International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 1

Top-K Dominating Queries on Incomplete Data with Priorities

Dr. Prabha Shreeraj Nair
Dean Research, Tulsiramji Gayakwade Patil College of Engineering and Technology, Nagpur, India

Prof. Dr. G. K. Awari
Principal, Tulsiramji Gaikwad-Patil College of Engg and Technology (NAAC Accredited), Nagpur, India

ABSTRACT

Top-K dominating query returns the k objects that are dominated in a dataset. Finding dominated elements on incomplete dataset is more complicated than in case of complete dataset. In the real- time datasets the dataset can be incomplete due to various reasons such as data loss, privacy preservation or awareness problem etc. In this paper we aims to find top-k elements from an incomplete dataset by providing priority values to each dimension in the data object. Skyline based algorithm is applied for that purpose. Since the priority value is used while determining the dominance this method return the most suitable and efficient result than other previous methods. The output will be more preferable according to the users purpose.

Keywords: *TopKQuery, Dominance, Skyline, Bucketing, Priority, Local Skyline*

1. INTRODUCTION

Top-K dominating query aims to find the top elements from a dataset. Movie recommendation system is a practical example for finding the top elements. That system will return the top movies from a set of movies based on ratings by different users. In real-time applications the datasets may be incomplete due to various reasons which will lead to uncertainty in data. Assuming the unknown value is not an accurate method for solving this problem.

Finding top elements from complete dataset is quite easy because we have all dimensions available. We just need to compare the values. But when the values are not completely available it becomes difficult. Dataloss or other technical or non technical reasons may reside behind the missing values. Consider an object in a dataset S with four dimensions A(1,-,2,1).For A there is four dimensions but only three values are available and „-„ indicates a missing value. So for A the second dimension value is missing. This is what means by the missing dimension value. A sample dataset is shown below.

A1(2,-,3,1)
A2(-,1,1,1)
A3(1,2,3,-)
A4(-,6,1,1)
A5(1,1,-,3)

To output the top elements we need to define the dominance relationship on incomplete data.

Definition: *(dominance relationship on incomplete data [1]).*

Given two objects o and o' in a dataset S. o dominates o' (i.e., o < o') if the following conditions hold: I) for every dimension i, either o. [i] is less than o'.[i] or at least one of them is missing.

There is at least one dimension j in which both o. [j] and o'. [j] are observed and o.[j] is less than o'.[j].

The dominance is determined by comparing one dimension value of an object with the same dimension value of the second object. If one of them is missing then that dimension will not be considered. Dominance is considered based on the small value. That is 1is dominated than 2. An object is dominating another object only if it dominates the second object in all available dimensions. Otherwise it is ignored. If in one dimension the first object dominates the second and in second object dominates the first in another dimension will make difficulty in determining the dominance. Here is the importance of priority occurs. In our work a priority value considered for each dimension. That priority value can be assigned by the user. This priority value can be used while the uncertainty cases like stated above occur.

Priority value can be set as 1, 2, 3.. for each dimension. The least number indicates the highest priority. When the uncertainty condition occurs the highest priority dimension will be selected for the decision making. The object which dominates at the high priority dimension can be considered as the dominating element. If the decision cannot be taken using the high priority element ie, if the value is missing on that dimension then we can move to the second priority dimension and continue the process. Using the priority value will help in determine the user‟s preference about the output. Because if user gives high priority to the second dimension then the output will also reflect the priority. That is the output will more depend on the second dimension.

2. RELATED WORKS

This section includes the descriptions some related work on this topic.

In [1] four algorithms are proposed for finding top-k dominating elements from incomplete data. Extended Skyband based algorithm (ESB) uses the same skyline based approach used in [2].Another algorithm Upper Bound based(UBB) uses and a MaxScore value which is calculated for each dimension based on dominance. The third algorithm BIG(Bit map Index Guided) uses calculations based on bit map index and a MaxBitScore like MaxScore. Improved BIG algorithm uses a compression technique CONCISE to compress the bitmap index vertically and a binning strategy to cut down the bitmap storage consumption horizontally.

A restaurant recommendation system is implemented using preference query over incomplete information [3]. [2] Restaurant recommendation system have different interaction module like query submission, result explanation and dataset interaction. The user can submit query by specifying interest and constraints like region, price level etc for the restaurant. Query will be processed at the server and results will be returned. Users can write review about restaurant and rate them. Based on the rating the restaurant details will be updated. At the server side the dataset is stored in PostgreSQL database. They integrate the PostgreSQL database by integrating two algorithms lksb[2] and UBB [1].Explaining the query will help the user to understand why an empty result set or mismatch occurred. This is an example of applying top-k queries on incomplete data.

For evaluating top-k queries on incomplete data the common technique used in various papers are skyline based approach. In the skyline approach basically steps as bucketing, local skyline etc are implemented. Bucketing means sort the data into different buckets based on the bit number of its dimension. That is if the item A(1,-,5,6) is in the dataset its bit number will be 1011.Like this dataitems with same bit numbers will be added to the same bucket .Local skyline will be the dominating items from each bucket.A model for processing skyline queries on incomplete data is proposed in [5].The proposed model have 4 components ,data clustering builder, group constructor and local skylines identifier ,k-dom skyline generator and incomplete skyline identifier. This method divides the database into different clusters grouping the data items in the clusters based on local skyline.

In evaluating top-k queries on incomplete data stream [6], two algorithms are proposed. Sorted List Algorithm (SLA) and Early Aggregation Algorithm (EAA) describe tracking top-k items over multiple data streams in a sliding window. Sort-based Incomplete Data Skyline algorithm (SIDS) [7] also uses skyline algorithm. In SIDS first the dataset is presorted in non- increasing order of each dimension, and then each dimension is choosed in round robin fashion for comparison. On each iteration the dominated items are removed from the set, at the end an item which is not removed and processed k times are returned.

3. PROPOSED SYSTEM

Architecture of the proposed system is shown in fig3.1.

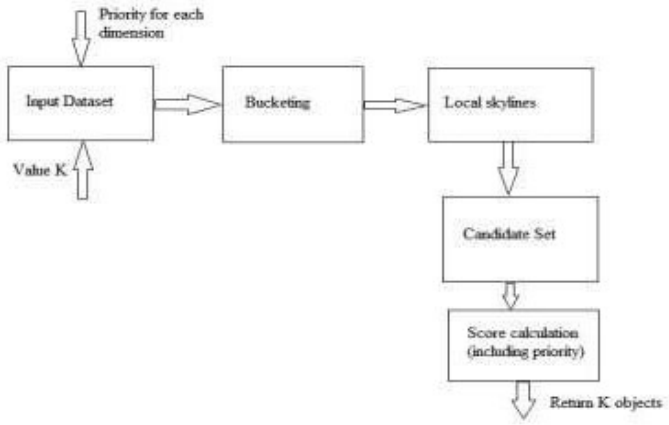

Fig 3.1: Architecture

The incomplete dataset is taken as the input. Priority values are assigned for each dimension.

Skyline based algorithm is used here.

Algorithm: Sk yline Algorithm with Priority

Input: an incomplete data set *S*, a parameter *K* and Priorities for each dimension.

Output: the result set *Sg*.

1: Initialize *Sc* &*Sg* as 0.

2: for each object o ∈ S do

3: insert o into a bucket O based on βo (create if not exist)

4: for each bucket do

5: compare (,) for every , ∈ O

6: add K objects with highest score to the

7: add all o in

8: for each o ∈ Sc

9: compare (,) for every , ∈ *Sc*

10: add K objects objects in Sc having the highest score to the *Sg*

11: return *Sg*.

The first step in the method is bucketing. In this step each object in the dataset is grouped into different buckets based on a bit number(βo). For each dimension the bit value is calculated as, if the value is available then bit value is 1 otherwise it is 0. For example the bit number corresponding to A(1,-,2,1) is 1011. That is 1011.A bucket 1011 will be created and A will be added to that bucket. In this way the bit number for each object is determined and objects with same bit number are taken into one bucket. After categorizing each objects into buckets the local skylines( ) of each buckets are determined. Local skylines are K- objects with high score in each bucket.

Compare function is based on the dominance relation. compare (,) returns the score for . Consider two objects A(1,-,2,1) and B(2,3,-,3) while considering the dominance between A and B , A dominates B since A have smaller value in all the available dimension than B. Then the score of A become one. If A dominates another object C in all dimensions then score of A becomes 2.That is score of an object is the number of objects that are dominated by the particular object. While finding the local skyline only the objects in corresponding bucket will be considered for the score calculation.

In the score calculation stage if we have two objects O1(1,-,3,2) and O2(-,1,1,3). At the first dimension we have missing value in O2 and for second dimension in O1.So the third and fourth dimension determines the dominance. But at third dimension O2 dominates O1 and in fourth O1 dominates O2.According to the existing systems the dominance will not be valid in this case. But in this system the priority value is used at this point. If the third dimension have high priority than fourth then O2 will be considered as the dominating object. Else if fourth dimension has the high priority then O1 will become the dominating element and score increase accordingly. The priority can be set by the user as values 1, 2, and 3 etc...according to the number of dimensions. The least value indicates high priority.

Local skylines from every bucket will be added to a candidate set(*Sc*). For every object in the candidate set the score calculation is again conducted with every object in the dataset. That is, if 8 objects A,B,C,D,E,F,G and H are included in the candidate set then the 8 objects will be compared with the entire dataset and score will calculated for the them. Then K-objects with high score in the candidate set are returned as the output(*Sg*).

4. CONCLUSION

This paper is the first work which uses a priority value for Top-K dominating query. Skyline based algorithm with priority is used here for returning the top elements. This method can be used in systems like movie recommendation with user preferred priorities for more accurate outputs.

REFERENCES

1. Xiaoye Miao, Yunjun Gaor "Top-k Dominating Queries on Incomplete Data ", IEEE Transactions on Knowledge and Data Engineering,VOL. 28, NO. 1, January 2016.
2. Yunjun Gao ,Xiaoye Miao ,Huiyong Cui Gang Chen ,Qing Li, "Processing k-skyband, constrained skyline, and group by skyline queries on incomplete data", International Journal of Expert System with Applications, 2014.
3. Xiaoye Miaoa,Yunjun Gao," ^{2} :A Restaurant Recommendation System Using Preference Queries over Incomplete Information", Proceedings of the VLDB Endowment, Vol. 9, No. 13,2016 .
4. Mohamed E. Khalefa, Mohamed F. Mokbel,Justin J. Levandoski,"Skyline Query Processing for Incomplete Data" ,DTC Digital Technology Initiative programme University of Minnesota,2006.
5. Ali A. Alwan, Hamidah Ibrahim, Nur Izura Udzir, "A Model for Processing Skyline Queries over a Database with Missing Data", Journal of Advanced Computer Science and Technology Research, Vol.5 No.3, September 2015, 71-82.
6. Parisa Haghani, Sebastian Michel, Karl Aberer," Evaluating Top-k Queries over Incomplete Data Streams ",2009 ACM 978-1-60558-512.
7. Rahul Bharuka P ,Sreenivasa Kumar," Finding Skylines for Incomplete Data ",Proceedings of the Twenty-Fourth Australasian Database Conference (ADC 2013), Adelaide, Australia