doi:10.3772/j.issn.1006-6748.2019.03.005

Spatial skyline query method based on Hilbert R-tree in multi-dimensional space①

Li Song(李　松)②, Zhang Liping, Li Shuang, Hao Xiaohong
(College of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, P. R. China)

Abstract

In view of the shortage of the spatial skyline query methods (SSQ methods) in dealing with the problem of skyline query in multidimensional space, a spatial skyline query method based on Hilbert R-tree in multidimensional space is proposed. This method takes the advantages of Hilbert R-tree which combines R-tree and Hilbert curve with high efficiency and dimensionality reduction. According to the number of query points, the proposed method in static query point environment is divided into single query point of SSQ method(SQ-HSKY algorithm) and multi-query points of SSQ method (MQP-HSKY algorithm). The SQ-HSKY method uses the spatial relationship between objects to propose pruning strategy and the skyline set in the filtering and refining process are computed. The MQP-HSKY method uses the topological relationship between data points and query points to prune non skyline points and generate the dominant decision circle to obtain the global skyline set. Theoretical study and experiments confirm the effectiveness and superiority of these methods on the skyline query.

Key words: skyline query, Hilbert curve, R-tree, Hilbert R-tree, topological relationship

0 Introduction

The skyline query is a kind of typical multi-objective optimization problem[1]. It has been successfully applied in market analysis, decision support, location service, environment monitor, data exploration, data mining, machine learning, database visualization and other fields. The spatial skyline query is a kind of important skyline query. Traditional skyline query is to find the data set which is not controlled by other collections of data points. The spatial skyline query is to search the data set which is not spatially dominated by other data points in the d dimension.

Some important achievements have been made in the field of the skyline query. Ref.[2] proposed a computational sharing strategy based on efficient identification and computational independence. At the same time, two algorithms which are Bottom-Up and Top-Down algorithm are proposed to calculate the sky cube based on these sharing strategies. But the renewal cost is big. Ref.[3] first proposed the concept of reverse skyline query and proposed the BBRS algorithm which is an improvement of BBS algorithm. For spatial skyline queries under multiple query points environment in low dimensional space, Ref.[4] studied the farthest spatial skyline query. A basic algorithm TFSS and a progressive algorithm BBFS are proposed which help to identify the spatial locations far from the bad locations. For skyline queries in multidimensional spaces, In Ref.[5], Salsa algorithm was proposed to deal with the skyline problem through classification. Ref.[6] proposed a skyline algorithm based on sorting which could deal with pairwise comparisons between incomplete data. This algorithm used two optimization techniques: block generation order and point generation order. The method can retrieve meaningful skyline points by adjusting the two specific parameters. The algorithm proposed in Refs[5,6] is very efficient for small amount of data, but when the amount of data is large, the algorithm is costly and inefficient.

Ref.[7] studied the skyline evaluation of multidimensional data with partially ordered domains and two new methods were proposed. Inspired by the grid theory and the existing skyline algorithm, the first method used an appropriate plan to achieve complete order

① Supported by the National Natural Science Foundation of China (No. 61872105), the Science and Technology Research Project of Heilongjiang Provincial Education Department (No. 1253lz004) and the Scientific Research Foundation for Returned Scholars Abroad of Heilongjiang Province of China(No. LC2018030).

② To whom correspondence should be addressed. E-mail: lisongbeifen@163.com

Received on Oct. 28, 2018

from half order. Inspired by the queue, the second method used an appropriate access and index method. Ref. [8] studied spatial skyline queries and proposed three algorithms: B^2S^2, VS^2 and VCS^2.

In recent years, the skyline queries have been extended further to the k-dominant skyline query[9], the reverse skyline query[10], the probabilistic skyline query[11-13], the skyline-Join query[14], the Top-k reverse skyline query[15], the dynamic skyline query[16], the skyline queries on probabilistic data[17], the MapReduce skyline query[18] and so on. However, there are some gaps in the skyline query of multi-dimensional space. Therefore, it is of great significance to study the skyline query method in multidimensional space.

The existing results have shortcomings in dealing with the dimension, which leads to the problem of low efficiency of data processing when dealing with multidimensional spatial skyline queries. And the problem of dimension processing in multidimensional space is more critical. It is of great significance for dealing with data and solving the skyline set in multidimensional space. Therefore, the method of reducing dimension based on the Hilbert R-tree[19] is adopted in this paper. In order to further improve the performance of the skyline query in multi-dimensional space, this paper proposes a spatial skyline query method based on Hilbert R-tree, which has high performance in dealing with the spatial skyline query problem in multidimensional space.

1 Basic definitions

Given a set of d-dimensional data points $P = \{p_1, p_2, \cdots, p_m\}$ and a set of d-dimensional query points $Q = \{q_1, q_2, \cdots, q_n\}$, and $m >> n$.

Definition 1[20] (the distance from q to spatial objects): In the Euclidean space $E(n)$, the distance from point q to a space object p is represented by $\|p, q\|$:

$$\|p, q\| = (\mathrm{Min}\sum_{i=1}^{n} |x_i - q_i|^2)^{1/2}, \quad \forall x = (x_1, x_2, \cdots, x_n) \in p \tag{1}$$

Definition 2[20] (Mindist distance): In d-dimensional Euclidean space $E(n)$, the minimum distance of point q to rectangular E in the same space is expressed as $Mindist(E, q)$:

$$Mindist(E,q) = (\sum_{i=1}^{n} |E_i - q_i|^2)^{1/2} \tag{2}$$

in which

$$E_i = \begin{cases} s_i & \text{if } q_i < s_i \\ t_i & \text{if } q_i < t_i \\ q_i & \text{the others} \end{cases} \tag{3}$$

Definition 3[17] (MinMaxdist distance): In d-dimensional Euclidean space $E(n)$, the maximum-minimum distance from point q to minimum bounding rectangle $E = (S, T)$ in the same space is represented by $MinMaxdist(E, q)$:

$$MinMaxdist(E,q) = (\min(|E_{mk} - q_k|^2 + \sum_{1 \leqslant i \leqslant n}^{i \neq k} |E_{mi} - q_i|^2))^{1/2} \quad 1 \leqslant k \leqslant n \tag{4}$$

in which

$$E_{mk} = \begin{cases} S_k & \text{if } q_k \leqslant \dfrac{(S_k + t_k)}{2} \\ t_k & \text{the others} \end{cases} \tag{5}$$

$$E_{mi} = \begin{cases} S_i & \text{if } q_i \leqslant \dfrac{(S_i + t_i)}{2} \\ t_i & \text{the others} \end{cases} \tag{6}$$

Definition 4[1] (Dominance): Given data point $p_1 \in P$, $p_2 \in P$, a query point q, p_1 dominates p_2, if and only if the conditions are satisfied by

(1) $\forall i \in \{1, \cdots, n\}$, $|q^i - p_1^i| \leqslant |q^i - p_2^i|$;

(2) $\exists j \in \{1, \cdots, n\}$, $|q^j - p_1^j| < |q^j - p_2^j|$。

Definition 5[1] (Skyline set): A set of data points that are not dominated by any other data points are called skyline set. The global skyline set takes the entire dataset as the query scope and is denoted as GS.

Definition 6[8] (Spatial skyline query): Given data point set $P = \{p_1, p_2, \cdots, p_m\}$ and query point set $Q = \{q_1, q_2, \cdots, q_n\}$ in d-dimensional space, a spatial skyline query returns a collection of points that are not dominated by other data points in P over a series of derived attributes.

Definition 7[21] (Hilbert curve): One-to-one mapping between d dimensional data space R^d and one-dimensional data space I, which can be denoted as H: $R^d \rightarrow I$. If point $p \in R^d$, then $H(p) \in I$. $H(p)$ is also called as the H value of point p.

Definition 8 (multidimensional spatial skyline query): In the d-dimensional Euclidean space, given a data point set $P = \{p_1, p_2, \cdots, p_m\}$ and a query point set $Q = \{q_1, q_2, \cdots, q_n\}$. In multidimensional space, the dimension is not less than 3 ($d \geqslant 3$). The multidimensional spatial skyline query is to find a space skyline set about Q in dataset P. The specific definition form is as follows: $SSQ(P) = \{p_i \in I \mid \|p_i, Q\| < \|p_j, Q\|, R^d \rightarrow I, d \geqslant 3, i \neq j, 1 \leqslant i, j \leqslant m\}$.

2 Spatial skyline query method based on Hilbert R-tree

According to the number of the query point, the skyline query in multidimensional space is divided into

the skyline query method under the single query point environment and the spatial skyline query method under the multiple query points.

2.1 Spatial skyline query under the single query point environment

The skyline query method under the single query point environment (SQ-HSKY algorithm) proposed in this section can be divided into two stages. In the first stage, data points are grouped through the Hilbert-R tree, the dominated data points are filtered according to the proposed pruning strategy and the more accurate candidate set is obtained (CLS_HSKY algorithm). In the second stage, the dominated data points are filtered out by using the definition of dominance and the global skyline set is obtained (MLS_HSKY algorithm).

2.1.1 Filtration process

Firstly, the Hilbert curve is used to reduce the dimension of the data set to delete the unnecessary attributes. The structure of the Hilbert-R tree is used for partitioning, and it is easier to handle the data. Then, the dominance relationship between the data points is obtained by Theorem 1 and Theorem 2. In order to determine the dominance relation and prune the non skyline points, Theorem 1 is given. At first, Lemmas 1 and 2 are given in this section.

Lemma 1[20] Given query point q and the minimum bounding rectangle E, the spatial object set $O = \{o_i, 1 \leqslant i \leqslant m\}$ of bounding rectangle E, then for $\forall\ o \in O$, $Mindist(E,q) \leqslant \| o,q \|$.

Lemma 2[20] Given query point q and the minimum bounding rectangle E, the spatial object set $O = \{o_i, 1 \leqslant i \leqslant m\}$ of bounding rectangle E, then for $\exists\ o \in O$, $\| o,q \| \leqslant MinMaxdist(E,q)$.

Theorem 1 Given query point q and the minimum bounding rectangle E and E', if the minimum distance $Mindist\ (E,q)$ between E and q is larger than the Maximum-minimum distance $MinMaxdist\ (E', q)$ between E' and q, then E is pruned.

Proof According to Lemma 1 and Lemma 2, it can be seen that, $Mindist\ (E, q) \leqslant \| o, q \|$, $\| o,q \| \leqslant MinMaxdist(E, q)$, then $Mindist(E,q) \leqslant MinMaxdist\ (E,q)$, and if $Mindist(E,q) > MinMaxdist(E', q)$, so there must be $Mindist(E,q) > Mindist\ (E',q)$ and $Mindist(E,q) > \| E',q \|$. So E is pruned.

Theorem 2 Given query point q and minimum bounding rectangle E. If there is data point o, $\| o,q \| > MinMaxdist(E,q)$, then data point o is pruned.

Proof According to Lemma 2, it can be seen that, there is data point o' in the surrounded data points of E, which satisfies $\| o',\ q \| \leqslant MinMaxdist(E,q)$. And for data point o, $\| o,\ q \| > MinMaxdist(E,q)$, then there must be $\| o,q \| > \| o',q \|$. At the same time, according to the domination rule of spatial skyline query, o is dominated by o', then o is pruned.

Theorem 3 Given query point q and minimum bounding rectangle E, if there is data point o, $Mindist\ (E,q) > \| o,\ q \|$, then bounding rectangle E is pruned.

Proof According to Lemma 2, it can be seen that, there is arbitrary data point o' in the outsourced data points in E which satisfies $Mindist\ (E,\ q) \leqslant \| o',q \|$. And for data point o, $Mindist\ (E,q) > \| o,q \|$, then there must be $\| o',q \| > \| o,q \|$. At the same time, according to the domination rule of spatial skyline query, o is dominated by o', and o' is an arbitrary data point that is outsourced in E, so o' dominates E, and E is pruned.

As shown in Fig. 1, p_1, p_2, p_3, $\cdots$, p_{35} are the data points, q is the query point. According to theorem 1, it is known that $Mindist(E_8,q) > MinMaxdist(E_7, q)$, then E_8 is pruned, and p_{32}, p_{33}, p_{34}, p_{35} in E_8 are pruned. There is $Mindist\ (E_3,q) > MinMaxdist\ (E_4, q)$, then E_3 is pruned, and p_9, p_{10}, p_{11}, p_{12} in E_3 are pruned . There is $Mindist(E_5,q) > MinMaxdist(E_6, q)$, then E_5 is pruned, and p_{19}, p_{20}, p_{21}, p_{22}, p_{23} in E_5 are pruned. According to Theorem 2, it is known that $\| p_{26},q \| > MinMaxdist(E_4,q)$, $\| p_{27},q \| > MinMaxdist(E_4,q)$, $\| p_{28},\ q \| > MinMaxdist\ (E_4, q)$, $\| p_1,q \| > MinMaxdist(E_4,q)$, $\| p_{17},q \| > MinMaxdist(E_4,q)$, $\| p_{24},q \| > MinMaxdist\ (E_4,q)$, $\| p_{25},q \| > MinMaxdist(E_6,q)$, $\| p_2,q \| > MinMaxdist\ (E_8,q)$, so p_1, p_2, p_{17}, p_{24}, p_{25}, p_{26}, p_{27}, p_{28} are pruned. According to theorem 3, There are $Mindist\ (E_4,q) > \| p_5,q \|$ and $Mindist(E_6,q) > \| p_5,q \|$, so E_4 and E_5 are pruned. Finally, candidate set $\{p_3, p_4, p_5, p_6, p_{29}, p_{30}\}$ is obtained.

The main idea of the algorithm proposed in this section is: According to Theorem 1, Theorem 2 and theorem 3, data points are processed and a large number of dominated data points are pruned and the candidate set after pruning is obtained. $P = \{p_1, p_2, \cdots, p_n\}$ is a data point set, Q is a query point, the dimension of data points through the Hilbert curve is reduced. The Hilbert R-tree according to the H value and the distribution of data points are obtained. And the data points along the Hilbert R-tree are processed. At first, according to theorem 1, if there are two bounding rectangles E and E', meanwhile there is the relation that the minimum distance between E and q is greater than the maximum-minimum distance between E' and q, then the bounding rectangle E is pruned. According to Theorem 2, if there are bounding rectangle E and

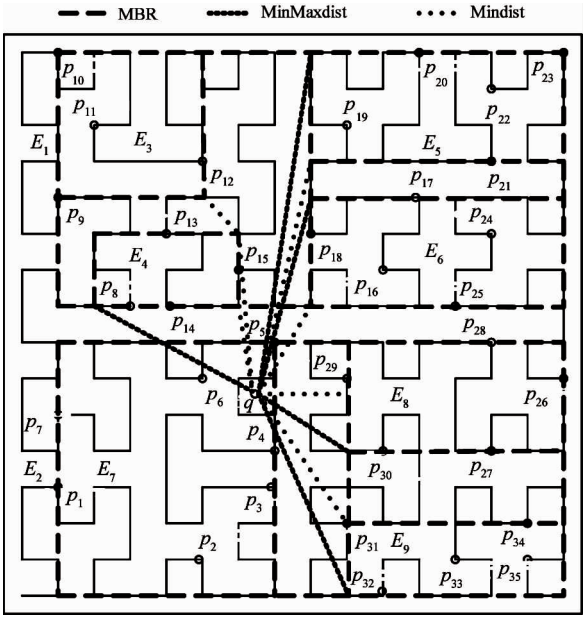

Fig. 1 The relations of dominance among the partition between the data points

data point p, and they meet the condition that the distance between p and q is greater than the maximum-minimum distance between E and q, then data point p is pruned. According to theorem 3, if there are bounding rectangle E and data point p, and they meet the condition that the minimum distance between E and q is greater than the distance between p and q. Then bounding rectangle E is pruned. The remaining data points which are not pruned constitute a more accurate candidate set.

Based on the above discussion, an algorithm is given to obtain the candidate set as shown in Algorithm 1.

Algorithm 1 CLS _ HSKY(P, Q)

input: data point set P, query point Q

output: candidate set LS

begin

1. Creat-Hilbert(P);// To reduce the dimension of data set P by Hilbert curve
2. H_p←Calculate _ $H(P)$;// H values of all data points are calculated on the Hilbert curve
3. LS←P;// Putting data point set P into the LS set
4. Create Hilbert R-tree;// Hilbert R-tree is established by H value and data distribution
5. while not _ empty(Hilbert R-tree) do
6. for E_i in Hilbert R-tree do
7. if $Mindist(E_i, q) > MinMaxdist(E_j, q)$ then //theorem 1
8. E_i is pruned;
9. LS←$LS - E_i$;
10. end if
11. for p_i in Hilbert R-tree do
12. if $\| p_i, q \| > MinMaxdist(E_i, q)$ then //theorem 2
13. p_i is pruned;
14. LS←$LS - p_i$;
15. end if
16. if $Mindist(E_i, q) > \| p_i, q \|$ then //theorem 3
17. E_i is pruned;
18. LS←$LS - E_i$;
19. end if
20. end for
21. end for
22. end while
23. return LS;

end

2.1.2 Refining process

Some of the dominated skyline sets have been filtered out through the CLS _ HSKY algorithm. But the remaining data points may also have data points which are dominated. Therefore, this section mainly deals with the dominance relationship between the remaining data points, and finally the final skyline set is obtained by refining each data point.

The main idea of MLS _ HSKY algorithm is to judge dominance relationship between the data points in the candidate skyline set (LS set) one by one. By comparing the Euclidean distances from each data point to the query point, a set of points are obtained which are not dominated by other data points. Finally the spatial skyline set of single query point condition in the static query point environment is obtained.

Based on the above discussion, the refining algorithm is further presented in this section as shown in Algorithm 2.

Algorithm 2 MLS _ HSKY(P, q)

input: query point q, data point set P in candidate skyline set LS

output: The final skyline set GS

begin

1. GS←∅
2. GS←LS ;
3. for p_i in GS do
4. if $\| p_i, q \| > \| p_j, q \|$ then
5. GS←$GS - p_i$;
6. end if
7. end for
8. return GS

end

2.2 Spatial skyline query under the multiple query points environment

SQ-HSKY algorithm is a spatial skyline query method under the environment of single query point. But in real life, the number of the query points is often multiple. So this section studies the spatial skyline query method under the multiple query points condition in the multidimensional space, and the MQP-HSKY algorithm based on Hilbert R-tree is given.

When the query point is unique, the dominance condition is determined only by judging the relationship between the distance from each data point to the query point. But when the number of the query points increases, the situation has changed. After data points are added, not only the distance relationship between data points and a query point is considered, but also the distance relationship between data points and all query points is considered, so the situation is more complex. In order to make use of the topological relationship between points, the dominance relationship between data points is determined, and Theorem 4 and Theorem 5 are given.

Theorem 4[8] If $\exists p \in P$ and p is in CH(Q), then p must be a skyline point.

In order to give Theorem 5, this section first gives the definition of the dominating decision circle, such as Definition 9.

Definition 9 (Dominant decision circle): Given data points set $P = \{p_1, p_2, \cdots, p_m\}$ and query points set $Q = \{q_1, q_2, \cdots, q_n\}$, the dominant decision circle is defined as a circle where q_i is the circle center and $dist(q_i, p)$ is the radius. $dist(q_i, p)$ is the distance between q_i and p.

Theorem 5 $\exists p_1, p_2 \in P$, the circle $Circle(q_i, p_2)$ with q_i as the center and $dist(q_i, p)$ as the radius. If p_1 is in the outside of the dominant decision circle, then p_1 is dominated by p_2.

Proof Taking q_i as the center and $dist(q_i, p)$ as the radius to generate the circle, then all the positions in the obtained $Circle(q_i, p)$ are closer to q than p. By the definition of skyline, it is known that the data points closer to the query point set dominate the data points far from the query point set. Therefore, taking q_i as the center and $dist(q_i, p)$ as the radius to generate circles, the interior points of $Circle(q_i, p)$ are closer to Q than the points outside $Circle(q_i, p)$. Because p_1 is outside of $Circle(q_i, p_2)$, $dist(q_i, p_2) < dist(q_i, p_1)$, so p_2 dominates p_1.

As shown in Fig. 2, q_1, q_2, q_3 and q_4 are query points, p_3 and p_{13} are data points, and the distance between the two points is the length of the dotted line. Generate $Circle(q_1, p_3)$ with q_1 as the center of a circle, p_3 is on the circle, p_{13} is outside the circle, so $dist(q_1, p_3) < dist(q_1, p_{13})$. $Circle(q_2, p_3)$ is generated with q_2 as the center of a circle, p_3 is on the circle, p_{13} is outside the circle, so $dist(q_2, p_3) < dist(q_2, p_{13})$. $Circle(q_3, p_3)$ is generated with q_3 as the center of a circle, p_3 is on the circle, p_{13} is outside the circle, so $dist(q_3, p_3) < dist(q_3, p_{13})$. $Circle(q_4, p_3)$ is generated with q_4 as the center of a circle, p_3 is on the circle, p_{13} is outside the circle, so $dist(q_4, p_3) < dist(q_4, p_{13})$. In conclusion, p_3 is closer to Q than p_{13}, p_3 dominates p_{13}.

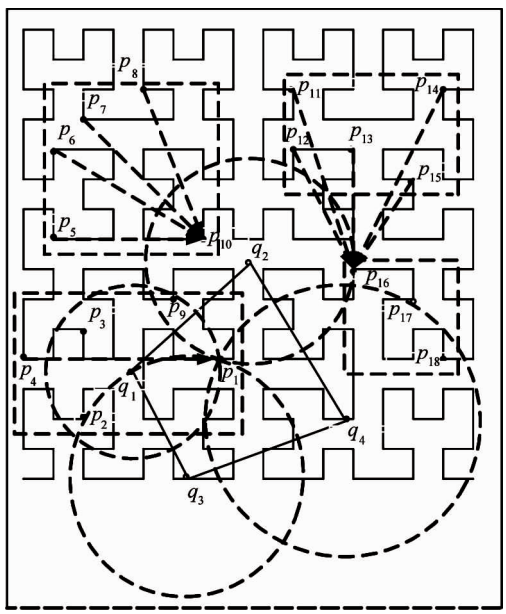

Fig. 2 Multi-query point of dynamic skyline query

As shown in Fig. 2, the node of the tree in which p_1 is located intersects with CH(Q). So p_1 is added to the skyline set and generate $Circle(q_i, p_1)$. There is no intersection between the four circles. Point p_4 is outside of $Circle(q_1, p_1)$, p_5, p_6, p_7 and p_8 are outside of $Circle(q_2, p_1)$, p_{11}, p_{12}, p_{13}, p_{14} and p_{15} are outside of $Circle(q_2, p_1)$. So the above nodes are dominated by p_1, and all of them are added to the non skyline set (*NS* set). Then the rest of the data points are judged again. Finally, the global skyline set is $\{p_1, p_2, p_3, p_9, p_{10}, p_{16}, p_{17}, p_{18}\}$.

The main idea of the MQP-HSKY algorithm proposed in this section is to judge the data points according to the theorems. Query point is q_i. Data point is p_i. The partition that the circle with q_i is center and $dist(q_i, p)$ is radius subtracts the intersection part of several circles is expressed by *ND*. The data points within the *ND* range are skyline points. The intersec-

tion of each dominant decision circle of data point p is the dominated region of data point p. Data points in this region dominate data point p. According to the topological relationships between the Hilbert R-tree nodes and the convex hull of Q, it can be divided into two cases. When the node intersects with the convex hull, the data points of the intersection can be added directly to the skyline set. And the range of the ND region is updated. When the node does not intersect with the convex hull, then the topological relation between the data point in the node and the ND is judged. If the data point is in the ND region, then it can be judged as a skyline point. And finally, a complete global skyline set is obtained.

Based on the above discussion, the MQP-HSKY algorithm is further presented as shown in Algorithm 3.

Algorithm 3 MQP-HSKY(P, Q)

Input: data point $P = \{p_1, p_2, \cdots, p_m\}$, query point $Q = \{q_1, q_2, \cdots, q_n\}$.

Output: the global skyline set GS of P about Q.

begin

1. Calculate _ H(P); // To calculate the H value of each data point
2. Create _ Hilbert-R-tree(H_P); //To build a Hilbert-R tree according to Hilbert coding sequence and H value
3. Calculate _ CH(Q); // To calculate the convex hull CH(Q) of Q for the query point
4. $GS = \{\}$; $NS = \{\}$; $ND = \varnothing$;
5. for MR_i in Hilbert-R-tree do //MR_i is the MBR of the tree
6. if $GS = \varnothing$ then
7. $GS \leftarrow GS + p_1$;// To add the first data point p_1 to the GS set
8. $ND = \sum_{j=1}^{n} Circle(q_j, p_i) - \cap \sum_{j=1}^{n} Circle(q_j, p_i)$;
9. end if
10. if $MR_i \cap \mathrm{CH}(Q) \neq \varnothing$ then
11. $GS \leftarrow GS + p_i$;//Add data point p_i of intersection between MR_i and CH(Q)
12. $ND = \sum_{j=1}^{n} Circle(q_j, p_i) - \cap \sum_{j=1}^{n} Circle(q_j, p_i)$;
13. end if
14. for p_i in $MR_i - (MR_i \cap \mathrm{CH}(Q))$ do
15. if $p_i \notin ND$ then
16. $NS \leftarrow NS + p_i$;
17. $P \leftarrow P - p_i$;
18. end if
19. if $p_i \in ND$ then
20. $GS \leftarrow GS + p_i$;
21. $ND = \sum_{j=1}^{n} Circle(q_j, p_i) - \cap \sum_{j=1}^{n} Circle(q_j, p_i)$;
22. end if
23. if $MR_i \cap CH(Q) = \varnothing$ then //MR_i is the MBR of the tree
24. for p_i in MR_i do
25. if $p_i \notin ND$ then
26. $NS \leftarrow NS + p_i$;
27. $P \leftarrow P - p_i$;
28. end if
29. if $p_i \in ND$ then
30. $GS \leftarrow GS + p_i$;
31. $ND = \sum_{j=1}^{n} Circle(q_j, p_i) - \cap \sum_{j=1}^{n} Circle(q_j, p_i)$;
32. end if
33. end for
34. end if
35. end for
36. end for
37. return GS;

end

3 Experimental results and analysis

In this section, the proposed method is evaluated by experiments, and the performance efficiency of the method is verified. The proposed SQ-HSKY algorithm mainly handles the spatial skyline query problem under the single query point environment in the multi-dimensional space. The MQP-HSKY algorithm is mainly suitable for the skyline query under the multi query points environment in multi-dimensional space. In this section, the SQ-HSKY algorithm is compared with the PCS algorithm of Ref. [22], and the MQP-HSKY algorithm is compared with the Top-k MSSQ algorithm of Ref. [23]. PCS algorithm is to deal with the skyline query under the single query point condition, and the skyline set is calculated by invoking classical algorithms. At first the PCS algorithm calls the BBS algorithm to calculate local skyline, and then the D&C algorithm is called. The Top-k MSSQ algorithm is to deal with the skyline query under multiple query points condition, and the Top-k MSSQ algorithm mainly uses a scoring function to prune skyline.

The configuration of the experiment platform: Pentium 4-core 2.216 GHz CPU, 8 GB memory, 500 GB hard disk, the programs are achieved by Microsoft Visual 2005 on the Windows10 operating system. The data

set used in this paper is the real data set provided by the State Library of California, which is a general survey of the population distribution information of California (http://www.library.ca.gov/lds/demographicprofiles/index.html).

The experiment compares the performance of the SQ-HSKY and PCS algorithm, MQP-HSKY and Top-k MSSQ algorithm by comparing the effects of dataset size on algorithms at different distributions. At the same time, the experiment also compares the influence of dimension on algorithm under different data distribution. The experiments show that the response time of the SQ-HSKY algorithm and MQP-HSKY algorithm proposed in this paper are better than that of the PCS algorithm and Top-k MSSQ algorithm in the different data sets and multi dimension.

First, the SQ-HSKY algorithm and the PCS algorithm are compared experimentally. Fig. 3 verifies the influence of the size of the data set on the response time in the case of the negative correlation and independent distribution. The dimension is 8. In Fig. 3, it can be seen that in the case of different data distribution, the response time of SQ-HSKY algorithm and the PCS algorithm are both increasing along with the data set increasing. Compared with the PCS algorithm, the SQ-HSKY algorithm has less time-consuming. Moreover, as the amount of data increasing, the time-consuming difference between the SQ-HSKY algorithm and the PCS algorithm is getting bigger and bigger. The SQ-HSKY algorithm proposed in this paper adopts the efficient method of decreasing dimension. Meanwhile the algorithm uses index structure which is suitable for handling large amounts of data. And the efficiency is obvious.

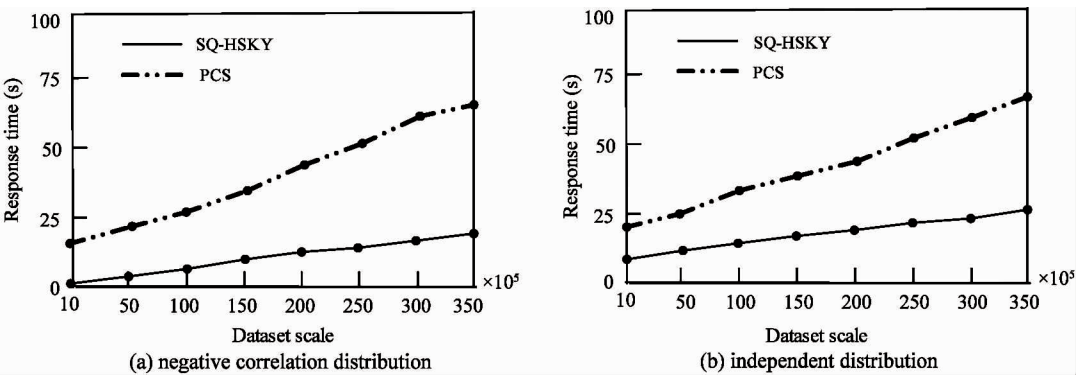

Fig. 3 Effects of data set size on response time at different distribution

Fig. 4 verifies the effect of dimensionality on response time in the case of negative correlation and independent distribution. The size of the fixed data set is 300×10^5.

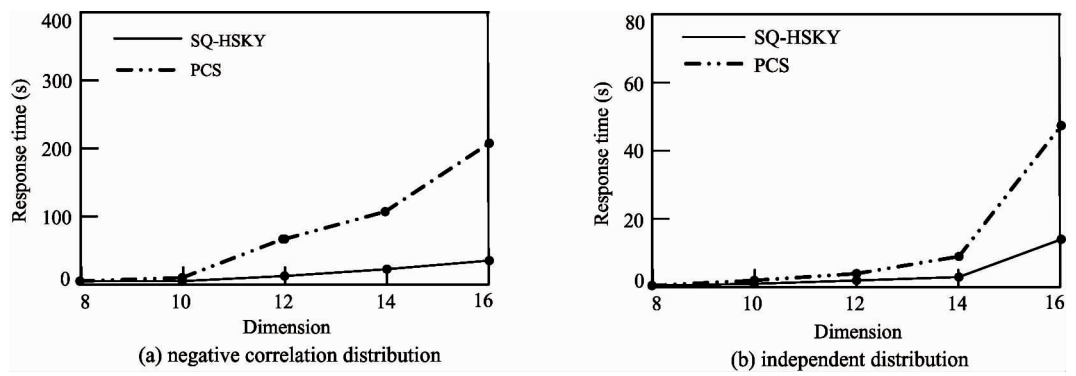

Fig. 4 Effects of dimensionality on response time at different distribution

It can be seen from Fig. 4 that the curves both have an increasing trend in the case of negatively correlated and independently distributed. In negative correlation, when the dimension is below 10, the curves of the two algorithms are infinitely close to the X axis. Starting from the 10 dimension, the curve of the two algorithms is gradually increasing, and that of the PCS algorithm grows faster. From the overall view of the two images, the proposed SQ-HSKY algorithm requires less response time than the PCS algorithm. Starting from 12

dimension, the gap of algorithmic efficiency between SQ-HSKY algorithm and PCS algorithm is getting bigger and bigger. The response time consumed by PCS algorithm is more than that consumed by PCS algorithm in negative correlation and independent distribution.

The following experiment is to compare the MQP-HSKY algorithm with the Top-k MSSQ algorithm. Fig. 5 verifies the influence of the size of the data set on the response time of the MQP-HSKY algorithm and the Top-k MSSQ algorithm in the case of negative correlation and independent distribution. The fixed dimension is 10.

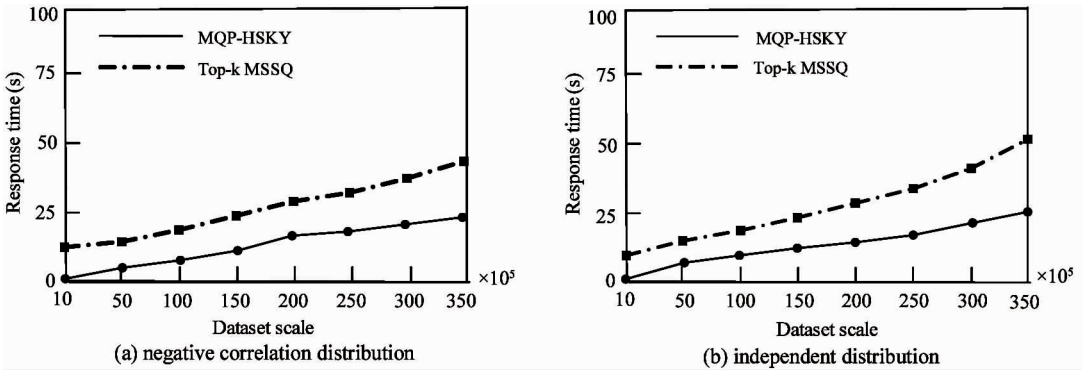

Fig. 5 Effects of data set size on response time at different distribution

In Fig. 5, it can be seen that in the case of the negative correlation and independent distribution, the response time of the two algorithms increases along with the increase of the data set size. The response time of the MQP-HSKY algorithm is less than that of the Top-k MSSQ algorithm regardless of the negative correlation or the independent distribution.

Fig. 6 verifies the influence of dimensionality of MQP-HSKY algorithm and Top-k MSSQ algorithm on response time under negative correlation and independent distribution respectively. In Fig. 6, it can be seen that in the case of negative correlation and independent distribution, the curve of the two algorithms is always monotonically increasing, and the curve is infinitely close to the X axis when the dimension is between 8 and 10. The curve begins to grews slowly when the dimension is above 10. But the curve of the MQP-HSKY algorithm grows slowly. The main reason is that the Top-k MSSQ algorithm uses the idea of sorting. And it deals with the data in order. Therefore, the response time consumed by the algorithm increases when the dimension increases. The MQP-HSKY algorithm uses Hilbert curve, which has a great advantage in dimensionality reduction, so it is very efficient for multi-dimensional situation.

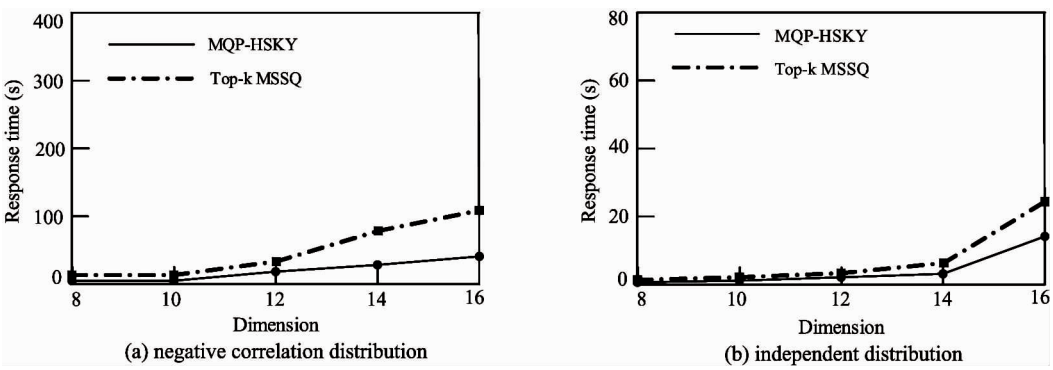

Fig. 6 Effects of dimensionality on response time at different distribution

4 Conclusions

A solution to spatial skyline queries in multidimensional spaces based on Hilbert R-tree is proposed. In the static environment, the SQ-HSKY algorithm under single query point environment and the MQP-HSKY algorithm under multiple query points environment are proposed. The two algorithms use the Hilbert curve to reduce the dimension, and Hilbert R-tree is used as the data allocation method. The SQ-HSKY algorithm has two processes including filtering and refining to calculate skyline. The MQP-HSKY algorithm prunes non

skyline points by using the topological relationship between data points and query points, and generates the dominant decision circle. Then it prunes dominated data points to obtain the global skyline set. The future research focuses on the following:

1) Uncertain data skyline query with the fusion of vague direction relation.

2) Uncertain data skyline query with the real-time query platform for big data based on Hadoop[24].

References

[1] Borzsonyi S, Kossmann D, Stocker K. The skyline operator[C]. In: Proceeding of the 17th International Conference on Data Engineering, Heidelberg, Germany, 2001. 421-430

[2] Yuan Y, Lin X, Liu Q. Efficient computation of the skyline cube[C]. In: Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, 2005. 241-252

[3] Dellis E, Seeger B. Efficient computation of reverse skyline queries[C]. In: Proceedings of the 33rd VLDB Conference, Vienna, Austria, 2007. 291-302

[4] You W G, Lee M W, Im H, et al. The farthest spatial skyline queries[J]. *Information Systems*, 2013, 38(3): 286-301

[5] Bartolini I, Ciaccia P, Patella M. Efficient sort-based skyline evaluation[J]. *ACM Transactions on Database Systems*, 2008, 33(4): 133-135

[6] Jongwuk L, Hyeonseung I, Gae-won Y. Optimizing skyline queries over incomplete data[J]. *Information Sciences*, 2016, 361-362: 14-28

[7] Zhang S, Mamoulis N, Cheung D W. Efficient skyline evaluation over partially ordered domains[C]. In: Proceedings of the VLDB Endowment, Singapore, 2010, 3(1-2): 1255-1266

[8] Sharifzadeh M, Shahabi C. The spatial skyline queries[C]. In: Proceeding of the International Conference on Very Large Data Bases, Seoul, Korea, 2006. 751-762

[9] Miao X, Gao Y, Chen G. k-dominant skyline queries on incomplete data[J]. *Information Sciences*, 2016, 367-368: 990-1011

[10] Gao Y, Liu Q, Zheng B. On processing reverse k-skyband and ranked reverse skyline queries[J]. *Information Sciences*, 2015, 293: 11-34

[11] Park Y, Min J K, Shim K. Processing of probabilistic skyline queries using MapReduce[C]. In: Proceedings of the VLDB Endowment, Kohala Coast, Hawaii, USA, 2015, 8(12): 1406-1417

[12] Zhao Y Wang Y J, Wang Y, et al. An efficient method of parallel skyline query processing over uncertain data streams[J]. *Journal of Computer Research and Development*, 2013, 50(z2): 132-139

[13] Sun S L, Dai D B, Huang Z H, et al. Algorithm on Computing skyline over probabilistic data stream[J]. *Acta Electronica Sinica*, 2009, 37(2): 285-293

[14] Vlachou A, Doulkeridis C, Polyzotis N. Skyline query processing over joins[C]. In: Proceeding of the ACM Sigmod International Conference on Management of Data, New York, USA, 2011. 73-84

[15] Zhang B, Jiang T, Gao Y J, et al. Top-k query proessing of reverse skyline in metric space[J]. *Journal of Computer Research and Development*, 2014, 51(3): 627-636

[16] Zhang L, Zou P, Jia Y, et al. Continuous dynamic skyline queries over data stream[J]. *Journal of Computer Research and Development*, 2011, 48(1): 77-85

[17] Trieu M N, Cao J L, He Z. Answering skyline queries on probabilistic data using the dominance of probabilistic skyline tuples[J]. *Information Sciences*, 2016, 340: 58-85

[18] Koh J L, Chen C C, Chan C Y. MapReduce skyline query processing with partitioning and distributed dominance tests[J]. *Information Sciences*, 2017, 375: 114-137

[19] Kamel I. Hiblert R-tree: An improved R-tree using fractals[C]. In: Proceeding of the International Conference on Very Large Data Bases, San Francisco, USA, 1994. 500-509

[20] Hao Z X. Query and Reasoning in Spatio-Temporal Database[M]. Beijing: Science Press, 2010. 27-35(In Chinese)

[21] Zhang R X, Zheng S J, Xia Q G. Self-adaptive image segmentation method based on Hilbert scan and wavelet transform[J]. *Journal of Image and Graphics*, 2008, 13(4): 666-671

[22] Cosgaya-Lozano A, Rau-Chaplin A, Zeh N. Parallel computation of skyline queries[C]. In: Proceedings of the IEEE 21st International Symposium on High Performance Computing Systems and Applications, Saskatoon, Canada, 2017. 12-20

[23] Wanbin S, Fabian S, Christian K, et al. Top-k manhattan spatial skyline queries[J]. *Lecture Notes in Computer Science*, 2014, 8344: 22-33

[24] Liu X L, Xu P D, Liu M L, et al. Design and development of real-time query platform for big data based on hadoop[J]. *High Technology Letters*, 2015, 21(2): 231-238

Li Song, born in 1977. He is a professor at Harbin University of Science and Technology. He received the Ph. D. degree from College of Computer Science and Technology, Harbin University of Science and Technology, Harbin, China, in 2009. His research interests include database theory and application, data mining, data query, big data, etc.