An Optimal and Progressive Algorithm for Skyline Queries

Dimitris Papadias, Yufei Tao, Greg Fu and Bernhard Seeger

Abhishek Srivastava
Student ID: 861307778

February 23, 2017
CS 236, Winter 2017

The problem:

The paper discusses the necessity of fast evaluation of early skyline points. Nearest Neighbor being the existing fastest method. But it suffers from multiple problem such as redundant traversing, large main memory necessity and duplicate eliminations for high dimensions.

The contribution:

The authors propose *Branch and Bound Skyline* Algorithm, an extension of nearest neighbor to overcome its shortcomings. It improves over multiple variants of nearest neighbor techniques not only in terms of I/O and CPU cost but also reduced duplications. Similar to NN it also does not depend on indexed data-structure used.

The method:

The Authors approach is similar to NN by partitioning the data recursively. They used R-tree for implementation in this paper. L1 Norm is used for computing distance i.e *mindist* of a point is equal to sum of its point with *mindist* of its intermediate Bounding Region. Pruning is applied to the R-tree during traversal of the tree.

Starting at Root, child nodes are insert in the heap. Node with minimum *mindist* from heap is popped and its children are insert in the heap. Repeat till it reaches leaf node which belongs to skyline points, in further expansion we prune based on skyline points dominance region. A node is expanded only if it dominates skyline points otherwise marked pruned with its child nodes.

Authors stated following Lemma used in the algorithm:

- Nodes are visited in increasing order of their *mindist*.
- Data points classified as skyline points during traversal remains as a skyline points in the final list.
- Every leaf node gets examined unless its ancestor in marked as pruned.

These Lemma provide few properties to the BBS algorithm such as instant first skyline output and more skyline points as execution proceeds, No false misses and false hits, does not favor any specific points in any dimensions and useful for any kind of dataset distribution and dimensionality which used hierarchical index structure. Extensions of the skyline queries which can be used with BBS such as : *Ranked Skyline Queries*, *Constrained Skyline Queries*, *Dynamic Skyline Queries* and *Enumerating and K-dominating Queries*.

Comments:

The paper extends NN algorithm and provide BBS algorithm which improves upon its shortcomings. Authors compare BBS efficiency and effectiveness with NN using different parameters such as independent and correlated databases with various dimensionality and cardinality. Progressive behavior and Constrained skyline queries performance is also measured.

However, Some of drawbacks of the algorithms were:

- Use of heap cause additional cost with insertion and deletion in the case when data is highly dense and is highly dimensional as well i.e mindist calculation cost.
- Efficiency in low memory system and very large R-tree with high fan-out can cause high I/O cost because of the need to store data on disk.