databases. In autonomous databases, incompleteness and uncertainty could not be avoided where the integration of data from various heterogeneous schemas is essential. It has been satisfied with paying extra cost for the hotel as long as it is as near as possible to the beach.

Uncertainty in independent databases is not surprising as it can arise in a variety of scenarios. One of the many scenarios can be seen when organizations attempted to share data stored between different databases, where these databases are individualistically developed and maintained to cater the needs of a single organization. The data can be exchanged between underlying databases could be difficult due to not only because of the dissimilarities in the representation of data, but also due to the incomplete data between the databases. [2].

Most of the approaches do not capture well the nature of autonomous databases, where imperfections of data caused by ambiguous values are inherent in today's real world application. To overcome these limitations, we present IIDSkyline, a framework for progressively evaluating skyline queries over incomplete independent databases. To avoid transforming data into meaningless information during data integration in autonomous databases, IIDSkyline instead directly performs skyline queries on integrated data as presented. Without loss of generality, we adopt the assumption that incompleteness during data integration does not occur. Incompleteness may occur due to several reasons such as incomplete data entry, or dissimilarities in schemas. Overall, our contributions are briefly described as follows:

- IIDSkyline is the first framework that can evaluate skyline queries over data with uncertain dimensions. Subsequently, it is suitable for performing skyline queries on uncertain and incomplete autonomous databases, without having to pull the data in order to conform them to a specific form.
- It defines the concept of uncertain dimensions as well as dominance relation that is able to capture dominance between data in uncertain dimensions.

2.Related work

There are many existing works of skyline queries was proposed in database and data mining field. Skyline computations are used to identify the most dominant result, which should not be dominated by any other result. Pei et al. [10] was proposed two algorithms namely bottom up and top down, for p-skyline computations and it is also extended by Bohm et al. ,they proposed two algorithms priority and indexed for continuous distributions, but it is limited to two dimensional datasets.

Khalefa et al. [4] has also focused on uncertainty in continuous range of values, however it is limited to values in the form of real intervals. Recently, skyline has attracted extensive attention and many algorithms are proposed. A set of skyline algorithms, such as Bitmap, NN [6], BBS [7], SUBSKY, and ZBtree, utilize indexes to reduce the explored data space and return skyline results. However, because of the prohibitive pre-computation cost and space overhead to cover the attributes involved in skyline on big data, index-based algorithms have serious limitations and the used indexes can only be built on a small and selective set of attribute combinations. A set of more scalable and practical skyline algorithms, such as divide and conquer [1], BNL [1], SFS, and LESS, do not require preprocessing steps or data-structures. But they have to scan the entire table at least once, which will incur high I/O cost on big data. BFS algorithm is essentially a best-first search in an R-tree, guided by the upper bound skyline probabilities of sub regions. This method can restrict the search space as the sub-regions that affect top-k skyline objects with the highest probabilities [9]. The framework IIDSkyline was introduced in this paper is both efficient and faster than the naive algorithm.

3.Preliminaries

This section defines the concept of evaluating skyline queries and its dominance relation in uncertain dimensions.

Uncertainty: Managing uncertain data has gained a lot of attention, as near-infinite amounts of automatically collected data have become available. The data inherently have various levels of uncertainty. This phenomenon incurs a new challenge of treating uncertainty as a key factor in data management. In particular, we consider an uncertainty model in which each object employs a confidence value for both alternative and maybe uncertainty. Our retrieval model is also related to the probabilistic database that annotates Boolean query results with probability values. The confidence or probability notions associated with uncertain data can be used [8].

To support advanced queries, i.e., top-k queries and skyline queries. Specifically, our work aims to support the skyline queries for uncertain data with maybe confidence. The environment cannot be efficiently supported with the existing algorithm based on the alternative uncertainty model.

Skyline queries: Given a set of objects o, an object $x \in o$ is an skyline object, it should not be dominated by any other object $y \in o$ that dominates x.

Dominance Relation: For two d-dimensional objects $x = (x_1, x_2, \ldots x_d)$ and $y = (y_1, y_2, \ldots y_d)$, x is said to dominate y (formally written as $x<y$, where we assume less is much preferred) if $x_i <= y_i$ and $x_j < y_j$.

4.IIDSkyline Framework

IIDSkyline is a general framework for evaluating skyline queries over incomplete and independent databases. It is mainly used to avoid transforming data into meaningless information during data integration in the independent databases. Mostly incompleteness may occur due to dissimilarities in schema representation during the data integration and incomplete data entry. However, it is not easy to perform skyline queries in incomplete independent databases. It is difficult to overcome the challenges arise from integration process, because the data can be integrated from different schemas. It consists of two phases isolation and Extraction.

The IIDSkyline architecture diagram provides a framework for computing exact result in uncertain dimensions. The sources of our dataset are from autonomous databases, where each database has different scheme. It has been integrated to form a global schema. Once the user has given a query, it will enter into the global schema to find the result. Then it partitions the dataset into different groups. For each group, it applies range reduction to prune out dominated objects using the threshold value and it performs the probability computation to find the candidates of skyline objects. The main purpose of virtual points is to reduce the number of points in the candidate skyline list. The IIDSkyline algorithm introduces the concept of shadows skylines that works together with virtual points to alleviate the problem of storing and comparing all input data. The main idea is instead of storing all points in each group, it stores the skyline set of points not found in the local skyline list. Moreover, the candidate skylines need to be compared only against points in the shadow skyline rather than all points to perform the exact query result.

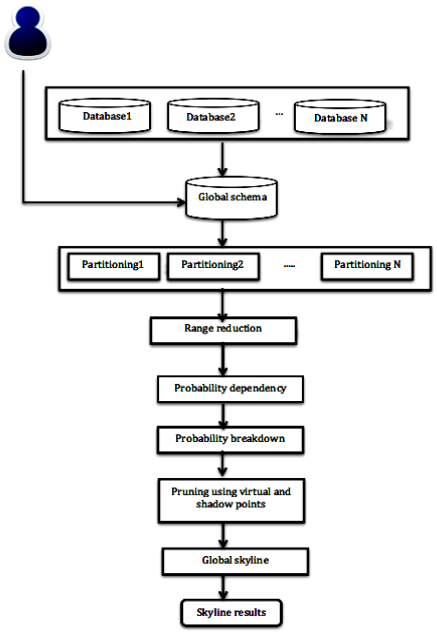

Fig 2 IIDSkyline framework

4.1 Object Isolation

This phase performs an elimination process to isolate most dominant objects and eliminate the unwanted objects. It is mainly used to reduce the number of comparisons between all the skyline points. It is impossible to integrate all the database with different schema directly and it is quite expensive. To avoid this, we propose three methods distinctive partitioning, range reduction and selection of the best result

I. Dataset Grouping and partitioning:

Datasets consist of all of the information gathered during a survey, which needs to be analyzed. In the algorithm 1, initially all the datasets are grouped together to form a global schema, then it partition the dataset of objects into distinct groups depending on the form that each object applies to, and then applying several different skyline techniques to retrieve the local skyline in every group. After dominance relation in each group has been determined and a set of skyline objects has defined is retrieved from each group, then it combine those retrieved objects into a global group and perform a dominance relation technique and a more refined probability calculation that is able to cater the objects that has different characteristics in each dimension. Partitioning is used to categorize the dataset into different groups before any skyline dominance relations are performed.

Algorithm 1: IIDSkyline (UD : uncertain dimension data set , H: threshold , E : tolerance ,DG : Dataset Group,IG : Integrated Group)

1. for each object O∈UD do
2. Apply Distinctive partitioning
3. for each group G_i∈DG , where (1 ≤i≤ n) do
4. for each object Y∈G_i do
5. Apply Range Reduction,
6. If upper limit Y< H then
7. Remove object and continue to next
8. object y
9. Apply probability dependency
10. **if** upper limit Y <H **then**
11. Remove W and continue to next object Y
12. **while** upper limit Y>E **do**
13. Apply Probability Breakdown
14. for each group G_i∈DG , where (1 ≤i≤ n) do
15. for each object Y∈G_i do
16. **if** upper limit Y >H then
17. Add W to IG

It also assume that an object has a uniform probability distribution of being anywhere in its uncertainty range, as well as for all dimensions, minimum values are preferred for our skyline computations.

II. Range reduction

Having groups of objects with different characteristics, different dominance relation techniques and skyline probability computations are needed to cater each group based on the dimensions. In the first group, where it satisfies the form of the conventional dominance relation technique is sufficient enough to be implemented to this group since it is such a straightforward method without having to take into account the problem of continuous real range. The conventional dominance relation states that in a dataset, skyline objects are those objects that are not dominated by any other objects. In this ,it calculates the upper limit of an object, if it is less than the threshold value, it discards the

object. In order to compute the better probability precision, it split the ranges into segment.

III. Selection of the upper result

All the objects that survived the filtering process of their own group are now considered as the candidates of skyline objects. These objects however have to go through another more refined filtering process, where they now will be compared to different groups in order to be finally accepted as the true skyline objects.

In this algorithm 2, it performs the continuity correction to compute the exact result because it is challenging to perform dominance relation when they objects overlap. It performs the probability dependency and breakdown to evaluate candidate skyline.

Algorithm 2: Selection of the best result

1. for each object Y ∈IG do
2. Apply Range Reduction with Continuity
3. Correction
4. **If** upper limit Y < H **then** Remove Y and
5. continue to next object W
6. Apply Probability Dependency with
7. Continuity Correction
8. **If** upper limit Y < H **then** Remove W and
9. continue to next object Y
10. **while** upper limit Y >E **do**
11. Apply Probability Breakdown with Continuity Correction
12. If upper limit Y < H then Remove object and continue to next object W
13. Add Y to S
14. Return S

4.2 Extraction

This phase employs two new concepts, namely, virtual points and shadow skylines to enable efficient execution of skyline queries over incomplete data. It is used to alleviate the problem of storing and comparing all input data.In the previous phase ,it discards the bad objects. In order to obtain the accurate result,it is necessary to consider the bad objects also.

Virtual points:

The main purpose of *virtual points* is to reduce the number of points in the *candidate* skyline list. The main idea is that a point *X* in a bucket *Ni* can be used to reduce the number of *local* skylines in each list group *Nj* where $i = j$. By doing so, the number of *local* skyline at each bucket can be reduced, and hence, the number of *candidate* skylines can be reduced significantly.

Shadow points:

The concept of *shadow skylines* that works together with *virtual points* to alleviate the problem of storing and comparing all input data. The main idea is that we do not need to store all points in each bucket, instead, we only need to store the skyline set of points not found in the local skyline list.

Algorithm 3 :*Global* Skyline

1: **for each** point $Q \in$ *Candidate Skyline* where *P* and *Q* are *comparable* **do**
2: **if** *P* dominates *Q* **then**
3: Delete *Q* from *Candidate Skyline* list
4: *Insert Virtual Point* (*P*, Node *N* of *Q*)
5: **else if** *Q* dominates *P* **then**
6: *Insert Virtual Point* (*Q*, Node *N* of *P*)
7: **end if**
8: **end for**
9: **if** *P* is not dominated by any point, **then** Insert *P* in *Candidate Skyline* list
10: **Procedure** Update global Skyline ()
11: **for each** pair of *comparable* points $P \in$ *Global Skyline* and $Q \in$ *Candidate Skyline* **do**
12: **if** *P* dominates *Q* OR *Q* dominates *P*, **then** Mark the dominated point
13: **end for**
14: Delete all marked points from *Candidate Skyline* and *Global Skyline* lists
15: **for each** point $P \in$ *Global Skyline* **do**
16: **for each** node *N* with comparable bitmap to *P* and a true *updated* flag **do**
17: **if** any point in *N shadow skyline* list dominates *P*, **then** delete *P* from the *Global Skyline* list
18: **end for**
19: **end for**
20: **for each** point $Q \in$ *Candidate Skyline* **do**
21: **for each** node *N* with comparable bitmap to *Q* **do**
22: **If** any point in *N Shadow Skyline* list *Skyline* dominates *Q*, **then** delete *Q* from the *Candidate* list

23: **end for**
24: **end for**
25: *Global Skyline* ← *Global Skyline* ∪ *Candidate Skyline*
26: set all *updated* flags to *false*

In the algorithm 3,it aims to insert those *local* skyline point into the *candidate skyline* list. Basically, we compare *P* against all *comparable* points in the *candidate skyline list* (i.e., those points that have common complete dimensions with *P*).

For each comparable point *Q*, we check if either *P* or *Q* dominates the other. If it is the case that *P* dominates *Q*, we delete *Q* from the *candidate skyline* list and insert *P* as a *virtual point* in the *Q*'s node. For the case where *Q* dominates *P*, we just insert *Q* as a *virtual point* in *P*'s node. Finally, if no point in the *candidate skyline* list dominates *P*, we insert *P* into the *candidate skyline* list. After that, we have to validate the global skyline, It is important to note that throughout Algorithm 3, deleting a point from either the *candidate* or the *global* lists indicates that the point is stored in the *shadow skyline* list of its corresponding node. Finally it computes the most dominated object and it will return the skyline result to user.

5.Conclusion and Future work

A two phase query algorithm was proposed to identify the most probable top-k skyline tuples in terms of data dependency and also enhance the accuracy for better results In this, it first defined the skyline probability, based on the probability it identifies top-k skyline tuples with the highest probability using IIDSkyline framework. It gradually and efficiently calculates skyline probability of an object to be considered, as skyline objects as well as eliminating unwanted objects. First, it partitioned the objects into different groups so that each group now will consist of objects with the same form of structure and different dominance relation techniques will be used to compute the skyline objects for each group. Then it combined together all the groups that have been partitioned in the previous phase and performed a final probability computation using virtual and shadow points. It demonstrates that the proposed algorithms are significantly faster than a naive algorithm by orders of magnitude. In future work, the freshness of the query result for dynamic objects is a very big challenging issue. It also extends this work to road network environments.

6.References

[1] Dongwon Kim, HyeonseungIm, and Sungwoo Park, "Computing Exact Skyline Probabilities for Uncertain Databases", IEEE, VOL. 24,2012
[2] Nurul Husna Mohd Saad, Hamidah Ibrahim, Ali Amer Alwan, Fatimah Sidi1, Razali Yaakob**, "**A Framework for Evaluating Skyline Query over Uncertain Autonomous Databases", Elsevier, Volume 29, 2014, Pages 1546–1556
[3] M. E. Khalefa, M. F. Mokbel, and J. J. Levandoski, " Skyline query processing for incomplete data"
[4] M. E. Khalefa, M. F. Mokbel, and J. J. Levandoski, " Skyline query processing for uncertain data", Proceedings of the Conference on Information and Knowledge Management, 2010, 1293-129
[5] Stephan Borzsonyi, Donald Kossmann, Konrad Stocker, "The skyline operator", ICDE, 2001
[6] Ling Hu; Wei-Shinn Ku; Bakiras, S.; Shahabi, C. ,"Spatial Query Integrity with Voronoi Neighbors" Volume: 25 , Issue: 4 DOI: 10.1109/TKDE.2011.267 , Publication Year: 2013
[7] Dimitris Papadias, Yufei Tao, Greg Fu, Bernhard Seeger, "An optimal and progressive algorithm for skyline queries", SIGMOD, 2003
[8] Mohamed A. Soliman, Ihab F. Ilyas, Kevin Chen-Chuan Chang, "Top-k query processing in uncertain databases", ICDE, 2007
[9] Hyountaek Yong, Jongwuk Lee, Jinha Kim, Seung-won Hwang, "Skyline ranking for uncertain databases", IEEE
[10] J. Pei, B. Jiang, X. Lin, and Y. Yuan, "Probabilistic Skylines on Uncertain Data," Proc. Int'l Conf. Very Large Data Bases (VLDB),pp. 15-26, 2007.