Grid-Based Probabilistic Skyline Retrieval on Distributed Uncertain Data

Xiaowei Wang and Yan Jia

School of Computer, National University of Defense Technology, Hunan
Changsha, 410073, China
gfkdwxw@yahoo.com.cn

Abstract. The skyline queries help users make intelligent decisions over complex data. It has been recently extended to the uncertain databases due to the existence of uncertainty in many real-world data. In this paper, we tackle the problem of probabilistic skyline retrieval on physically distributed uncertain data with low bandwidth consumption. The previous work incurs sharply increased communication cost when the underlying dataset is anti-correlated, which is the typical scenario that the skyline is useful. In this paper, we propose a knowledge sharing approach based on a novel grid-based data summary. By sharing the data summary that captures the global data distribution, each local site is able to identify large amounts of unqualified objects early. Extensive experiments on both efficiency and scalability have demonstrated that our approach outperforms the competitor.

1 Introduction

Skyline queries help users make intelligent decisions over multiple dimensional data when different and often conflicting criteria are considered. Specifically, a skyline query returns objects not dominated by any other objects. An object o is said to dominate o', if o is not worse than o' in every single dimension, and better than o' in at least one dimension.

In many real-world applications, massive data is integrated from a large number of data sources, and assembled at query time. Meanwhile, with large scale emerges data uncertainty as a factor that can not be ignored. Consider a web-based hotel recommendation system which integrates from multiple data sources, a common feature of such kind of systems is to allow users to rate the hotels based on their consumer experiences. The favorite rate associated with each hotel can be regarded as the existential probability because it represents the probability that the hotel occurs exactly as claimed in the advertisement. The system is supposed to recommend confidential hotels according to multi-criteria based ranking, such as low price and more bedrooms. Such kind of problems can be modeled as probabilistic skyline query, that is, the system returns the hotels which are not dominated by any other hotels on “price” and “bedrooms” above a confidence level.

This work concentrates on probabilistic skyline query on distributed uncertain data. Our goal is retrieving the global probabilistic skyline objects in a communication

J. Xu et al. (Eds.): DASFAA Workshops 2011, LNCS 6637, pp. 538–547, 2011.

efficient way. The only existing probabilistic skyline algorithm on distributed uncertain data (named e-DSUD) [1] utilizes a priority-based scheme to compute the global probabilistic skyline progressively. However, their approach is limited in communication efficiency because of the lack of global data distribution in local sites. We tackle this problem by a data summary sharing approach which reduces the communication cost by pruning unqualified local skyline objects early. In summary, our contributions are as follows:

- We propose a grid-based data summary which captures the distribution of an uncertain dataset in a data independent manner. Based on the data summary proposed, the general framework of our algorithm is proposed to seemly integrate the knowledge sharing and early pruning process to e-DSUD.
- A big challenge of our algorithm is sharing the data summaries with low communication cost. We tackle this problem by further optimization which significantly reduces the information need to be transferred for a data summary.
- We conduct comprehensive experiments on both real and synthetic datasets. The results have shown that our approach outperforms the existing one in communication efficiency.

The rest of the paper is organized as follows. In Section 2, we formulate the problem. Section 3 addresses the framework of our algorithm. Section 4 describes further optimizations. Section 5 reviews related works. Section 6 reports the experimental results. Section 7 concludes the paper.

2 Problem Definition

Given m local sites $S = \{s_1, \ldots, s_m\}$, each holding a local database $D_i = \{t_{i,1}, \ldots, t_{i,n_i}\}$, which is a horizontal partition of a global uncertain database D in a d dimensional data space U, and a centralized server H which can communicate with any local site via Internet. We consider the tuple-level uncertainty [2] in this paper because it widely used in confidence-aware applications, i.e., each object t in D is associated with a existential probability $p(t)$ and the probabilities of objects are mutually independent. Without loss of generality, we assume smaller values are preferred in all dimensions. And we denote t' dominate t by "$\succ$".

In the uncertain data context, an object takes a probability to be in the skyline. Following the definition in [3], the skyline probability of object t in D (denote by $p_{sky}(t)$) equals the probability that t exists and all objects that dominate t do not exist:

$$p_{sky}(t) = p(t) \times \prod_{t' \in D \wedge t' \succ t} (1 - p(t')) \tag{1}$$

Given a user-defined probability threshold q, we use $q\text{-}SKY(D)$ to denote the answer set of the probabilistic skyline query on the global dataset D. $q\text{-}SKY(D)$ is a set of objects in D each of which takes a skyline probability at least q:

$$q\text{-}SKY(D) = \{\ t \in D \mid p_{sky}(t) > q\} \tag{2}$$

In the aforementioned distributed environment, the query delay mainly depends on the amount of data transferred. Therefore, our goal is retrieving the global probabilistic skyline *q-SKY*(D) from the set of distributed database $\{D_i\}$ with low communication cost.

The recent work [1] is sensitive to the data distribution, and incurs a sharply increase communication cost when the cardinalities of local skylines is relatively large. A feasible approach is pruning unqualified objects early in the local sites, which calls for knowledge of global data distribution sharing among local sites. Nevertheless, this is not trivial as two challenges naturally emerge: (1) A data summary that captures the probabilistic data distribution of the datasets compactly and data-independently, and enable early pruning of unqualified skyline candidates in local sites. (2) A mechanism sharing the data summary across the local sites which introduces minimal additional communication cost.

3 The Grid-Based Probabilistic Skyline Algorithm

3.1 The Framework

Our framework gracefully integrates three pre-processing steps into [1]. We first present the general framework as follows, and then detail each step in the sequel.

The general framework

1. *Loading data* (in local sites): each local site S_i maps local dataset D_i to the local data summary $G(D_i)$ and then transfers $G(D_i)$ to H.
2. *Merge and share* (in centralized server): H merges $G(D_i)$ into a global data summary $G(D)$. Then H broadcast $G(D)$ back to every local sites.
3. *Local pruning* (in local sites): each S_i uses $G(D)$ to prune unqualified local skyline objects.
4. *Skyline computation* (in local sites and centralized server): compute probabilistic skyline using [1] progressively.

3.2 Loading Data

The Grid-based Data Summary. A grid G splits each dimension of the data space U into n consecutive slices. That is, each of the n^d cells in G is a d dimensional hypercube of the same width. A cell $c \in G$ is identified by a d dimensional vector $\vec{c} = <c[0], \ldots, c[d-1]>$. $c[i]$ represents its order in G in the *ith* dimension from the origin point of U. A cell c is mapped to a one dimensional order $c.id$ to reduce the information representing c. The one dimensional order is computed by Hilbert curve [4], as it can be reconstructed back to coordinates of cells and facilitate computation in local pruning in section 3.4. For sake of brevity, we use c_i to denote the cell owns order i, i.e., $c_i.id = i$. Fig.1 shows an example of cell ordering for a 2 dimensional data space (n = 4). For example, $\vec{c_7}$ =<1, 2>, its order is 7. An uncertain data set is encoding by a grid-based data summary through mapping each object to the corresponding cell.

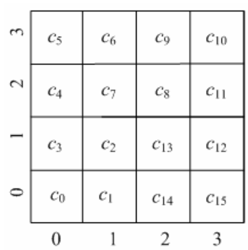


Fig. 1. Cell ordering

Mapping a Dataset to a Grid. When mapping D_i to $G(D_i)$, each object t in D_i is mapped to a cell $c(t)$. We use $c.CS(D)$ to denote the set of objects in D mapped to cell c. Each cell c is associated with a cell probability $c.cp$, which represents the probability that none of the objects in $c.CS(D)$ exist, i.e.,

$$c.cp = \prod_{t \in c.CS(D)} (1 - p(t)) \tag{3}$$

After mapping D_i to $G(D_i)$, the local sites S_i send the information of cells of $G(D_i)$ to H. Note that empty cells contribute nothing to data distribution of local database. For cell c ($c.cp < 1$), the information need to be transferred is: $< c.id, c.cp>$.

3.3 Merge and Sharing

In this phase, H merge the received cells to a global data summary G. H first initialize an empty grid G, i.e., for each c in G $c.cp$ is set to 1. And then, once H receives a cell c^L from a local site, it sets the probability of corresponding cell in H (the cell c^H which fulfills $c^H.id = c^L.id$) by the following equation:

$$c^H.cp = c^H.cp \times c^L.cp \tag{4}$$

Lemma 1. *Let G(D) represents the grid-based data summary of D, G is the data summary merged as above, then G is the data summary sketching D, i.e., G = G(D).*

Proof: We need prove that for any c and c' in G, if $c.id = c'.id$, then $c.cp = c'.cp$. By (3) we have $c'.cp = \prod_{t \in c'.CS(D)} (1 - p(t))$, note that $D = \bigcup_i D_i$, thus $c'.cp$ can be rewrite to $\prod_i \prod_{t \in c'.CS(D_i)} (1 - p(t))$. Then we get $c.cp = \prod_i \prod_{t \in c'.CS(D_i)} (1 - p(t))$ $= c'.p$ from (4). Thus we obtain $G = G(D)$.

3.4 Local Pruning

After equipped with the global data summary G, the local sites attempt to prune unqualified objects utilizing G. We first illustrate the pruning heuristic by the example in fig.2. Suppose a local site S_i holds a grid G received from H. For object t ($c(t) = c_7$) in

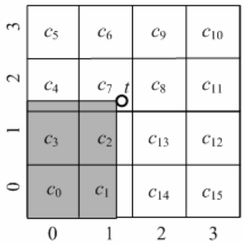


Fig. 2. Upper-bound object t with G

D_i, all the objects that dominate t is in the gray region, which covers cell c_0 and c_3 entirely. From equation (3) we can derive the upper-bound of $p_{sky}(t)$ as $p(t)\times c_0.cp\times c_3.cp$. Then t can be safely pruned if the upper-bound is less than q.

In order to prove the above observation theoretically, we first extend the dominate relationship between objects to cells naturally:

Definition 1. Dominate relationship between cells ("$\succ_c$"): $\forall c, c' \in G$, $c \succ_c c'$ if and only if $\vec{c} \succ \vec{c}'$. i.e., $\forall i \in [1,d]$, $\vec{c}[i] \le \vec{c}'[i]$, and $\exists i \in [1,d], \vec{c}[i] \ne \vec{c}'[i]$.

Definition 2. Strictly dominate relationship between cells ("$\succ_c^s$"): $\forall c, c' \in G$, $c \succ_c^s c'$ if and only if $c \succ_c c'$ and $\forall i \in [1,d], \vec{c}[i] \ne \vec{c}'[i]$.

Lemma 2. *For an object t in D_i, and a grid G for D, $p_{sky}(t)$ is upper-bounded by the product of p(t) and* $\prod_{c\in G \wedge c \succ_c^s c(t)} c.cp$.

Proof: For any t' mapped to $c(t')$, t' dominates t if $c(t')$ strictly dominates $c(t)$. Thus $\prod_{t''\in G \wedge t'' \succ t} (1-p(t''))$ is upper-bounded by $\prod_{c(t')\succ_c^s c(t)} c(t').cp$. From (1) we have $p_{sky}(t) \le p(t) \times \prod_{c \succ_c^s c(t)} c.cp$.

4 Further Optimization

Besides the empty cells, there are still large portion of cells that are redundant for determining the upper bound of skyline probability. We first present the observation by an example in fig.3. Suppose q equals 0.3, and the cell probability of c_0 and c_1 is 0.6 and 0.4 respectively, i.e., $c_0.cp \times c_1.cp$ = $0.6\times 0.4 < q$. Then all the cells strictly dominated by both c_0 and c_1 ($c_{8\sim13}$) are not necessary to transferred to H. This is because, any object that falls into $c_{8\sim13}$ will inevitably strictly dominated by c_0 and c_1, thus its

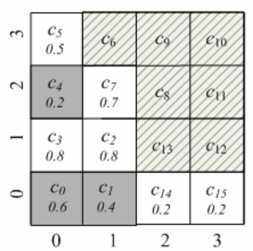


Fig. 3. Redundant cells

skyline probability is always less than q following lemma 2. Then cells in $c_{8\sim13}$ will never contribute to the probabilistic skyline computation and can be safely excluded from the cells that need to be sent. We can exclude c_6 in the same way.

We then formally proof the above heuristic. As depicted by fig.4, we use *DownLeft*(c) to denote the set of cells that dominate cell c as well as c itself, i.e., *DownLeft*(c)={ $c' \in G(D_i)$ | $c' \succ_c c$ } $\bigcup \{c\}$, and use *DownLeft*(c).*cp* to denote the product of cell probability of cells in *DownLeft*(c), i.e., *DownLeft*(c).*cp* $= \prod_{c' \in Downleft(c)} c'.cp$. The cells strictly dominated by c are denoted by *UpRight*(c), i.e., *UpRight*(c) = { $c'' \in G(D_i)$ | $c \succ_c^s c''$ } holds. Then we obtain lemma 3:

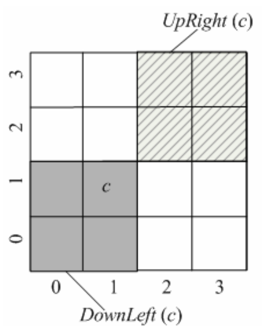


Fig. 4. Illustration of lemma 3

Lemma 3. *For any cell c in* $G(D_i)$, $UpRight(c)$ *are not necessarily transferred if* $Downleft(c).cp$ *is less than q.*

Proof: From the definition of $Downleft(c)$ and $UpRight(c)$ we can conclude that for any cell c in $UpRight(c)$, and any c' in $DownLeft(c)$, it always holds that $c' \succ_c^s c$. If t is dominated by c in $UpRight(c)$, then it must be dominated by any cell in $DownLeft(c)$. In another word, any object that can be prune by $UpRight(c)$ will always be pruned by $Downleft(c)$, thus cells in $UpRight(c)$ are not necessarily transferred.

5 Related Works

The skyline operator is first introduced into the database community by [5]. There has been considerable works on the distributed skyline query processing. Balke [6] first investigates the skyline computation under the distributed environments, where data is vertically scatted in multiple distributed nodes. Several works deal with distributed skyline retrieval in Peer-to-Peer network where different overlays are considered [7-9]. The recent literature considers distributed skyline retrieval under more general network architectures [10, 11]. However, the above algorithms can not be extended to uncertain data because of the semantic gap.

The skyline query was first extended to uncertain data by Pei [3]. ZHANG [12] studied the probabilistic skyline query in streaming uncertain data. They proposed an in-memory R-tree based approach to maintain the candidate set efficiently. Yiu [13] extended R-tree to facilitate the probabilistic skyline computation under the spatial database context. However, none of the above work considers the distributed probabilistic skyline query under the uncertain context.

More recently, DING [1] has proposed the first probabilistic skyline algorithm on distributed uncertain data. The communication efficiency of their algorithm stems from a priority-based scheme, where the central server computes global skyline in the order of local skyline probabilities. However, the communication cost of their algorithm highly depends on the data distribution, and increases sharply when the cardinality of local probabilistic skyline is relatively large.

6 Experimental Evaluations

6.1 Experimental Setup

Both our algorithm (Grid-based Probabilistic Skyline, GBPS) and e-DSUD are implemented in C++ and compiled by VC 2005 on a 3.8GHz Dual Core AMD processor with 2G RAM running Windows OS. The performance measures are number of bytes transferred during the query processing. Both real and synthetic datasets are used in our experiments. The real dataset (available from www.zillow.com) contains information about real estate all over the United States. We deal with 2 dimensions namely number of bedrooms and price gap. The price gap of a house is computed as the maximum house price in the dataset minus the price of the house. Intuitively, the real dataset with the dimensions (number of bedrooms, price gap) is more likely anti-correlated as a large room tends to has small price gap. We also generate the synthetic anti-correlated dataset following [5]. The dimensionality d of synthetic dataset ranges from 2 to 4. Both of the real and synthetic datasets contain 1 million objects.

We associate uncertainty to the aforementioned datasets by randomly assigning each object with an occurrence probability following normal distribution with the mean value μ equals 0.8 and the standard deviation σ equals 0.3. The threshold q is set to 0.3. In each experiment, the dataset of cardinality n is horizontally split to m partitions equally, each simulate a local database. The default setting of d and m is 2 and 100 respectively unless otherwise specified.

6.2 Experimental Results

Optimal value of *n*. We first study the optimal setting of the number of slides on both real and synthetic datasets. As depicted by fig.5, the optimal value of *n* is around 14 and 16 respectively. When *n* is relatively large, transferring the cells consume too much additional communication cost. However, when *n* is relatively small, the pruning power of the grids will be limited. In the sequel, we set *n* to 14 for both datasets.

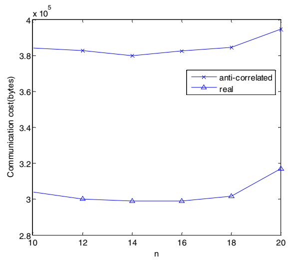
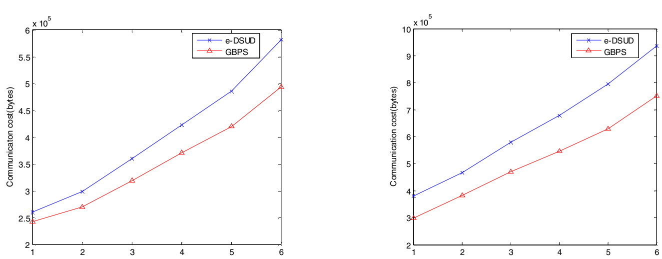


Fig. 5. Optimal value of *n*

Communication cost vs. number of local sites. Both algorithms incur more communication cost on anti-correlated dataset than on real dataset, since the former generates much more skyline candidates. GBPS saves average 12.1% and 19.7% bandwidth against e-DSUD for real dataset(depicted in fig.6 (a)) and anti-correlated dataset (depicted in fig.6 (b)) respectively, which confirms that GBPS perform better than e-DSUD for more skewed data. On both datasets, GBPS scales better with respect to *m*.

Communication cost vs. dimensionality. We test the scalability of GBPS against e-DSUD with respect to dimensionality by varying *d* from 2 to 4 on anti-correlated dataset. As shown by fig.7, GBPS always incurs lower communication cost than e-DSUD. Furthermore, the percentage of bandwidth saved by GBPS increases when *d* gets larger, which indicates that GBPS has better scalability than e-DSUD with respect to dimensionality.



(a) Real dataset (b) Anti-correlated dataset

Fig. 6. Communication cost vs. Number of local sites

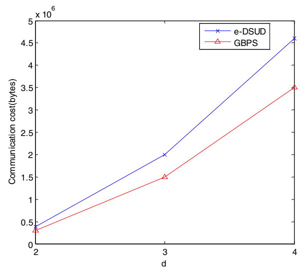


Fig. 7. Communication cost vs. dimensionality

7 Conclusions

In this paper, we propose a grid-based communication-efficient algorithm of probabilistic skyline on distributed uncertain data. By sharing a grid-based data summary, our algorithm improves the communication efficiency against existing approach.

As can be expected, grid-based approach performs not well on correlated and independent datasets. This suggests that, after samples or histograms are used to estimate the distribution of the underlying data, our approach can be an alternative when the underlying dataset is more likely anti-correlated, which is deemed as the challenging problems for skyline computation.

Acknowledgement

This work is supported by the National High-Tech Research and Development Plan of China under Grant No. 2006AA01Z474, No. 2007AA01Z451, and No. 2007AA010502.

References

1. Ding, X., Jin, H.: Efficient and Progressive Algorithms for Distributed Skyline Queries over Uncertain Data. In: 30th International Conference on Distributed Computing Systems. IEEE Computer Society, Genova (2010)
2. Aggarwal, C.C., Yu, P.S.: A Survey of Uncertain Data Algorithms and Applications. IEEE Transactions on Knowledge and Data Engineering 21(5), 609–623 (2009)
3. Pei, J., Jiang, B., Lin, X., Yuan, Y.: Probabilistic Skylines on Uncertain Data. In: 33rd International Conference on Very Large Data Bases. VLDB Endowment, Vienna (2007)
4. Jagadish, H.V.: Linear Clustering of Objects with Multiple Attributes. In: ACM SIGMOD International Conference on Management of Data. ACM, New Jersey (1990)
5. Borzsony, S., Kossmann, D., Stocker, K.: The Skyline Operator. In: 17th International Conference on Data Engineering. IEEE Computer Society, Hannover (2001)
6. Balke, W.-T., Güntzer, U., Zheng, J.X.: Efficient distributed skylining for web information systems. In: Hwang, J., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 256–273. Springer, Heidelberg (2004)

7. Wang, S., Ooi, B.C., Tung, A.K.H., Xu, L.: Efficient Skyline Query Processing on Peer-to-Peer Networks. In: 23rd International Conference on Data Engineering. IEEE Computer Society, Istanbul (2007)
8. Wu, P., Zhang, C., Feng, Y., Zhao, B.Y., Agrawal, D.P., El Abbadi, A.: Parallelizing Skyline Queries for Scalable Distribution. In: Ioannidis, Y., Scholl, M.H., Schmidt, J.W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., Böhm, C. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 112–130. Springer, Heidelberg (2006)
9. Vlachou, A., Doulkeridis, C., Kotidis, Y., Vazirgiannis, M.: Skypeer: Efficient Subspace Skyline Computation over Distributed Data. In: 23rd International Conference on Data Engineering. IEEE Computer Society, Istanbul (2007)
10. Cui, B., Lu, H., Chen, Q.X.L., Dai, Y., Zhou, Y.: Parallel Distributed Processing of Constrained Skyline Queries by Filtering. In: 24rd International Conference on Data Engineering. IEEE Computer Society, Cancun (2008)
11. Zhu, L., Tao, Y., Zhou, S.: Distributed Skyline Retrieval with Low Bandwidth Consumption. IEEE Transactions on Knowledge and Data Engineering 21(3), 384–400 (2009)
12. Zhang, W., Lin, X., Zhang, Y., Wang, W., Yu, J.X.: Probabilistic Skyline Operator over Sliding Windows. In: 25rd International Conference on Data Engineering. IEEE Computer Society, Shanghai (2009)
13. Yiu, M.L., Mamoulis, N., Dai, X., Tao, Y., Vaitis, M.: Efficient Evaluation of Probabilistic Advanced Spatial Queries on Existentially Uncertain Data. IEEE Transactions on Knowledge and Data Engineering 21(1), 108–122 (2009)