A Survey of Skyline Queries and its Variants

Yaqing Wang & Ming Wen
HKUST
Outline

1. Introduction

2. Road Map

3. Skyline Queries
   1. Maximum Vector Problem
   2. Skyline Operator
   3. Divide & Conquer
   4. Block Nested Loop
   5. Sort First Search
   6. Nearest Neighbor
   7. Branch and Bound
   8. Comparison

4. Variants
   1. Dynamic Skylines
   2. Spatial Skylines
   3. Reverse Skylines
   4. Probabilistic Skylines
   5. Skylines Related to K
   6. Others

5. Applications
Introduction

Skyline Queries
Given a set of points \( \{p_1, p_2, \ldots, p_N\} \), the skyline queries returns a set of points (referred to as the skyline points), such that any point \( P_i \) is not dominated by any other point in the dataset.

Dominance Relation:
A point dominates another point if it is as good or better in all dimensions and better in at least one dimension.
Maximum Vector Problem

Definition

Find vectors that is **not dominated** by any of the vectors from the set.

A vector dominates another if

- **Each** of its components has an **equal or higher** value than the other vector’s corresponding component
- And it has a **higher value on at least one** of the corresponding components

Actually, this is the Skyline!
Skyline Operator

**Property**

One of the nice properties of the Skyline of a set $M$ is that for any monotone scoring function $M \rightarrow \mathbb{R}$, if $p \in M$ maximizes that scoring function, then $p$ is in the Skyline.

The Skyline does not contain any point which are nobody’s favorite.

---

2001
Skyline Operator

Example

Find a cheap hotel that is close to beach.
- Minimize price (x-axis)
- Minimize distance to beach (y-axis)
- Points not dominated by other points

Skyline contains everyone favorite hotel regardless of preferences

The Skyline does not contain any hotels which are nobody’s favorite.
SELECT *
FROM Hotels h
WHERE h.city = 'Hawaii' AND NOT EXISTS(
    SELECT *
    FROM Hotels h1
    WHERE h1.city = 'Hawaii'
    AND h1.distance ≤ h.distance
    AND h1.price ≤ h.price
    AND (h1.distance < h.distance OR h1.price < h.price));

SELECT *
FROM Hotels
WHERE city = 'Hawaii'
SKYLINE OF price MIN, distance MIN;

This operator can filter out interesting points from a large set of data points.
Skyline Operator

Extend SQL’s `SELECT` statement by an optional `SKYLINE OF` clause as follows:

```
SELECT ... FROM ... WHERE ...
GROUP BY ... HAVING ...
SKYLINE OF [DISTINCT] d1 [MIN | MAX | DIFF], ..., dm [MIN | MAX | DIFF] ORDER BY ...
```

`d1 , ..., dm` denote the dimensions of the Skyline; e.g., `price`, `distance to the beach`, or `rating`. `MIN, MAX, and DIFF` specify whether the value in that dimension should be minimized, maximized, or simply be different.

Skyline operator can also be combined with other database operations (e.g., join and Top N).
Divide and Conquer

Main Idea

1. Divides the dataset into several partitions so that each partition fits in memory.
2. Compute the partial skyline of every partition.
3. The final skyline is obtained by merging the partial ones.
Divide and Conquer

Example-2 partition

1. Get median value
2. Divide tuples into 2 partition
3. Compute skyline of each partition
4. Merge partition

Authors’ Improvement: M-Way & Early Skyline

---

2001
Divide and Conquer

Example-4 partition

If the input does not fit into main memory, we can use median of other dimension to partition more! The beauty is that we need not merge S12 and S21 because the tuples of these two sets are guaranteed to be incomparable.

---

2001
Block Nested Loop

Main Idea

If no object dominates it, then it is a skyline.

1. Compare each tuple with one another
2. Keep a window in main memory to contain best tuple
3. Write to temp file (if window has no space)

Authors’ Improvement – self organizing list

---

2001
### Example

Assume we check the tuples against the window **alphabetically**.

We set the window can only contain 2 tuples.

<table>
<thead>
<tr>
<th>Window</th>
<th>Pruned</th>
<th>Temporary file</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>-</td>
</tr>
<tr>
<td>ac</td>
<td>b</td>
<td>-</td>
</tr>
<tr>
<td>ac</td>
<td>b</td>
<td>d</td>
</tr>
</tbody>
</table>

…

![Diagram showing data points](image)

Distance vs. Price graph with data points marked.

---

**2001**

Block Nested Loop

Example

Assume we check the tuples against the window *alphabetically*. We set the window can only contain 2 tuples.

<table>
<thead>
<tr>
<th>Window</th>
<th>Pruned</th>
<th>Temporary file</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>-</td>
</tr>
<tr>
<td>ac</td>
<td>b</td>
<td>-</td>
</tr>
<tr>
<td>ac</td>
<td>b</td>
<td>d</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

---

Sort First Skyline

Main Idea

BNL relies on main memory size and its inadequacy for progressive processing. The sort first skyline (SFS) is a variation of BNL.

1. First sorting the entire dataset according to a (monotone) preference function.
2. Use the ordered list as input for BNL algorithm

Presorting ensures that a point $p$ dominating another $p'$ must be visited before $p'$.

We can immediately output the points inserted to the list as skyline points!
Nearest Neighbor

Main Idea

NN uses the results of nearest neighbor search to partition the data universe recursively.
1. The partitions resulting after the discovery of a skyline point are inserted in a to-do list.
2. While the to-do list is not empty, NN removes one of the partitions from the list and recursively repeats the same process.

Assume that distances are computed according to the L1 norm, that is, the sum of the coordinates of p.
Branch and Bound Skyline

Main Idea

BBS is also based on nearest-neighbor search. It uses R-trees due to their simplicity and popularity.

1. Starts from the root of an R-tree and inserts all its entries into a heap sorted according to their mindist.
2. Remove out the top entry of the heap.
   (1) If e is dominated, discard it.
   (2) Otherwise
      i. If e is an intermediate node, put its children which are not dominated into the heap.
      ii. If e is a leaf, insert into the skyline list.
### Branch and Bound Skyline

<table>
<thead>
<tr>
<th>Action</th>
<th>Heap Contents</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access root</td>
<td>$&lt;e_7, 4&gt; &lt;e_6, 6&gt;$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>Expand $e_7$</td>
<td>$&lt;e_3, 5&gt; &lt;e_6, 6&gt; &lt;e_5, 8&gt; &lt;e_4, 10&gt;$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>Expand $e_3$</td>
<td>$&lt;i, 5&gt; &lt;e_6, 6&gt; &lt;h, 7&gt; &lt;e_5, 8&gt; &lt;e_4, 10&gt; &lt;g, 11&gt;$</td>
<td>${i}$</td>
</tr>
<tr>
<td>Expand $e_6$</td>
<td>$&lt;h, 7&gt; &lt;e_5, 8&gt; &lt;e_1, 9&gt; &lt;e_4, 10&gt; &lt;g, 11&gt;$</td>
<td>${i}$</td>
</tr>
<tr>
<td>Expand $e_1$</td>
<td>$&lt;a, 10&gt; &lt;e_4, 10&gt; &lt;g, 11&gt; &lt;b, 12&gt; &lt;c, 12&gt;$</td>
<td>${i, a}$</td>
</tr>
<tr>
<td>Expand $e_4$</td>
<td>$&lt;k, 10&gt; &lt;g, 11&gt; &lt;b, 12&gt; &lt;c, 12&gt; &lt;l, 14&gt;$</td>
<td>${i, a, k}$</td>
</tr>
</tbody>
</table>

---

2003

## Comparisons

<table>
<thead>
<tr>
<th></th>
<th>Progressiveness</th>
<th>Absence of false misses</th>
<th>Absence of false hits</th>
<th>Fairness</th>
<th>Incorporating preferences</th>
<th>Universality</th>
</tr>
</thead>
<tbody>
<tr>
<td>D&amp;C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SFS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BBS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Also lots of related data structures can be compared. e.g. bitmap, index, ZBtree, lattice structure.
Dynamic Skyline

**Description**

For all the data points in the original d-dimensional space, a dynamic skyline query specifies m dimension functions which map all the original data to a new m-dimensional dynamic space, and return the skylines in the new data space.

**Example**

Original Space \((p_x, p_y, p_z)\)

Dynamic Space \((f_1(p_x, p_y), f_2(p_z))\)

\[ f_1(p_x, p_y) = \sqrt{(p_x - u_x)^2 + (p_y - u_y)^2}, \text{ and } f_2(p_z) = p_z \]
Dynamic Skyline

**Definition**

Given a query point $q$ and a data set $P$, a Dynamic Skyline Query (DSQ) according to $q$ retrieves all data points in $P$ that are not dynamically dominated. A point $p_1 \in P$ dynamically dominates $p_2 \in P$ with regard to the query point $q$ if (1) for all $i \in \{1, \ldots, d\}$: $|q^i - p_1^i| \leq |q^i - p_2^i|$, and (2) at least one $j \in \{1, \ldots, d\}$: $|q^j - p_1^j| < |q^j - p_2^j|$.

**Difference**

In the above definition, it is equivalent to compute the traditional skyline, having transformed all points in the new data space where point $q$ is the origin and the absolute distances to $q$ are used as mapping functions.

---

2007 Dynamic Skyline
Dynamic Skyline

Usage Scenario

2007 Dynamic Skyline
Dynamic Skyline

Usage Scenario

2007 Dynamic Skyline
Spatial Skyline

Definition

Given a set of data points $P$ and a set of query points $Q$ in a d-dimensional space, an SSQ retrieves those points of $P$ which are not dominated by any other point in $P$ considering a set of query points in $Q$.
Spatial Skyline

Difference

p spatially dominates p’ with respect to Q iff we have $D(p, q_i) \leq D(p', q_i)$ for all $q_i \in Q$ and $D(p, q_j) < D(p', q_j)$ for some $q_j \in Q$. 

2006 Spatial Skyline Queries
Spatial Skyline

Solution

2006 Spatial Skyline Queries
Spatial Skyline

Solution

**Bruto Force** \( O(\mid P \mid^2 \mid Q \mid) \)

**With Pruning** \( O(\mid S \mid^2 \mid C \mid + \sqrt{\mid P \mid}) \)

|S| and |C| are the solution size and the number of vertices of the convex hull of Q, respectively

---

2006 Spatial Skyline Queries
Reverse Skyline

Motivation

(a) Points in a 2-d Space

(b) Database Table

Reverse Skyline

Motivation

(a) Dynamic Skyline

(b) Reverse Skyline

Reverse Skyline

Definition

Let \( P \) be a \( d \)-dimensional data set. A Reverse Skyline Query (RSQ) according to the query point \( q \) retrieves all points \( p_1 \in P \) where \( q \) is in the dynamic skyline of \( p_1 \). Formally, a point \( p_1 \in P \) is a reverse skyline point of \( q \in P \) iff \( \nexists p_2 \in P \) such that (a) for all \( i \in \{1, \ldots, d\} \): \( |p_2^i - q^i| \leq |p_1^i - q^i| \) and (b) for at least one \( j \in \{1, \ldots, d\} \): \( |p_2^j - q^j| < |p_1^j - q^j| \).
Reverse Skyline

Solution

Global Skyline

A point \( p_1 \in P \) globally dominates \( p_2 \in P \) with regard to the query point \( q \) if (1) for all \( i \in \{1, \ldots, d\} \): \( (p_1^i - q_i)(p_2^i - q_i) > 0 \), (2) for all \( i \in \{1, \ldots, d\} \): \( |p_1^i - q_i| \leq |p_2^i - q_i| \) and (3) for at least one \( j \in \{1, \ldots, d\} \): \( |p_1^j - q_j| < |p_2^j - q_j| \). The global skyline of a point \( q \), \( GSL(q) \), contains those points which are not globally dominated by another point according to \( q \).
Reverse Skyline

Solution

**Lemma 1.** Let $q$ be the query point and $\text{GSL}(q)$ be the set of global skyline points and $\text{RSL}(q)$ the set of reverse skyline points. Then, $\text{RSL}(q) \subseteq \text{GSL}(q)$.

**Lemma 2.** Given the global skyline $\text{GSL}(q)$ of point $q$. Assume point $s \in \text{GSL}(q)$ is a global skyline point. If $\exists p \in P$ such that for all $i \in \{1, \ldots, d\}$ $|p_i - q_i| < |s_i - q_i|$ then point $s$ is not a reverse skyline point of $q$. 

---

2007 Reverse Skyline

Probabilistic Skyline

Motivation

Uncertain Data

Sensing
Environmental Surveillance
Market Analysis

Pervasive

Data Randomness
Incompleteness
Delayed Data Updates

Reasons

2007 Probabilistic Skyline
Probabilistic Skyline

Motivation

Uncertain Data

An uncertain object has multiple instances, or alternatively, each object is associated with a probability density function.
Probabilistic Skyline

Motivation

Uncertain Data

Each object is associated with a probability density function (PDF) in the data space $D$, denoted as $f(u)$

$$\int_{u \in D} f(u) \, du = 1$$
Probabilistic Skyline

Dominance relation

Let $U$ and $V$ be two uncertain objects, and $f$ and $f'$ be the corresponding probability density functions, respectively. Then, the probability that $V$ dominates $U$ is

$$Pr[V < U] = \int_{u \in D} f(u) \left( \int_{v < u} f'(v) \, dv \right) \, du = \int_{u \in D} \int_{v < u} f(u) f'(v) \, dv \, du$$  \hspace{1cm} (1)

In the discrete case, let $U = \{u_1, \ldots, u_l_1\}$ and $V = \{v_1, \ldots, v_l_2\}$ be two uncertain objects and their instances. The probability that $V$ dominates $U$ is given by

$$Pr[V < U] = \sum_{i=1}^{l_1} \frac{1}{l_1} \cdot \frac{|\{v_j \in V | v_j < u_i\}|}{l_2} = \frac{1}{l_1 l_2} \sum_{i=1}^{l_1} |\{v_j \in V | v_j < u_i\}|$$  \hspace{1cm} (2)
Probabilistic Skyline

P-Skyline
Given a probability threshold \( p \) (0 <= \( p \) <= 1), the \( p \)-skyline is the set of uncertain objects each of which takes a probability of at least \( p \) to be in the skyline.

Solution

Bounding  Pruning  Refining

MBB: Minimum Bounding Box
Kd-Tree: Partition Tree

Variants Related to $K$

- K-Dominating Queries
- K-Dominant Skylines
- Top-K Representative Skyline
Variants Related to $K$

**Definition**

K-Dominating queries retrieves $K$ points that dominate the largest number of other points. Strictly speaking, this is not a skyline query, since the result does not necessarily contain skyline points.
Variants Related to $K$

**K-Dominating Queries**

**Motivation**
As the number of dimensions increases, the chance of one point dominating another point is very low. As such, the number of skyline points become too numerous to offer any interesting insights.

**Definition**
A point $p$ is said to $k$-dominate another point $q$ if there are $k$ ($\leq d$) dimensions in which $p$ is better than or equal to $q$ and is better in at least one of these $k$ dimensions.

**K-Dominant Skylines**

2006 $K$ Dominating Skylines
Variants Related to $K$

**K-Dominating Queries**

**Definition**

selecting $k$ skyline points so that the number of points, which are dominated by at least one of these $k$ sky-line points, is maximised.

**K-Dominant Skylines**

**Top-K Representative Skyline**

Other Variants

Parallel Skyline[14]
Subspace Skyline[9]
Constrained Skyline[17]
Skyband Skyline[17]
Application on the Skyline

- Road Network
- Mobile context
- Recommender system
- P2P
- Web information
- Social network


Thank You!