Volume 3, Issue 7, July 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Adaptive Algorithm for Computing Subspace SKYLINE Queries Over Distributed Uncertain Data

T.Sunitha* *PG Scholar, Department of CSE P.B.Collge of Engineering, India*

L.Indu *Assistant Professor, Department of CSE P.B.College of Engineering, India*

S.Anbu *Professor & HOD, Department of CSE P.B.College of Engineering, India*

Abstract-The skyline queries has received considerable attention from the database community, due to its importance in many applications including multicriteria decision making, preference answering, and so forth. Data collected from different sources in distributed locations exhibit a kind of uncertainty. A skyline query retrieves the set of non-dominated data points in a multi-dimensional dataset. For efficient subspace skyline processing, a notion of subspace dataset is derived, which contains all data elements that are necessary to answer a skyline query in any arbitrary subspace. Then the problem of distributed skyline computation is studied and proposed an adaptive algorithm towards retrieving the global skyline tuples from all the distributed local sites with minimum communication cost.

Keywords - Skyline, distributed database, uncertain data.

I. Introduction

Data's are increasingly stored and processed distributively[1] [2], as a result of the wide deployment of computing infrastructures and the readily available network services. More and more applications collect data from distributed sites and derive results based on the collective view of the data from all sites. Examples include sensor networks, data integration from multiple data sources[7], and information retrieval from geographically separated data centers. In the aforementioned application domains, it is often very expensive to communicate the data set entirely from each site to the centralized server for processing, due to the large amounts of data available nowadays and the network delay incurred, as well as the economic cost associated with such communication. Fortunately, query semantics in many such applications rarely require reporting every piece of data in the system. Instead, only a fraction of data that are the most relevant to the user's interest will appear in the query results. Skyline queries help users make intelligent decisions over complex data[3], where different and often conflicting criteria are considered. Such queries return a set of interesting data points that are not dominated by any other point on all dimensions. Distributed skyline computation[17], over uncertain data has many important applications. For instance, consider the stock market application where customers may want to select good deals (transactions) for a particular stock over all the distributed stock exchange centers. A deal is recorded by two attributes (price, volume) where price is the average price per share in the deal and volume is the number of shares. such scenario, before making trade decisions, a customer may want to know the top deals over all the distributed local sites. However, such a top deals are often difficult to find, especially since the user typically has no information on the database content. Moreover, complex decision making on real world data usually involves several dimensions of interest. As an alternative, the skyline query returns all offers which may be of interest. Therefore, a set of deals recorded in the database may be treated as a set of uncertain elements and some customers may only want to know "top" deals (skyline) among the entire deals over distributed sites; and thus we have to take the uncertainty of each deal into consideration. This is a scenario of distributed skyline queries over uncertain data. Consider, for instance, a database containing information about hotels. Assume a user is looking for hotels at a specific location that are as cheap as possible and as close as possible to the beach. In this case, it is not obvious whether the user would prefer a hotel that is very close to the beach but more expensive than others or rather a hotel that is very cheap but farther away from the beach. The skyline set contains all hotels that are not worse than any other hotel based on all criteria, without requiring a scoring function that defines the relative importance of the different criteria. Thus, the skyline set contains all tuples that represent the best trade-offs between the different criteria. Figure 1 shows an example, where each point represents a hotel with price per night and distance to the beach as coordinates; hotels p3, p1, p5 are in the skyline set.

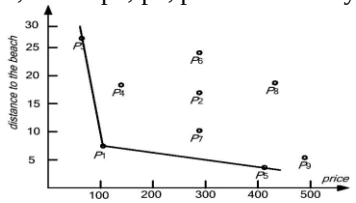


Fig 1: A General Skyline Example

The same considerations also hold for a variety of applications (e.g., electronic marketing places or real-estate databases for houses), where the user is interested in mobiles, cars, houses, or other products. The user might, for instance, be looking for a new mobile supporting all fancy features such as Wi-Fi, high resolution camera and display, but with long talk/standby times and still minimum weight and size. Likewise, a user who is interested in buying a car wants to find a good trade-off between minimum mileage, minimum age, and minimum price.

II. Literature Survey

L.Chen et al proposed a skyline of a multidimensional point set is a subset of interesting points that are not dominated by others[8]. In this project, they investigate constrained skyline queries in a large-scale unstructured distributed environment, where relevant data are distributed among geographically scattered sites. They first propose a partition algorithm that divides all data sites into incomparable groups such that the skyline computations in all groups can be parallelized without changing the final result. They then develop a novel algorithm framework called Pad Skyline for parallel skyline query processing among partitioned site groups. They also employ intra group optimization and multi filtering technique to improve the skyline query processes within each group. In particular, multiple (local) skyline points are sent together with the query as filtering points, which help identify unqualified local skyline points early on a data site. This paper focuses the computation in unstructured environment. *W. Zhang et al said s*kyline computation has many applications including multi-criteria decision making [6]. In this project, the problem of efficient processing of continuous skyline queries over sliding windows on uncertain data elements regarding given probability thresholds were studied. They first characterize what kind of elements we need to keep in our query computation. Then they have shown the size of dynamically maintained candidate set and the size of skyline. Efficient techniques to process a continuous, probabilistic skyline query were developed. Finally, extend the techniques to the applications where multiple probability thresholds are given or we want to retrieve "top-k" skyline data objects.

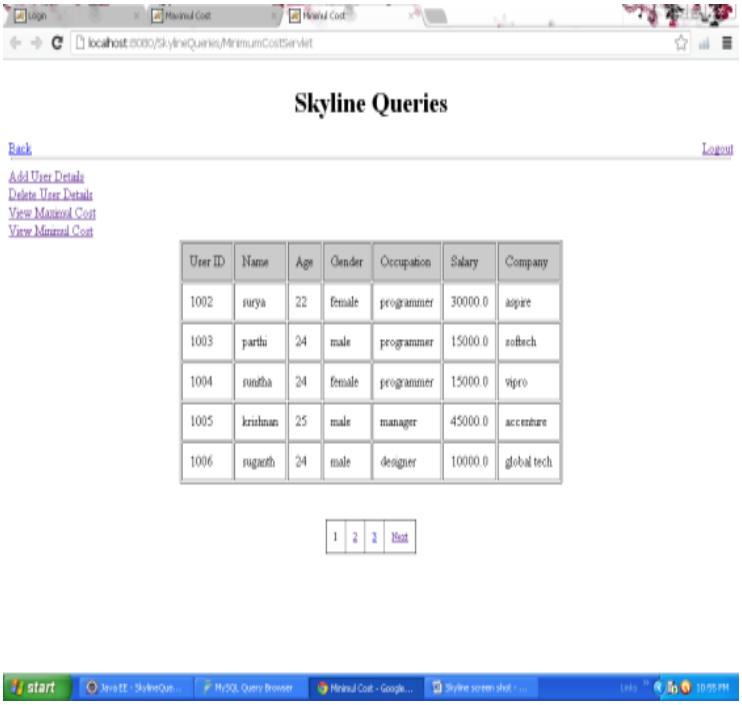
III. The Proposed Approach

An important problem in this scenario is to retrieve the minimum skyline tuples from all the local node with minimum communication cost. Two communication- and computation-efficient algorithms are proposed to retrieve the qualified subspace skylines from local node. The proposed algorithm, called subspace data set, computes the global skylines in a distributed, uncertain environment with low bandwidth cost. Furthermore, an enhance version of algorithm is also proposed to further speedup the query efficiency and reduce the communication cost.

IV. Related Works

A. Analysis of Subspace data

Given a set of sites that store locally relevant data, the skyline sets are the same if the query is evaluated on (i) The union of the local datasets or (ii) First on each dataset in separate and then once more on the union of the result sets. The query can be processed in a distributed fashion, where each queried node processes a skyline query based on the data that are stored locally and reports back its local skyline set.



The performance of a distributed skyline approach is analyzed using query routing, which is the process of deciding which node may contribute to the skyline set and hence which node should be queried in the subsequent round.

B. Efficient Retrieval of Subspace Skyline

The Efficient retrieval of subspace skyline is to locally evaluate as many parts of the query as possible. However, accurate skyline computation over widely distributed data, demands that all data is taken into account, since even a single point neglected could be part of the skyline and, thus, prune out other points already processed. Thus, each local node needs to collect from the associated node only the skyline points of all subspaces. Each local site individually processes a subspace skyline request and transmits the results to the query initiator. Some local skyline points may not belong to the global skyline. Thus, the query initiator needs to collect the results from all super-peers and merge them by discarding dominated points. In order to avoid the transmission of all data, need to calculate a subset of the original dataset that contains all the skyline points for any subspace. After the initiator has executed a local subspace skyline computation and has collected the local subspace skyline result set of all other local node, initiator merges the local result sets of the individual local site to one global result set.

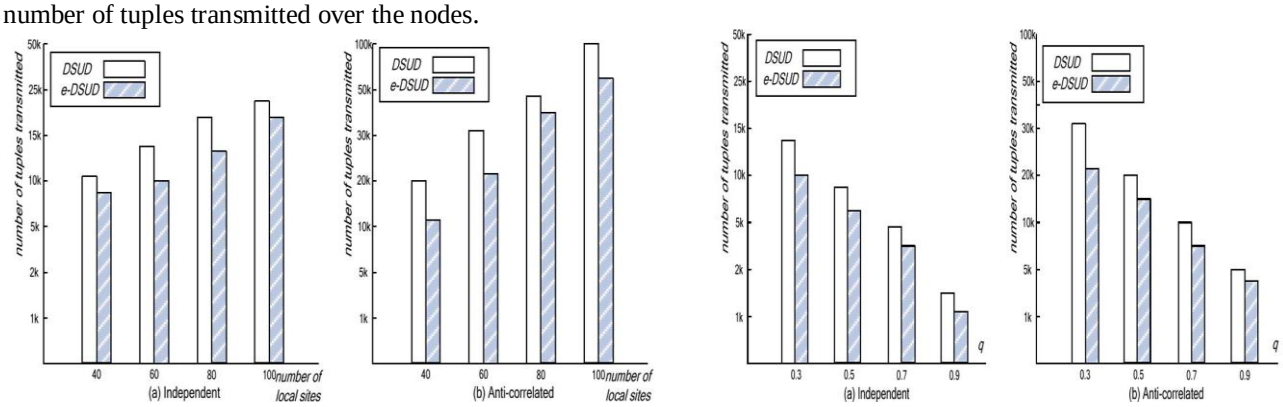
C. Optimized feedback mechanism

The main categories are determined by (ii) how are results propagated back to the query initiator, and (ii) whether filter points are used to distinguish relevant and irrelevant data and paths. Each Local site calculates its own local subspace skyline result. These local subspace skyline results have to be merged into a global result set.

Instead of forwarding all results back to user, each local node merges the results of its neighbouring nodes, and forwards the merged result back to the user. Thus the transferred data is reduced and it is time-consumed.

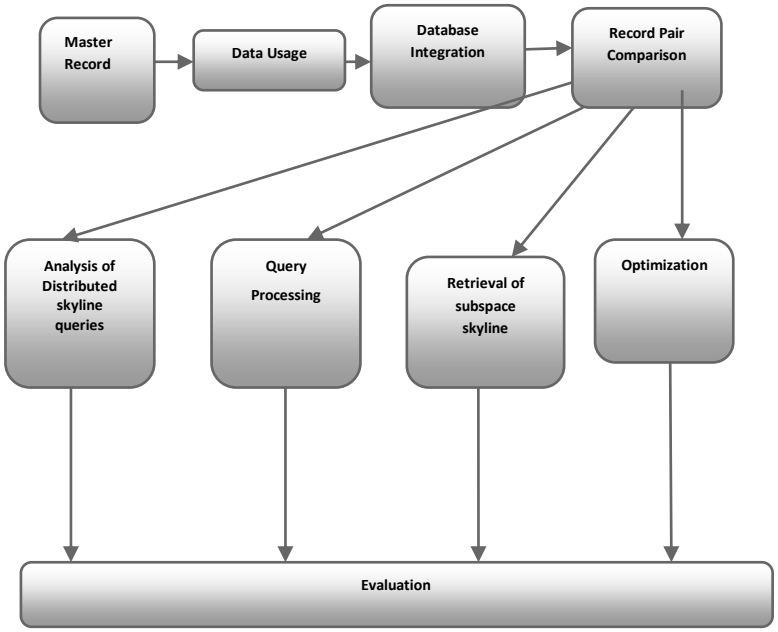
D. Performance Evaluation of Algorithm

The efficiency and progressiveness of the proposed algorithm and its Enhanced version of algorithm are evaluated. Both the synthetic and real data sets are used for the evaluation. The efficiency of the algorithms are evaluated in terms of bandwidth consumption against dimensionality, number of local databases. Also the progressiveness of the methods under different location distributions is evaluated. Specifically, bandwidth consumption is measured by the number of tuples transmitted over the nodes.



Progressiveness, on the other hand, is evaluated by measuring the bandwidth consumption cost and CPU runtime as a function of the number of qualified skyline tuples received .

E. Architecture Overview



IV. Conclusion

The proposed algorithm focuses the subspace skyline queries for distributed data. The significant communication cost could be saved by analysing the difference between the global skyline and its approximate value. Also demonstrate how to alleviate the computation burden at each distributed node so that communication and computation efficiency are achieved simultaneously. Extensive experiments have been conducted to verify that the techniques can process skyline queries over distributed uncertain data both communication- and computation-effectively.

In this paper an arbitrary horizontal partitioning is focused, where a local site has all the attributes but stores only a subset of the entire tuples. In future this work can be extended to process ad-hoc skyline query.

References

[1] F. Li, K. Yi, and J. Jestes, "Ranking Distributed Probabilistic Data," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '09),June 2009.

[2] W. Zhang, X. Lin, Y. Zhang, W. Wang, and J. Yu, "Probabilistic Skyline Operator over Sliding Windows," Proc. IEEE 25th Int'l Conf. Data Eng. (ICDE '09), pp. 305-316, Mar. 2009.

[3] S. Borzsonyi, D. Kossmann, and K. Stocker, "The Skyline Operator," Proc. 17th Int'l Conf. Data Eng. (ICDE '01), pp.421-430, 2001.

[4] X. Lian and L. Chen, "Monochromatic and Bichromatic Reverse Skyline Search over Uncertain Database," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 213-226, 2008.

[5] X. Lian and L. Chen, "Probabilistic Ranked Queries in Uncertain Databases," Proc. Int'l Conf. Extending Database Technology (EDBT '08), pp. 511-522, 2008.

[6] M. Hua, J. Pei, W. Zhang, and X. Lin, "Ranking Queries on Uncertain Data: A Probabilistic Threshold Approach," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2008.

[7] D. Papadias, Y. Tao, G. Fu, and B. Seeger, "An optimal and Progressive Algorithm for Skyline Queries," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 467-478, 2003.

[8] L. Chen, B. Cui, and H. Lu, "Constrained Skyline Query Processing against Distributed Data Sites," IEEE Transaction on Data and Knowledge Eng., vol. 23, no. 2, pp. 204-217, Feb. 2011.

[9] B. Cui, L. Chen, L. Xu, H. Lu, G. Song, and Q. Xu, "Efficient Skyline Computation in Structured Peer-to-Peer Systems," IEEE Trans. Knowledge and Data Eng., vol. 21, no. 7, pp. 1059-1072, July 2009.

[10] S. Wang, Q.H. Vu, B.C. Ooi, A.K.H. Tung, and L. Xu, "Skyframe: A Framework for Skyline Query Processing in Peer-to-Peer Systems," The VLDB J., vol. 18, pp. 345-362., 2009.

[11] I. Sharfman, A. Schuster, and D. Keren, "A Geometric Approach to Monitoring Threshold Functions over Distributed Data Streams," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2006.

[12] N. Dalvi and D. Suciu, "Efficient Query Evaluation On Probabilistic Databases," The VLDB J., vol. 16, no. 4, pp. 523-544, 2007.

[13] K. Deng, X. Zhou, and H.T. Shen, "Multi-Source Skyline Query Processing in Road Networks," Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE '07), 2007.

[14] A. Vlachou, C. Doulkeridis, and Y. Kotidis, "Angle-Based Space Partitioning for Efficient Parallel Skyline Computation," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2008.

[15] L. Zhu, Y. Tao, and S. Zhou, "Distributed Skyline Retrieval with Low Bandwidth Consumption," IEEE Trans. Knowledge and Data Eng., vol. 21, no. 3, pp. 384-400, Mar. 2009.

[16] W.-T. Balke, U. Guntzer, and J.X. Zheng, "Efficient Distributed Skylining for Web Information Systems," Proc. Ninth Int'l Conf. Extending Database Technology (EDBT '04), pp.256-273, 2004.

[17] A. Vlachou, C. Doulkeridis, Y. Kotidis, and M. Vazirgiannis, "Skypeer: Efficient Subspace Skyline Computation over Distributed Data," Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE '07), pp. 416-425, 2007.

[18 J. Pei, B. Jiang, X. Lin, and Y. Yuan, "Probabilistic Skylines on Uncertain Data," Proc. Int'l Conf. Very Large Data Bases (VLDB),2007.

[19] X. Ding and H. Jin, "Efficient and Progressive Algorithms for Distributed Skyline Queries over Uncertain Data," Proc. IEEE 30th Int'l Conf. Distributed Computing Systems (ICDCS), pp. 149-158, 2010.

[20] L. Chen, B. Cui, H. Lu, L. Xu, and Q. Xu, "iSky: Efficient and Progressive Skyline Computing in a Structured P2P Network," Proc. IEEE 28th Int'l Conf. Distributed Computing Systems (ICDCS), 2008.

