Implementing Movie Supporter System Using Query Processing On Partial Data

Marella Suresh Babu[1], Mylapoor Madhu[2]
[1]*Student, Master of Computer Applications, SKIIMS, Srikalahasti, Andhra Pradesh, India*
[2]Asst.Professor, *Master of Computer Applications, SKIIMS, Srikalahasti, Andhra Pradesh, India*

***Abstract*- In this paper, we used Top-k Dominating (TKD) for partial query processing which returns the k objects that dominate the maximum number of objects in a given dataset. It combines the advantages of skyline and top-k queries, and plays an important role in many decision support applications. Partial data exists in a wide spectrum of real datasets, due to device failure, privacy preservation, data loss, and so on. In this paper, for the first time, we carry out a systematic study of TKD queries on Partial data, which involves the data having some missing dimensional value(s). Extensive experimental evaluation using both real and synthetic datasets demonstrates the effectiveness of our developed pruning heuristics and the performance of our presented algorithms. The purpose of this study is to develop a 'Movie Support System' with the help of Collaborative Filtering approach.**

***Index Terms*- Partial data, Query processing Clustering, Collaborative Filtering, Movie Support System.**

I. INTRODUCTION

An automated Recommendation System can help in this context by aggregating the opinions of a community of users to help individuals in that community more effectively identify content of interest from a potentially overwhelming set of choices. In this, the first supporter system Tapestry, coined the term "Collaborative Filtering (CF)", variants of which have become famous.

Generally, Collaborative Filtering (CF) is an algorithm that filters information for a user based on a collection of user profiles. Users having similar profiles may share similar interests. For a user, information can be filtered in/out regarding to the behaviors of his or her similar users. Nonetheless, we would like to highlight that existing work related to this query only focuses on complete data or uncertain data.

In a real movie support system, it is very common that the ratings from some users are missing; because a user tends to only rate those movies he/she knows. As a result, each movie is denoted as a multi-dimensional object with some blank (i.e., partial) dimensions. Therefore, the set of movie ratings is incomplete. As shown in Fig. 1, since the audience a2 watches the last three movies m 2 , m 3 , and m 4 but not the first one m 1 , i.e., Schindler's List (1993) ,a2 only rates movies m2 ,m3, and m4. Data incompleteness is universal, and querying Partial data has become more and more important recently. It has also triggered lots of efforts in the database community, including Partial data model, query evaluation, indexing, skyline computation, similarity search, top-retrieval etc.

ID	Film Name	Film Ratings from Audiences				
		a_1	a_2	a_3	a_4	a_5
m_1	Schindler's List (1993)	–	–	3	4	2
m_2	The Godfather (1972)	5	3	4	–	–
m_3	The Silence of the Lambs (1991)	–	2	1	5	3
m_4	Star Wars (1977)	3	1	5	3	4

Fig. 1: Example of a movie recommender system

At first glance, TKD queries on Partial data share some similarities with the skyline operator over Partial data, since they both are based on the same dominance definition. However, we would like to highlight that TKD queries on Partial data have a desirable advantage, i.e., its output is controllable via a parameter k , and hence, it is invariable to the scale of the Partial dataset in different dimensions. In addition, we want to emphasize the dominance relationship definition on Partial data is actually meaningful. Take movies m1 and m2in the recommender system depicted in Fig. 1 as an example. The audiences a1and a2only rate m2 but not m1, whereas the audiences a4 and a5only rate m1but not m2. Thus, we cannot determine the dominance

relationship between m1and m2 according to the rates from audience's a1, a2, a4, and a5.

To the best of our knowledge, this is the first attempt to explore the TKD query on Partial data. Although the TKD query over complete data or uncertain data has been well studied, TKD query processing on Partial data still remains a big challenge.

In brief, the key contributions of this paper are summarized as follows.

- We formalize the problem of TKD query in the context of Partial data. To our knowledge, there is no prior work on this problem
- We propose efficient algorithms for processing TKD queries on Partial data, using several novel heuristics.
- We present an adaptive binning strategy with an efficient method for choosing the appropriate number of bins to minimize the space of bitmap index.
- We conduct extensive experiments using both real and synthetic datasets to demonstrate the effectiveness of our developed pruning heuristics and the performance of our proposed algorithms.

II. SUPPORTER SYSTEM

Supporter systems form a specific type of information filtering (IF) technique that attempts to present information items (movies, music, books, news, images, web pages, etc.) that are likely of interest to the user.

Typically, a Supporter system compares the user's profile to some reference characteristics, and seeks to predict the 'rating' that a user would give to an item they had not yet considered. These characteristics may be from the information item (the content-based approach) or the user's social environment (the collaborative filtering approach).The most common algorithms that are widely used in Supporter systems are Collaborative Filtering. One of the most common types of Collaborative Filtering is item -to-item collaborative filtering (people who buy x also buy y), an algorithm popularized by Amazon.com's Supporter system.

III. COLLABORATIVE FILTERING

It is the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoint s, data sources etc. Applications of collaborative filtering typically involve very large data sets.

Collaborative filtering methods have been applied to many different kinds of data including sensing & monitoring data–such as mineral exploration, environmental sensing over large areas or multiple sensors, financial data –such as financial service institutions that integrate many financial sources ; or in electronic commerce and web 2.0 applications where the focus is on user data etc.

IV.TKD QUERY ON PARTIAL DATA

In this section, we first present three efficient algorithms, i.e., extended skyband based (ESB) algorithm to support TKD query processing on partial data.

4.1 Extended Skyband Based Algorithm

The intuitive method (denoted as Naïve) for the TKD query on Partial data is to compute the score of every object o by conducting exhaustive pair wise comparisons among the whole dataset, and to return the k objects with the highest scores.

However, this Naïve approach is inefficient due to the extremely large size of the candidate set and the expensive cost of score computation. Fortunately, the objects with observed attributes falling inside the same set of dimensions actually satisfy the transitive dominance relationship. To this end, we re-organize the objects into buckets.

Here, each bucket corresponds to a given subset of d dimensions, and it accommodates all the objects whose observed attributes fall in those d′ dimensions exactly. Accordingly, we present our first algorithm, namely, extended skyband based (ESB) algorithm, which uses local skyband technique to answer the TKD query over Partial data.

Here, we borrow the concept of k -skyband (KSB) query on Partial data. The KSB query is a variant of skyline queries, and it retrieves the objects dominated by less than k objects. Since the objects within the same bucket share the same bit vector, they can be regarded as a complete dataset in d ′-dimensional space. If we perform a KSB query for each bucket, the KSB query results collectively form a candidate set for a TKD query

Algorithm: A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.
Input: a Partial data set S, a parameter k
Output: the result set SG of a TKD query on S

1: initialize sets $S_C \leftarrow S_G \leftarrow \phi$
2: for each object o ∈S do
3: insert o into a bucket O based on $\aleph_0$ (create O if necessary)
4: for each bucket O do
5: $S_C \leftarrow S_C \tilde{U}$ KSB(O)
6: for each object o ∈ S_C do
7: update score(o) by comparing o with all the objects in S
8: add the k objects in SC having the highest scores to SG
9: return S_G

4.3. Upper Bound Based Algorithm:
Input: an Partial data set S, a parameter k, a pre-computed priority queue F sorting all objects from S in descending order of their Max Score.
Output: the result set SG of a TKD query on S

1: initialize sets $S_C \leftarrow S_G \leftarrow \phi$ and $\tau \leftarrow -1$
2: while F is not empty do
3: o ← de-queue(F)
4: if Max Score (o) ≤ τ then break //Heuristic1
5: else
6: score(o) Get _ Score(o)
7: if score(o) > τ or τ < 0 then
8: $S_C \leftarrow S_C \tilde{U} \{o\}$
9: if $|S_C| > k$ then

V.PROPOSED SYSTEM

We treat a data object with missing value(s) as an Partial data object, which follows the model introduced in that requires zero prior knowledge of missing dimensional value(s).
As pointed out explicitly in, "missingness is the state of being missing" for missing data and the missingness implies "a static state, not fluid or probabilistic one". Thus, the Partial data model (the one our work is based on) and the probabilistic concept (the one used by uncertain data) are two approaches for handling missing data. It is worth mentioning that, compared with uncertain data model, Partial data model has one significant advantage, i.e., it does not require any assumption on data correlation or prior knowledge. We consider Partial dataset where some objects face the missing of attribute values in some dimensions, and study the problem of TKD query processing over Partial data. In particular, a TKD query on Partial data returns the k objects that dominate the maximum number of objects from a given Partial data set.
TKD queries on Partial data share some similarities with the skyline operator over Partial data, since they both are based on the same dominance definition. To the best of our knowledge, this is the first attempt to explore the TKD query on Partial data. Although the TKD query over complete data or uncertain data has been well studied, TKD query processing on Partial data still remains a big challenge. This is because existing techniques cannot be applied to handle the TKD query over Partial data efficiently. Specifically, the R-tree-=aR-tree and the transitivity of dominance relationship used in traditional and uncertain databases are not directly applicable to Partial data. It is partially because R-tree=aR-tree could not be built on Partial data directly, since the MBRs of tree nodes do not exist due to the missing dimensional values of data objects. Also, the transitivity of dominance relationship does not hold for Partial data. In addition, the probability model of uncertain TKD queries is different from our model as mentioned earlier. Consequently, new efficient algorithms catered for Partial data are desired.

ADVANTAGES:
1.Trade the efficiency and computation cost
2.TKD query performs on incomplete data.

VI.MODULES DESCRIPTION

6.1.DOMINANCE RELATIONSHIP ON PARTIAL DATASET

Given two objects o and o0, if we have no information about their missing value(s), there is no clear judgment on which object is better for the dimensions with missing value(s). Therefore, we can only utilize the common observed dimensional values to decide the dominance relationship of those two objects.
It is worth noting that, there are other similar dominance definitions over incomplete data, such as missing flexible dominance (MFD) operator which is flexible, reasonable, and fair in many real-life applications.

6.2 CALCULATION OF SCORE VALUE

Given an PARTIAL DATASET S and an object o € S, the score of o, denoted as score (o), is the number of the objects o'€ S –(o) that are dominated by o.

6.3.TKD QUERY ON INCOMPLETE DATA

We first present three efficient algorithms, i.e., extended sky band based algorithm, upper bound based algorithm, and bitmap index guided algorithm, to support TKD query processing on incomplete data. Then, we propose an improved BIG (IBIG) algorithm based on BIG, to further minimize the space cost of bitmap index.

6.4.EXTENDED SKYBAND BASED ALGORITHM

The intuitive method (denoted as Naive) for the TKD query on incomplete data is to compute the score of every object o by conducting exhaustive pair wise comparisons among the whole dataset, and to return the k objects with the highest scores. However, this Naive approach is inefficient due to the extremely large size of the candidate set and the expensive cost of score computation.

6.5.UPPER BOUND BASED ALGORITHM

Motivated by the instability of ESB algorithm, we propose the upper bound based algorithm for supporting the TKD query over incomplete data, which utilizes the upper bound scores of objects to determine the access order of objects to reduce the candidate set size. In particular, UBB integrates the ranking process into the object evaluation, and enables an early termination of TKD query processing before evaluating all the candidates. Based on this intention, we present the concept of upper bound score that returns the maximum number of the objects that a specified object o dominates.

6.6.BIT MAP INDEX GUIDED ALGORITHM

Our proposed UBB algorithm limits the size of candidate set by utilizing upper bound score pruning technique for the TKD query on incomplete data. However, the upper bound score may be rather loose; thereby we have to derive the real scores for many objects (even the whole dataset) via exhaustive pair comparisons, which degrade search performance significantly. Furthermore, we also develop an improved version of BIG (denoted as IBIG) to minimize the bitmap storage cost via the bitmap compression techniques and an adaptive binning strategy.

6.7.IMPROVED BIG ALGORITHM

Although BIG algorithm can significantly improve the search performance of TKD queries over incomplete data compared with ESB and UBB algorithms, we are also aware that the bitmap index size (denote as costs) is rather large.

6.8.EVALUATION RESULTS ANALYSIS:

We verify the effectiveness of the bitmap compression and the binning strategy of bitmap index, and evaluate the performance of our proposed algorithms for TKD queries over incomplete data. All algorithms are implemented in Java SE7, and all experiments are conducted on an Intel Core i5 Duo 3.10 GHz PC with 4 GB RAM, running Microsoft Windows 7 Professional Edition.

VII. REQUIREMENT SPECIFICATION

7.1. Software Requirements

Language	:	JDK (1.7.0)
Frontend	:	JSP, Servlets
Backend	:	Oracle10g
IDE	:	my eclipse 8.6
Operating System	:	windows XP
Server	:	tomcat

7.2. Hardware Requirements

Processor	:	Pentium IV
Hard Disk	:	80GB
RAM	:	2GB

VIII. SYSTEM DESIGN

Software design is an iterative process through which requirements are translated into a "blue print" for constructing software. The design is represented at a high level of abstraction a level that can be directly translated to a specific data, functional behavior requirements.

- The design should exhibit uniformity and integrity.
- The design should be structured to accommodate changes.

- ❖ The design is not coding, the coding is not a design.
- ❖ The design should be reviewed to minimize.

IX. OUTPUT SCREENS

A Screen shot is an image taken by a person to record the visible items displayed on the monitor. It is usually, this is a digital image using the operating system.

1) Admin Login Page:

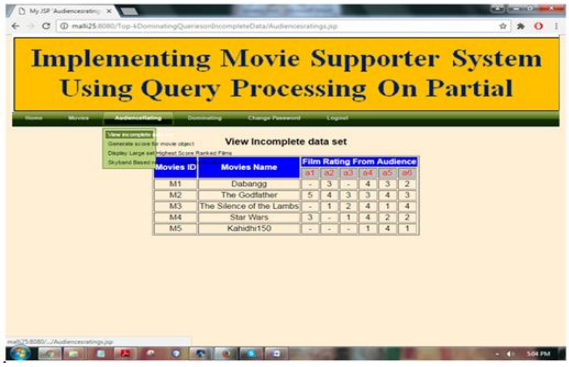
Description**:** The above screen is the admin login form to be create login form two textboxes is created to enter user name and password to click the the login button then displays a admin home page.

2)Add Movie Details:

Description: In The above screen is the add movie details. first we have to the open the admin home page then click the movies option after the two options are displayed .one is add movies To enter the film names, produced by, starring, music by, Release year, country, language after the click the add film button.

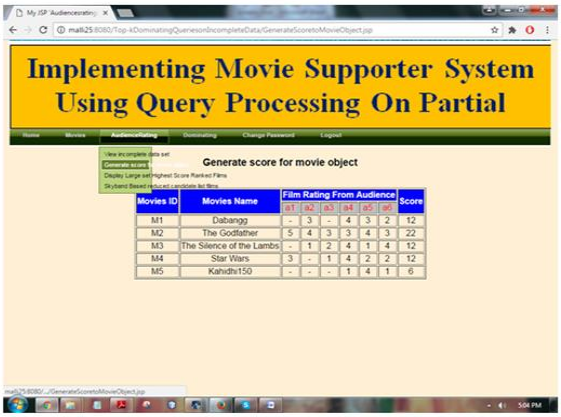
3)View Incomplete Data Set:

Description: In the above screen is display the view incomplete data set. First audiences give the rating in all films some audiences give the no rating then also form the incomplete data set.



4)Generate The Score For the Movie Object:

Description: In the above screen is Generate the score for the movie object. First display audience rating. Then after calculate the score for the all audiences. Total score for the movie objects.

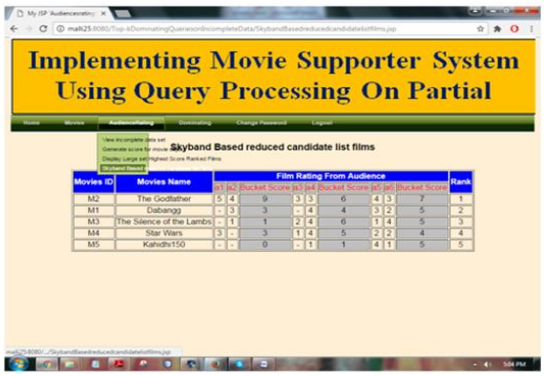


5)Display Large Set Highest Score Ranked Films:

Description: In the above screen is Display Large Set Highest Ranked Films. In the incomplete data from the audience rating then calculate the score for movie objects display the highest score for the movie then the movie give the ranking purpose.

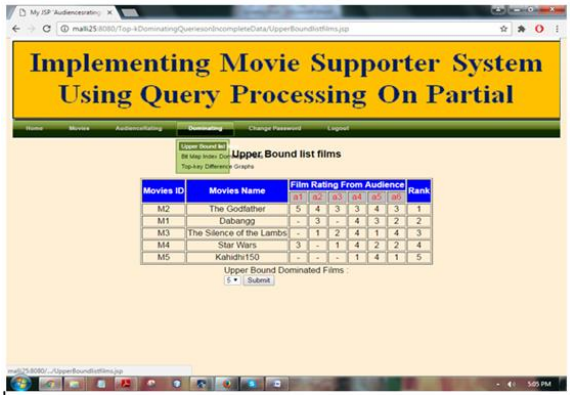
6)Sky Band Based Reduced Candidate List Films:

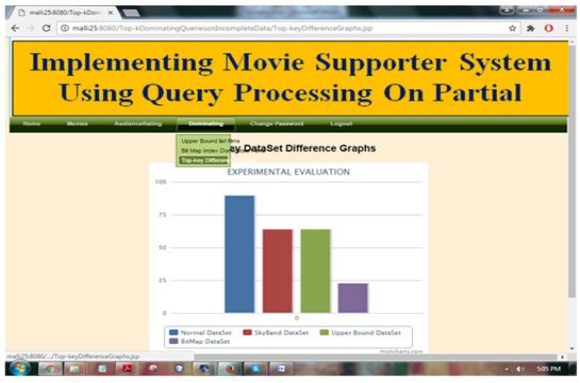
Description: In the above screen is Sky band based candidate list films. The sky band is used to the reduced the all candidate rating films .after the reducing calculate the score for the reducing films after give the ranking so as well as sky band is also calculate the bucket score. The bucket score depends on the ranking for highest films



7)Upper Bound List Films:

Description: In the above screen is Upper bound list films. Upper bound is also used to the display the most top of the dominated films at audience rating then after type the dominated films after click the submit button .





X. CONCLUSION

Consider the wide range of applications for Top- K Dominating (TKD) queries and the pervasiveness of Partial data, we, in this paper; study the problem of the TKD query on Partial data where some dimensional values are missing. To efficiently address this, when first propose ESB and UBB algorithms, which utilize novel techniques (i.e., local sky band technique and upper bound score pruning) to prune the search space.

REFERENCES

[1] P. Resnick and H. R. Varian, Special issue on recommender systems, Communications of the ACM, 40(3), 1997.

[2] D. Goldberg, D. Nichols, B. Oki, and D. Terry, Using collaborative filltering to weave an information tapestry, Communications of the ACM, 35(12):61-70, 1992.

[3] D. M. Pennock and E. Horvitz, Collaborative Filtering by Personality Diagnosis: A Hybrid Memory-and Model-Based Approach, In IJCAI Workshop on Machine Learning for Information Filtering, Stockholm, Sweden, August 1999. International Joint Conference on Artificial Intelligence.

[4] Mark O'Connor & Jon Herlocker, Clustering Items for Collaborative Filtering. Dept. of Computer Science and Engineering University of Minnesota Minneapolis, MN {oconnor,herlocke}@cs.umn.edu.

[5] By A. P. DEMPSTER, N. M. LAIRD and D. B. RDIN, Maximum Likelihood from Partial Data via the EM Algorithm. Harvard University and Educational Testing Service.

[6] M. E. Khalefa, M. F. Mokbel, and J. J. Levandoski, "Skyline query processing for Partial data," in ICDE, pp. 556–565, 2008.

[7] http://www.netflixprize.com/.