Adaptive Mining the Approximate Skyline over Data Stream

Liang Su, Peng Zou, and Yan Jia

School of Computer Science National University of Defense Technology
Changsha 410073, China

suliangnudt@gmail.com, zpeng@nudt.edu.cn, jiayanjy@vip.sina.com

Abstract. Skyline queries, which return the objects that are better than or equal in all dimensions and better in at least one dimension, are useful in many decision making and monitor applications. With the number of dimensions increasing and continuous large volume data arriving, mining the approximate skylines over data stream under control of losing quality is a more meaningful problem. In this paper, firstly, we propose a novel concept, called *approximate skyline*. Then, an algorithm is developed which prunes the skyline objects within the acceptable difference and adopts correlation coefficient to adjust adaptively approximate query quality. Furthermore, our experiments show that the proposed methods are both efficient and effective.

Keywords: Approximate skyline, adaptive algorithm, data stream, data mining.

1 Introduction

For two objects $X=(x_1, x_2, \dots, x_d)$ and $Y=(y_1, y_2, \dots, y_d)$ in the d -dimensional space, X dominates Y if $x_i \leq y_i$ for $1 \leq i \leq d$. Skyline computing aims to find all objects that are not been dominated by other objects and roots in many decision making and monitor applications. Borzsönyi et al. [1] proposed first two algorithms for the skyline computation: the BNL (block nested loop) and DC (divide and conquer) algorithm. Subsequently, skyline problem and its variants [2,3,4,5] have been extensively studied, and a lot of algorithms have been developed. BBS [2] compute the skyline using nearest neighbor search. However, in [7], Bentley et al. proved that the average number of skyline objects is $O((\ln N)^{d-1})$ (N is the size of the original objects) which is the cardinality bound most often cited and employed. In paper [3], Xuemin Lin et al. consider the problem of efficiently computing the skyline against the most recent N elements in a data stream, they had realized the algorithm scalability and proposed approximate skyline problem. None of the algorithms referenced above was originally designed to support approximate skyline computation over data stream. These are the motivations of our research in the paper. To the best of our knowledge, there is no similar work existing in the literature in the context of adaptive approximate skyline computation over data stream.

The rest of the paper is organized as follows. Sections 2 provide our techniques for processing approximate skyline problem and algorithm. Results of comprehensive experimental studies are discussed in section 3. Section 4 concludes the paper.

2 Approximate Skyline Problem and Algorithms

To eliminate the number of skyline set and improve the manageability, intuitively, first we depict accurately how close the degree of two skyline objects is. We are naturally aware of the geometry distance. Then we use an acceptable difference distance to prune some neighbor objects, and find the approximate skyline set. Behind of the paper we only use the Euclidian distance as the *dist* function.

Definition 1. (Approximate Skyline Set) Let Ω be the set of d -dimensional objects, and $\Omega_{skyline}$ be the skyline set on Ω . The acceptable difference distance is ε and obviously satisfies $\varepsilon \geq 0$. $\Omega_{ASkyline}$ represents the approximate skyline set that satisfies:

1. $\forall X_i, X_j \in \Omega_{ASkyline}, X_i \neq X_j$, then $dist(X_i, X_j) \geq 2\varepsilon$
2. $\forall X_i \in \Omega_{Skyline}$, then $\exists X_j \in \Omega_{ASkyline}$, s.t. $dist(X_i, X_j) \leq \varepsilon$

Definition 2. (Strongly Dominating Relationship) An object $X_i \in \Omega$ strongly dominates another object $X_j \in \Omega$, noted as $X_i \triangleright X_j$, if $X_i + \varepsilon \succ X_j$ ($X_i \succ X_j$ means X_i dominates X_j). That is to say: $\forall 1 \leq k \leq d, x_i^k + \varepsilon \leq x_j^k$, and $\exists 1 \leq t \leq d, x_i^t + \varepsilon < x_j^t$, reversely, X_j is a strongly dominated object by X_i .

Definition 3. (Correlation Coefficient) Given n d -dimensional objects, which $X_i = (x_i^1, \dots, x_i^d)$, $1 \leq i \leq n$, set matrix $A = (X_1, \dots, X_n)^T$ and $Y_j = (x_1^j, \dots, x_n^j)^T$,

$$\text{then } A = \begin{pmatrix} x_1^1 & \dots & x_1^d \\ \vdots & \ddots & \vdots \\ x_n^1 & \dots & x_n^d \end{pmatrix} = (Y_1, \dots, Y_d), \text{ and } \rho(Y_i, Y_j) = \frac{E((Y_i - EY_i)(Y_j - EY_j))}{\sqrt{DY_i} \cdot \sqrt{DY_j}} = \frac{E(Y_i Y_j) - EY_i EY_j}{\sqrt{DY_i} \cdot \sqrt{DY_j}}$$

is the correlation coefficient; $EY_i = \frac{1}{n} \sum_{k=1}^n x_k^i$ is the expectation of vector Y_i ;

$DY_i = E(Y_i - EY_i)^2 = EY_i^2 - (EY_i)^2$ is the variance of vector Y_i . The correlation coefficient is between -1 and +1. We construct a translation function which

is $\Phi(\rho) = \frac{a+b}{2} - \rho \frac{b-a}{2}$, $1 \leq a \leq b$ and choose $\varepsilon \Phi(\rho)$ as the bound of difference distance to adaptively mining the approximate skyline in data stream.

Theorem 1. Given a dataset $\Omega_{Skyline}$, if $X_i, X_j \in \Omega_{Skyline}$ and $dist(X_i, X_j) \leq \varepsilon$,

$$\text{then } \sum_{k=1}^d |x_i^k - x_j^k| \leq \varepsilon \sqrt{d}.$$

2.1 Extended Micro-cluster Feature Tree Based Approximate Skyline Algorithm

To save the memory and run time, we partition the sliding window data into extended micro-cluster feature tree based on CF-tree [6] which only save the approximate

skyline set and the information of their close neighbors(two or more micro-clusters can build a larger one). The pseudo code of *EMCFTA* algorithm is showed below.

Definition 4. (Extended Micro-Cluster Feature Tree, EMCF-Tree) A extended micro-cluster feature tree for a set of d -dimensional objects $X_1, X_2, \dots, X_n$ ($X_i = (x_i^1, \dots, x_i^d)$), is defined as a $(3d+3+L(H_{skyline})+L(H_{dominated}))$ -tuple $(\overline{ASF1}, \overline{ASF2}, \overline{ASF3}, K(X_{centriod}), X_{centriod}, Parent, H_{skyline}, H_{dominated})$. The definition of each of these entries is as follows, Figure 2 is the whole data structure:

- The p -th entry of $\overline{ASF1}$ is equal to $\sum_{j=1}^n x_j^p$; The p -th entry of $\overline{ASF2}$ is equal to $\sum_{j=1}^n (x_j^p)^2$;
- The p -th entry of $\overline{ASF3}$ is equal to $\min_{j=1}^n (x_j^p)$; $K(X_{centriod})$ is the ordinal of $X_{centriod}$;
- $X_{centriod}$ is belong to $\Omega_{ASkyline}$, which represents the centroid of its micro-cluster;
- $Parent$ is pointer to its parent node, which is to trace back to modify its ancestors;
- $H_{skyline}$ is a heap to save the skyline objects within its micro-cluster; $H_{dominated}$ is a heap to save the dominated objects by $X_{centriod}$; n is the length of sliding window.

Fig. 1. Tree Micro-Cluster regions

Fig. 2. Extended Micro-Cluster Feather Tree

Algorithm EMCFTA(ϵ, Ω)

/* ϵ is the approximate skyline distance; input the original data objects in Ω ; Output is the approximate skyline set $\Omega_{ASkyline}$. The X_{new} represents a new object from the data stream.*/

- 1 find the closest micro-cluster from the EMCF-Tree, that is MC_d which also is the leaf node;
 - 2 if $X_{new} \succ X_{centriod}$, then X_{new} is located in region(V) Fig 1, so $X_{centriod} := X_{new}$;
 - 3 else if $X_{centriod} \succ X_{new}$
 - 4 begin
 - 5 if $K(X_{new}) - K(X_{centriod}) \leq n - K(X_{centriod}) \bmod n$, then no action;
 - 6 else X_{new} is located in region(I) Fig 1, so add X_{new} to $H_{dominated}$;
 - 7 end;
 - 8 else if $\text{dist}(X_{new}, X_{centriod}) \leq \epsilon$, then X_{new} is located in region(IV) Fig 1, and add X_{new} to $H_{skyline}$;
 - 10 else X_{new} is located in region(III) Fig 1, create a new micro-cluster leaf node to MC_d 's parent, and X_{new} is the centroid object.
-

3 Experiments

We use the three most popular synthetic benchmark data, *correlated*, *independent*, and *anti-correlated* [1,2,3,4,5] and implement them in C++. Experiments were performed on an Intel Pentium 4, CPU 2 GHZ, and RAM 512 MB, with Windows XP. All generated data value is between 0 and 1, the difference distance $\varepsilon \in [0,1]$ and dimension d is from 2 to 10.

Firstly, we compare $\Omega_{ASkyline}$ with $\Omega_{Skyline}$ in size at three data distributions. We observe that the result set produced by *EMCFTA* algorithm is between 30% and 55% of the basic skyline set with $d=5$ and $\varepsilon=0.1$. Secondly, we would examine the quality of approximate skyline set. Because the number of basic skyline objects in the correlation distribution is fewer than other two distributions sharply, we only use the anti-correlation and independent distributions as next experiments. To measure the quality of approximate skyline set, We use the average distance in $\Omega_{Skyline}$ and $\Omega_{ASkyline}$ which is the average distance among the arbitrary two objects. These two formulas are as follows:

$$AvgDist_{Skyline} = \frac{\sum_{i=1}^{|\Omega_{Skyline}|} \left(\sum_{j=2 \wedge j \neq i}^{|\Omega_{Skyline}|} dist(X_i, X_j) \right)}{(|\Omega_{Skyline}| \cdot (|\Omega_{Skyline}| - 1) / 2)} \quad (1)$$

$$AvgDist_{ASkyline} = \frac{\sum_{i=1}^{|\Omega_{ASkyline}|} \left(\sum_{j=2 \wedge j \neq i}^{|\Omega_{ASkyline}|} dist(X_i, X_j) \right)}{(|\Omega_{ASkyline}| \cdot (|\Omega_{ASkyline}| - 1) / 2)} \quad (2)$$

The average distance ratio (equals: $100 * AvgDist_{Skyline} / AvgDist_{ASkyline}$) is between 90 and 99, the skyline objects have been reduced to almost 45% of basic skyline objects. So our algorithm is efficient which are meaningful for data stream mining.

4 Conclusions and Future Work

In this paper, we propose a novel concept and an efficient algorithm which can prune adaptively the skyline objects. In future, we plan to develop a parallel and distributed algorithm to meet more scalable and changeable data stream environment.

References

1. Stephan Börzsönyi, D.K., Konrad Stocker. The Skyline Operator. in ICDE (2001).
2. Dimitris Papadias, Y.T., Greg Fu, Bernhard Seeger. An Optimal and Progressive Algorithm for Skyline Queries. in SIGMOD (2003).
3. Xuemin Lin, Y.Y., Wei Wang, Hongjun Lu. Stabbing the Sky: Efficient Skyline Computation over Sliding Windows. in ICDE (2005).
4. Chee-Yong Chan, H.V.J., et al. On High Dimensional Skylines. in EDBT (2006).
5. Y. Tao, et al. SUBSKY: Efficient Computation of Skylines in Subspaces. in ICDE (2006).
6. Tian Zhang, R.R., Miron Livny. BIRCH: An Efficient Data Clustering Method for Very Large Database. in SIGMOD (1996).
7. Bentley, J.L., Kung, H. T., Schkolnick, M., Thompson, C. D. On The Average Number of Maxima in a Set of Vectors and Applications. JACM (1978).