Keyword-matched Data Skyline in Peer-to-Peer Systems

Khaled M. Banafaa, Ruixuan Li, Kunmei Wen, Xiwu Gu, and Yuhua Li

Huazhong University of Science and Technology, Wuhan, China
Outline

• Background
• Motivation
• Keyword-matched Skyline in P2P
  – Chisky algorithm
  – Nk-sky algorithm
  – Ck-sky algorithm
• Performance Evaluation
• Conclusion
Skyline

- A **Skyline** is the set of all non-dominated tuples.
- A **skyline point** is a point that is not dominated by any other point in all dimensions.

*In general, the domination in one dimension is the user preference in that dimension (e.g. cheaper, shorter distance, and lower mileage).*
Problems in Skyline

• **Traditional skyline** may only give the skyline points that are not dominated by any other point in all dimensions.

• However, a user may only be interested in skyline for those points with some features.

• The user preferred features can be represented by some specific keywords.
A Motivating Example

The black-filled squares are the keyword-match skyline for the query $Q_{ks}(\text{cruiser control})$.

The black-filled circles are the usual skyline points.
Related Work

• **Distributed skyline algorithms** do not consider keywords.
  – Feedback-based distributed skyline algorithm (FDS)
  – Distributed SkyLine query (DSL) and SkyFrame
  – Parallel Distributed Skyline (PaDSkyline) and SkyPlan
  – **isky**: skyline in structured P2P network [ICDCS 2008]

• **Keyword-matched skyline** in centralized systems using R-tree [Choi, H. et al., Info. Sci. (2013)].

• **Traditional keyword search queries** in distributed systems and P2P systems ignore skyline incomparability and pruneability features.
Contributions

- **Bloom filters** are used to figure out the candidate peers for query keywords with *cover-set* tuples and nodes.

- **Keyword-matched skyline** algorithms are designed and implemented in P2P systems.

- Experiments have been carried out and show that the proposed approaches resulted in reduction of traversed peers while preserving progressiveness.
A tuple $\mathbf{t}$ in a $d$-dimensional space $D_d$ is defined as $<V,W>$

$V = (v_1, v_2, ..., v_d)$: a value vector of $d$-numerical values

$W = (w_1, w_2, ..., w_k)$: a set of $k$ keywords for the tuple $\mathbf{t}$. 
Problem Definition (II)

• Given a set of query keywords $W$ and a dataset $D_d$, a keyword-matched skyline query denoted as $Q_{ks}(D_d,W)$, retrieves the set of skyline tuples whose each textual attribute contains all words of $W$.

$$Q_{ks}(D_d, W) \equiv Q_s(Q_k(D_d, W))$$

$Q_{ks}$: keyword-matched skyline query
$Q_s$ : skyline query
$Q_k$ : keyword query
Ch-isky

- A **modified isky** is used to calculate keyword-matched skyline.

- **Filters**
  - *Skyline Filter*
    \[ SF_{\text{global}} = \min_{x \in S}(x_{\text{max}}) \]  
  - *Volume of Dominating Region (VDR)*
    \[ VDR_p = \prod_{i=1}^{d}(b_i - p_i) \]  
    - \( b_i \): Max value in dimension \( i \)
    - \( p_i \): Point value in dimension \( i \)

**Algorithm 1** Ch-isky: Chord-based isky algorithm

1: Input: MinMax-filter, VDR, Querying-Peer, keywords
2: BEGIN
3: if QueryingPeer then
4: for each peer \( P \) includes an integer (1 to \( D \)) in its period do
5: send Keyword-matched-Skyline-Query
6: end for
7: end if
8: Calc-key-matched-sky-using-VDR-MinMax()   
9: Calc-
10: Send
11: if min(nextPeer) then
12: send-query-to-next-peer(MinMax,VDR)
13: end if
14: END

All peers are candidates. Keywords are not used to select possible peers which could have the skyline points.
New Approaches

• **Data space** is partitioned using value attributes.
  – Minimum value preference is assumed.

• **Keywords** are hashed using Distributed Hash Table (DHT).

• A **query** undergoes two stages
  – Candidate peers are discovered using Bloom Filters on the keywords.
  – Traversing the peers for skyline in a way that allows pruning and progressiveness.
Node-based Keyword-matched Skyline Algorithm (Nk-sky)

• **Setup time:** Each peer sends the pair (peer#, keyword) to the responsible peer (using the keyword hash function).

• **Query Time:** Once a query is triggered, a Bloom filter is used to find out all peers with the query keywords.

  – False positives can be in the results but they do not affect the correctness of the skyline results. (theorem).
False Positives

- False positives are a result of
  1. Bloom Filters: the false positives $p_{fp}$
     \[ p_{fp} = 0.6185^{m/n} \]
     - $m$: Bloom filter bits
     - $n$: the number of elements in the set.
  2. A candidate peer may have no point that satisfy query keywords.
    - Because the node keywords are used.
      - (e.g. A peer can have $(k1,k2,k3,k4)$ with points: $p1(k1,k3)$ and $p2(k2,k4)$. A query can have $q(k1,k2)$. No point satisfies the query).
Ck-sky: cover-based keyword-matched skyline algorithm

- **Ck-sky** comes to minimize the number of false positives.

- In the Setup stage: Only cover-based keyword points (P_{ck}) are hashed.

  \[ P_{ck} = \{ p_i \in P \mid \forall p_j \text{ in that peer } p_i.w \subset p_j.w \} \]

---

**Algorithm 2** Ck-sky: cover-based keyword-matched skyline algorithm

1: Input: MinMax-filter, VDR, QueryingPeer traversed
2: BEGIN
3: if QueryingPeer then
4: /* first stage */
5: Peers-to-be-traversed = get-candidate-peers(BloomFilter())
6: for all P ∈ Peers-to-be-traversed do
7: if p closest above or equal peer to an integer (1 to D) then
8: send keyword-match-skyline-query to nextPeer
9: end if
10: end for
11: end if
12: /* Second stage */
13: Calc-keyword-matched-skyline()
14: Calc-VDR-and-MinMax()
15: Send-results-to-query()
16: nextPeer = closest-candidate-peer-in-an-increase-order
17: if min(nextPeer) \geq \text{MinMax} then
18: send-query-to-next-peer(MinMax,VDR)
19: end if
20: END

- Each keyword in the query is hashed and the bloom filter for each keyword is sent to
- In parallel, the peers are checked using Theorem 1
- Skyline Calculations use the filters: VDR to prune points within a peer and MinMax to prune peers.
- The query along with the VDR and MinMax are sent to the next candidate peer in a clockwise direction.
### Experimental Settings

#### Table 1. Parameter settings in the experiments

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cardinality (N) of tuples</td>
<td>100k, 200k, 400k, 600k, 800k, 1M</td>
</tr>
<tr>
<td>Dimensionality</td>
<td>2, 3, 4, 5</td>
</tr>
<tr>
<td>The number of query words (k)</td>
<td>1, 2, 3, 4, 5</td>
</tr>
<tr>
<td>Zipf skew factor (θ)</td>
<td>0.0, 0.2, 0.4, 0.6, 0.8, 1.0</td>
</tr>
<tr>
<td>Distribution (for values)</td>
<td>independent, correlated, anti-correlated</td>
</tr>
<tr>
<td>Tuple’s keywords</td>
<td>6</td>
</tr>
<tr>
<td>Network size (no. of peers)</td>
<td>100, 1000, <strong>2000</strong>, 3000, 4000</td>
</tr>
</tbody>
</table>
As we can see, the nk-sky reduces the number of visited peers. The Ck-sky reduces the number of false positives resulting in more reduction in visited peers.

In the Correlated data sets, the visited peers are less than the other because less number of skyline points are expected. This also affects other measures as we see later.

The Ch-isky and the nk-sky reduction in the visited peers as the tuples increase is due to reduction in the false positives, the high probability of finding peers with the query keywords.
Skyline points increase as dimensions increase.
As network size increases, the skyline points are expected to be distributed among more peers. This results in more visited peers.
As query keywords increase, the number of positive peers decrease. This is reflected by ck-sky. Nk-sky, on the other hand, slightly reflect it.
Traversed Peers vs. Skewness

Zipf does not have a very big effect on the visited peers. It has a slight effect in anti-correlated data sets because more points are expected for the skyline and the pruning is less effective.
Conclusions

• This paper addresses keyword-matched skyline in peer-to-peer systems.

• Node and tuple-based algorithms (Nk-Sky) are designed to solve keyword-matched skyline in P2P systems.
  – The algorithms use DHTs functions and Bloom filters to minimize the number of traversed peers.

• Ck-sky, a cover-based keyword-matched skyline algorithm, can greatly reduce false positives peers resulted from Bloom filters.
Thanks for your attention

Contact information:

• Ruixuan Li

• Huazhong University of Science and Technology

• rxli@hust.edu.cn

• http://idc.hust.edu.cn/~rxli/