International Journal of Engineering & Technology, 7 (2.7) (2018) 686-690

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

A Novel Filtered Based Grid partitioning multiple reducers skyline computation using Hadoop framework

P.Venkateswara rao [1*], Dr. Mohammed Ali Hussain[2]

[1]Research Scholar, Department of Computer Science Engineering, K L E F, Vaddeswaram, Guntur.
[2]Professor, Dept of Electronics and Computer Engineering, K L E F, Vaddeswaram, Guntur
**Email: pvrao.pd@gmail.com*

Abstract

Due to the exponential growth of multi-dimensional skyline objects, the computational memory and efficiency of the traditional skyline measures also increases on large spatial datasets. Skyline query computation has attracted a major research problem recently, due to its exponential time and space complexities over imbalanced datasets. A large number of sequential skyline query processing models have been implemented to evaluate the spatial pattern discovery on limited datasets. It is practically expensive and time consuming process to predict various spatial patterns from a large spatial candidate sets. Another major limitation with the sequential spatial pattern mining models is that a large number of spatial candidate sets are generated with duplicate event sets. In the proposed model, a novel parallel skyline processing using MapReduce framework is implemented on large spatial uncertain datasets. In this model, a filtered based k-nearest neighbor approach is used to eliminate the sparsity or empty patterns using the hadoop framework. Experimental results proved that the proposed model has high computational efficiency in terms of time and candidate sets are concerned.

Keywords: *Parallel Skyline processing, Spatial data mining's Hadoop, GPMRS, Map Reduced.*

1. Introduction

The rapid growth of the number of spatial data repositories on the internet is inherently distributed on multiple sites. These multiple sources of databases are stored in a central repository as Heterogeneous spatial database. Also, the continuously growing in the features and size of the databases will lead to many computational issues in the traditional skyline processing algorithms. Hence, there is a significant need to find essential spatial frequent patterns and infrequent patterns and its relations among heterogeneous databases. Also, it is necessary to discover hidden knowledge from those databases to improve the decision making. Finding the frequent skyline points is an essential part of all spatial mining approaches. Ensemble under-sampling technique is considered as a perfect solution for this skewed class distribution with limited data size. Due to memory and time limitations, an evolutionary approach can't be implemented in high dimensional applications. In order to resolve such types of problems, novel divide-and-conquer techniques are implemented using MapReduce scheme for large spatial datasets with limited dimensions. According to this technique, the whole dataset is divided into numbers of subsets of data for parallel processing. In the high dimensional data, divide-and-conquer methods are failed to find the majority and minority spatial problems due to high dimensionality issue. Traditional parallel skyline pattern mining models consider a significant amount of runtime due to medium to high database scans. Here, the number of candidate sets generated during the pattern mining process plays a vital role to improve the runtime and space complexity.

The skyline query processing is vital in many multi-criteria decision-making applications. Taking the case of two points in a n-dimensional space to be A and B. A dominates B if B is not better than A in all dimensions and A is better in at least one aspect compared to B, this is in accordance to the preferences given by users. Let a set of these points be S. The skyline set in S is a subset of S. The subset of S is made up of all the points that are not dominated by other points in S. Uncertain streams of data are monitored or analyzed effectively by using skyline operator.

For example hotels (a, b, c, d, e) close to the beach, they are considered skyline objects if none of the hotels are either having a lower price if they have the same distance from the beach, or if none of them is having a shorter distance from the beach with the same price, or there is none better in both distance from the beach and price than the hotels. These computations by skyline are complex on the evaluation of large datasets. Due to this, parallel skyline algorithms have been studied with an emphasis on the skyline additive property. A large dataset can be partitioned into smaller subsets. A local skyline evaluation is performed on each individual subset. After this, a global skyline computation is performed for a final check. In the parallel skyline computation, the dataset is

first partitioned in a set of subsets to the available machines. There is a similarity in structure between each of the partitions and the original dataset. An R-Tree indexing structure is then used by each machine to process the skyline over its local data.
Traditionally, there is one central server referred to as the coordinator. The coordinator controls a set of N servers. The coordinator is responsible for distributing the processing activities to the N servers due to high skyline computations, memory requirement, high dimensionality and cardinality. Each individual server computes the skyline over its local data then it gives the output to the coordinator. The coordinator then combine the result then calculates the global skyline result in a sequential approach. These sequential skyline computational models require high computational server capabilities along with memory, storage constraints.

Apache Hadoop is an open-source distributed framework used to process a large amount of data. Map-Reduce is a programming model in Hadoop framework, which is used to process a high dimensional dataset in parallel mode across multiple cluster nodes. This framework is designed and implemented to automatically divide large data into small segments, each of which can be executed on any cluster node and to handle node failures efficiently. It contains three essential components: Hadoop Distributed File System (HDFS), Hadoop Core and Map-Reduce. HDFS can store large data and partition these data into smaller blocks of 64MB each. Hadoop Core is responsible for providing efficient resources which are consumed by Hadoop components. To improve the performance of the single machine processor, an efficient Map-Reduce framework is used to develop large-scale applications, which are independent of hardware and distributed environment [1]. All the skyline query computational operations are performed using the MapReduce on multiple clustered machines. It gives an effective interface to support parallelization along with very high performance.

2. Related Work

Y. Cheng, et.al. implemented an ordered neighbourhood method for mining co-location patterns [2]. This technique basically depends upon the concept of maximal clique enumeration approach over spatial data. In this work, they presented a new technique known as Maximal Clique Enumeration Based on Ordered Star Neighborhood to filter the spatial co-locating objects and patterns.

SKY-MR is a Map-reduce algorithm proposed by Park [3]. The SKY-MR use spatial dataset and builds a sky-quad tree for locating the dominated sample regions before starting Map-reduce. The main aim of the Sky-quad tree is to repeatedly break down the data space into sub-regions for skyline object filtering. In this model, cluster nodes that manage the regions around the corners of data space often get more data than the skyline computation cluster nodes. In addition to this, the updating of the sky-quad tree can lead to high computation overhead in the case of large-scale datasets.

The grid partitioning based single reducer skyline computation in Map-reduce was proposed by Mullesgaard[4]. They divided the data space into grid partitioning scheme and represented these partitions using bit-string. In this model , a large number of sparse or null patterns are detected during the local skyline computation.

F. K. Deeb and L. Niepel proposed a new approach for detecting spatial co-location patterns efficiently [12]. This technique finds each and every spatial co-location pattern by using group of instances. Initially, a neighbourhood detection process is performed in order to identify individual co-location pattern of size 2 using KD-tree approach. KD-tree has several distinguished characteristics like:- tree generation and searching queries. An efficient pruning technique used in this model is participation index. Participation index is used to discard all non-prevalent patterns in the spatial candidate sets.

F. He, X. Deng, J. Fang introduced a faster technique in order to mine spatial skyline co-location patterns [5]. In this paper, they used a new spatial co-location mining algorithm and termed it as Grid-based technique. They implemented the skyline co-location computation using parallel programming for efficient computation. In this grid-based approach, all the spatial spaces are split into number of smaller cells and these smaller cells are known as grids. The appropriate length of cell is most vital because if the length is very small, it will induce additional computation overhead.

Y. Huang, J. Pei and H. Xiong developed an efficient mining technique for skyline co-location of rare events for spatial datasets [6]. Here, they identified all issues of spatial co-location mining patterns with rare events. They proposed a new measure known as maximal participation ratio (maxPR). maxPR gathers spatial co-location patterns with rare features. Besides this, another new measure known as pruneMax was presented in order to exploit weak monotonicity. By using both maxPR and pruneMAX, interesting rare co-location patterns can be determined easily for skyline computations. This technique is capable to handle only Boolean spatial features. But, most of the real world features are categorical and continuous in nature.

Foto N. Afrati et.al [7], proposed parallel skyline query processing using the map reduce framework for limited datasets. In this model, they minimized the load balancing steps for optimal query processing. This model is not applicable to high dimensional spatial datasets.

Hyeonseung et.al,[8] proposed a parallel skyline processing model using multicore architectures. They implemented a new skyline computational measure PSkyline using BBS,SFS and SSkyline measures. Proposed model is not applicable to GRID based skyline objects due to large set of candidate sets and null patterns.

Akrivi Vlachou et.al,[9] proposed a two-phase Map-reduce-based skyline computational framework for limited size datasets. During the first phase, input dataset is partitioned into multiple subsets using data partitioning method. The noisy objects are filtered out from each subset by a mapper function. Filtered objects are processed using the angle-based partitioning. The output from the map function is further processed by the reduce function. The reduce function aggregates the objects from the map function then filters them out. The local skylines are computed in each angle-based partition during the second phase. Global skyline objects are produced by the reducer function after merging the intermediate results. To process the Probabilistic skyline computations, Pei et al. [10] proposed efficient models based on a grid pruning method to compute skyline probabilities for all the skyline input dimensions. They used multivariate probability density function in terms of parametric equation. Zhang et al. [11] proposed a top-k skyline computational model that addresses the problem of uncertain data, and implemented two efficient methods for data filtering and randomization process.

3. Proposed Methodology

A novel filtered grid partition based multiple reducer skyline framework is designed and implemented on high dimensional spatial datasets. Grid Partitioning schemes in the parallel skyline computation are used to handle large datasets for skyline query processing. The main issues identified in the traditional grid and angular based skyline computations are solved using the following parallel processing features:

Parallelism: In the parallel skyline computation, each cluster node immediately processes the skyline query and returns the output to the coordinator node for skyline filtering.

Scalability: The workload should be evenly spread across the participating nodes for skyline processing.

Intermediate results: In the Map-reduce framework, each node should hold local data along with local skyline processing results for candidate filtering.

Equal-sized local skyline sets: The global skyline points should be spread uniformly to all servers so as to ensure that each node contributes the same to the global skyline result set.

Flexibility: During skyline processing by the individual mapper and reducer nodes, each node should be able to use its individual skyline algorithm for computing its locally stored data points.

This framework is implemented on hadoop framework. In the first phase, the input dataset is evenly partitioned into multiple subsets for bitstring representation.

Multiple Map-reducer jobs are used to obtain the bit-string representation to each sub-string. This bit-string is useful in pruning out data partitions without skyline tuples and it also gives the general view of the input data. The original tuples are distributed into multiple mappers by the FMR- GPMRS. This FMR-GPSRS also performs the parallel computation of local skylines. Global skyline computations are performed in each reducer phase by gathering the local skylines from multiple mappers. The general F- GPMRS process is in three phases as shown in figure1, the bitstring representation for pre-processing phase, filtered GPMRS using Knn approach and the local and global skyline computation phase. An n x n grid is used to partition the data space during the preprocessing phase. In the d-dimensional space, the total number of partitions that identifies the grid cell dominance is nd. This is because each of the dimensions is split into n parts. In the Map phase, all the non-empty partitions which dominate a cell are marked "1" contrary to the other cells which are marked "0". A reducer is applied with local skylines from all mappers in the reduce phase. The global skyline is then computed by merging them.

Since the runtime time and storage space of skyline computation will grow dramatically when the input dataset is large, applying the F-GPMRS method could remove more candidate sets before skyline calculation and save more execution time in the entire processing.

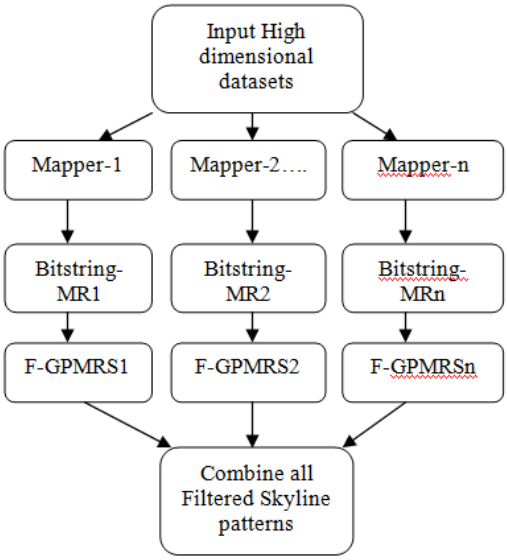

Figure 1: Filtered Grid Partition based multiple reducer skyline framework

This model is an extension of GPMRS [4] model using hadoop framework.

Step 1: Bitmap string representation:

In the grid partitioning, the non-empty tuple sets R of the partition are given importance for skyline computations. In this method, the n x n partitioning is represented as bitstring with d dimensions is given as

$$\text{BitString}_R[i] = 1, \text{if } p_i != \phi \\ = 0, \text{otherwise} \quad (1)$$

In the MapReduce framework, input data R is divided into multiple disjoint sets and each set is given input to mapper for bitstring computation using equation (1).

Step 2: Applying GPMRS[4] using multiple mapper and reducers. Here, independent partitioning group , local and global skyline computations are performed on each mapper and reducer phases.

Step 3: Filtering local skyline computations

Apply K-nearest neighbor algorithm to find nearest local skyline objects for global skyline computation.

For each local skyline object Oi in independent partition group
do
For each object Oi in IPG[]
Do
For each object Oj in IPG[] // where i!=j
Compute distance N_m^k using Manhattan distance.
Done
//Neighbor k-points in each severity level
N_m^k []=Find Top K- nearest objects from Sorted list of Dist;
Assign a class to p′ based on majority vote:
c′=argmaxy ∑(xi,ci) belonging to S, I(y=ci)
Done
Done

Step 4: Apply global skyline computation

Step 5: Merge global skyline computational results from the multiple reducers.

4. Experimental Results

Experimental results are implemented on 2.4GHz Intel core i5 processor with minimum 8GB RAM. We used Jama, Jfreechart, Scala, Spark third party libraries for the proposed filtered based GPMRS framework. We used spatial synthetic data with large number of instances and dimensions.

Sample Anti-correlation dataset:

```
4435 4778 2537 2198
617 2964 4258 3210
2223 4785 4464 705
4835 1948 4087 1411
```

```
3595 4261 1191 3465
2213 4890 4310 2050
1070 3207 4339 3830
2383 4433 3675 4032
4376 4158 510 3503
4662 3567 4305 920
3650 1105 4295 4825
2352 4211 2219 3725
4691 3227 3703 3484
4450 3086 356 3100
3923 2847 4036 1685
3645 1539 1737 3963
3187 4418 1369 2744
1880 2316 3527 4550
3807 2463 397 4080
2754 3757 4643 3201
4849 1472 4825 1968
4926 3350 4255 3343
```

Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
2 s	6/6		7.2 KB	65.7 KB	
4 s	6/6		83.2 KB		
0.9 s	6/6		688.8 KB		
0.2 s	6/6			8.3 KB	
0.8 s	6/6			65.7 KB	8.3 KB
3 s	6/6				65.7 KB
2 s	6/6				

Figure 2: Computational efficiency of the proposed model for 100000 skyline objects.

Figure2 describes the input, output skyline computational efficiencies in MapReduce framework.

Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
2 s	1/1 (1 skipped)	6/6 (6 skipped)
4 s	1/1	6/6
0.9 s	1/1	6/6
4 s	3/3	18/18
2 s	1/1	6/6

Figure 3: Computational duration of the proposed FGPMRS for 100000 skyline objects in hadoop framework.

Data size: 100000 records

Table1: Comparison of proposed model to the existing models in terms of runtime, dimensions and average storage are concerned.

Model	Average Runtime (Sec)	Dimensions	Average Storage (bytes)
Angle based Skyline	67	6	21388
GPMRS	36	6	16488
FGPMRS	27	6	6386

Table 1, describes the computational efficiency of the proposed model to the existing models in terms of runtime, dimensions and storage are concerned. From the table, it is clear that the proposed model has less storage space and less runtime compared to the traditional models on 100000 records.

Datasize: 100000 records

Table 2: Comparison of proposed model to the existing models in terms of number of reducers, dimensions and empty patterns.

Model	Reducers	dimensions	Avg. Filtered Patterns
AngleSkyline	6	5	56
GPMRS	6	5	42
FGPMRS	6	5	12

Table 2, describes the efficiency of the proposed filtered based GPMRS model in terms of number of reducers and null patterns are considered. From the table 2, proposed model has less null patterns compared to the existing models.

3. Conclusion

We study growth of the computational memory and efficiency of the GPMRS is increased with respect to increasing growth of multidimensional skyline objects. Further we discussed main issues identified in traditional grid and angular based skyline computations and are solved using parallel processing feature. To implement parallel processing we designed a Novel Filtered based MapReduced Grid partitioning on high dimensional spatial datasets. This is an extension of GPMRS model using hadoop framework. This F-GPMRS is implemented in three phases shown in figure1. This F-GPMRS method could remove more candidate sets before skyline calculation and save more execution time in entire processing. The experimental result shows average runtime of F-GMPRS is 27 seconds, and average storage is 6386 bytes for 100000 records contains 6 dimensions, this is very less compare to average runtime of grid based and angle based Skyline performance. Scope of this paper is by improving proposed algorithm to with stand feature requirements like increasing dataset size along with dimensions.

References

[1] T. Lappas and D. Gunopulos, "Efficient confident search in large review corpora," in ECML/PKDD (2), 2010.

[2] Yang Cheng; Zhang Tianjun; Lu Junli," A Maximal Clique Enumeration Based on Ordered Star Neighbourhood for Co-location Patterns", in 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics, 2013,pp. 164 - 167.

[3] Yoonjae Park, Jun-Ki Min and Kyuseok Shim, "Efficient Processing of Skyline Queries Using MapReduce", IEE Transactions on Knowledge and Data Engineering, 2017.

[4] Mullesgaard, Kasper; Pederseny, JensLaurits;Lu, Hua;Zhou,Yongluan. Efficient Skyline Computation in MapRe-

duce. Proc. 17th International Conference on Extending Database Technology(EDBT), Athens, Greece, March 24-28, 2014.EDBT,2014. pp. 37-48.

[5] Fei He1,2 , Xuemin Deng3 , Jinyun Fang1," A fast approach for spatial co-location pattern mining", 2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS,2013,pp.3654 – 3657.

[6] Yan Huang·Jian Pei·Hui Xiong , "Mining Co-Location Patterns with Rare Events from Spatial Data Sets", Geoinformatica, 2006, pp. 239–260.

[7] Foto N. Afrati , Paraschos Koutris , Dan Suciu , Jeffrey D. Ullman, "Parallel skyline queries", Proceedings of the 15th International Conference on Database Theory, March 2012, pp. 26-29.

[8] Parallel Skyline Computation on Multicore Architectures, Sungwoo Park, Taekyung Kim, Jonghyun Park, Jinha Kim, Hyeonseung Im, Department of Computer Science and Engineering, Pohang University of Science and Technology Gyeongbuk, Republic of Korea, pp. 790-784.

[9] Katja Hose, Akrivi Vlachou, "A survey of skyline processing in highly distributed environments", The VLDB Journal (2012), PP: 359–384.

[10] Jian Pei, Bin Jiang, Xuemin Lin, Yidong Yuan ,"Probabilistic skylines on uncertain data", Proceeding VLDB '07 Proceedings of the 33rd international conference on Very large data bases, PP: 15-26.

[11] Ying Zhang, Wenjie Zhang, Xuemin Lin, Bin Jiang, Jian Pei, Ranking uncertain sky: The probabilistic top-k skyline operator, Information Systems 36(2011), PP: 898-915

[12] Fadi K. Deeb; Ludovit Niepel," A methodology for discovering spatial co-location patterns", 2008 IEEE/ACS International Conference on Computer Systems and Applications , 2008, PP: 134 – 141.

[13] P. Venkateswara rao, Dr.Mohd Ali Hussain, "Mashup service implementation on multi-cloud Environment using Map Reduction Approach", Journal of Advanced Research in Dynamical and Control Systems, 2017, PP: 758-767.