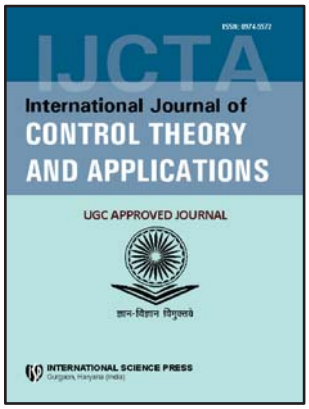


**International Journal of Control Theory and Applications**

ISSN : 0974-5572



**Volume 10 • Number 30 • 2017**

# A Glimpse on Top-K Dominating Query Processing on Incomplete Data

**V. Chandra Shekhar Rao[a] and Adeeba Tamkinath[b]**

[a]*Associate professor of computer Science and Engineering, Kakatiya institute of Technology & science, Warangal, Telangana, India, E-mail: vcsrao.kitswgl@gmail.com*

[b]*M.Tech scalar, Department of computer science and Engineering, Kakatiya institute of Technology & science, Warangal, Telangana, India E-mail:adeebatamkinath@gmail.com*

***Abstract:*** The objective is to return best K objects with the highest dominating score in a Dataset.Incomplete data may take place due to some reasons like human error, system crash etc. Merits of both Top-*k* Query and Skyline queries are put together to form a Top *k* dominating query. To solve this problem there are various algorithm for TKD queries on incomplete data upper bound score pruning, bitmap pruning, binning strategy is used in order efficiently carry out TKD query processing.

***Keywords:*** *Query, Dominate, Top k, Incomplete data, Dataset.*

## 1. INTRODUCTION

A Top-*k* Dominating Query binds the benefits of both top – *k* queries and Skyline queries. In a top – *k* query a user provides a ranking function to order the objects by their scores and thus retrieve top-k best objects, while A Skyline is defined as those points which are not dominated by any other point. A point dominates another point if it is as good or better in all dimensions and better in at least one dimension. In Top- *k* output size can be controlled by using parameter – *k*, and In skyline query you need not use any ranking function.

Let us consider a Dataset D let A($d_1$, $d_2$, $d_3$…. $d_n$), B($d_1$, $d_2$, $d_3$…. $d_n$) be the entities in a dataset D, where $d_1$, $d_2$, $d_3$….$d_n$ indicates dimensions of an entities.Entity A is set to be dominate object B when it is not bad in all dimensions or at least better one dimension while compared to B.Incomplete data indicates not having appropriate or necessary information. Let us take an entity X($d_1$, $d_2$,_,…,_,.$d_n$) where some of the dimensions are missing, so this is an incomplete object,where "_" indicates a missing dimension , a sample object X(3,_,2) which has three dimensions where one of it's the dimension is missing. Moreover,we apply dominance relationship only when object and comparing entities have common dimensions. For instance take two entities. A(1,_,4) and B( _,_,3) having three dimensions this can be comparable. But in case if A(1,_4) and B(_,4,_), there are no common dimension hence they cannot be comparable *i.e*; we cannot apply dominance relationship.

In a Real-time system, whenever people visit *e*-Commerce website,people only tend to rate a product which he/she buys,in this case missing data occurs as whenever a new user visits a e-commerce website he is enable to decidewhich product is good. So Tkd queries helps in finding a solution to the given problem by making use of Tkd algorithms.Let us consider an example of e-commerce website.where there are few products and ratings are given by website users.The below Table shows the ratings from various users for various products.

**Table 1**
**Ratings from various users for various products**

| *Products* | $U_1$ | $U_2$ | $U_3$ | $U_4$ |
|---|---|---|---|---|
| $P_1$ | 5 | 4 | – | 4 |
| $P_2$ | 3 | – | 2 | 3 |
| $P_3$ | 2 | 3 | – | – |

As we observe from the above table that product p1 is better than or dominates other products *i.e*; for $P_1$ $P_1$. [1] = 5 > $P_2$.[1] = 3 and $P_1$.[1] = 5 > $P_3$.[1] = 2, $P_1$. [2] = 4 > $P_3$.[2] = 3, $P_1$.[4] = 4 > $P_2$.[4] = 3 and for product $P_2$ $P_2$. [1] = 3 > $P_3$.[1] = 2, $P_2$ dominates product $P_3$. Score is calculated based on the number of objects it dominates product $P_1$ score[$P_1$] = 2 as it dominates two products, product $P_2$ score [$P_2$] = 1 as it dominates one product $P_2$. A Tkd query, given K = 1 it returns product $P_1$as best product.

## 2. RELATED WORK

Khalefa, [3] proposed the work on skyline query processing on incomplete data, skyline queries aim to prune the search space of large number of multidimensional data items to small set of interesting items by eliminating items that are dominated by other items,Gao et al.[8] propose an efficient kISB algorithm for processing k-skyband queries over incomplete data. Lofi et al. present an approach to compute the skyline using crowd-enabled databases with the challenge of dealing with missing information in datasets.

Papadias [4] first proposed Tkd queries as another form of skylinequeries and presented a skyline based algorithm for processing Tkd queries on traditional complete dataset indexed by R Trees.

There are various forms of top *k* dominating queries subspace dominating query that handles subset of dimensions in progressive manner, continuous top –*k* dominating queries over data streams, metric based top *k* dominating queries that processes top k dominating over distance-based dynamic attribute vectors, defined over metric space, top *k* dominating queries over massive data.

X. Lian and L. Chen[2]proposed probabilistic top *k* dominating queries in uncertain databases a probabilistic top *k* dominating query returns *k* uncertain objects that are expected to be dynamically dominate the largest number of uncertain objects . Yiu and mamoulis[7] proposed aRtrees to tackle Top *k* dominating queries.

## 3. TOP-K DOMINATING QUERY PROCESSING ON INCOMPLETE DATA

### 3.1. Features

1. Top-K dominating queries does not hold the property of transitivity

   Ex:x(_,5,4), *y*(_,4,_), *z*(4,_,5), here *x* dominates y, y dominates *z*, but *x* does not dominate *z*.

2. At least one dimension must be common, then only we can apply dominance relationship among two comparing objects

   Ex: x(_,5,4), *y*(_,4,_), these are comparable, if *x*(_,,5), *y*(_,4,_) we cannot compare.

**Dominance Relationship :** Dominance relationship can be applied only on common dimensions of comparing objects.

Given two entities A and B, A dominates B if and only if A is as good as B in all dimensions, and it is strictly better than B in at least one dimension.

**Score:** Score for a Tkd query is defined as number of entities it dominated by a particular entity in an incomplete dataset.

| |
|---|
| P(5,4,_,5,4) |
| Q(3,2,3,_,1) |
| R(_,3,2,_,3) |
| S(4,_,3,_,2) |
| T(3,_,_,5,4) |

**Figure 1: Prototype Of Incomplete Dataset**

Figure 1 illustrates a incomplete dataset,where there are five entitieswith five dimensions. For each entity there is some missing dimension(s), consider first entity P third dimension is missing, for second entity Q fourth dimension value is missing,for R third entity first and fourth dimension are missing, for fourth entity S second and fourth dimensions are missing, for fifth entity T, second and third dimensions are missing. Each entity is compared with every entity present in the dataset and score is calculated based on number of entities it dominates. Suppose for entity P, P is compared with Q and compared with each every dimension, if any of the dimension is missing in either of the entities then can't be comparable. In this case P dominates Q, so now score is 1, next another entity R is compared , as it dominates score is incremented ,now score is 2, and so on with each and every entity present in dataset.

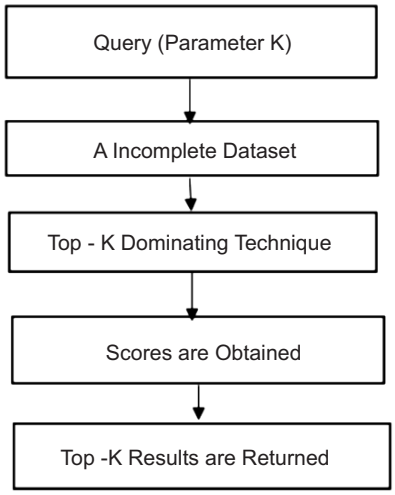


**Figure 2: Flow Of Execution**

The above figure illustrates the flow of system, first a query is imposed with parameter suppose parameter $k = 2$, to the incomplete dataset such as *a*(2,_,3,5) *b*(3.5,_2) *c*(4,_,_,5)…and so on datasets of variable size, top -*k* dominating technique is applied, scores are calculated entity having highest score is returned if $k = 2$ then two entities are returned as a final result.

To solve this problem [1], various algorithms are proposedinESB algorithm,pairwise comparisons are done for the whole dataset and scores are calculated it does not prune any entity present in the dataset so we introduce UBB algorithm which focuses on early windup of TKD Query processing on incomplete data before

evaluating all the candidates. *i,e*; it reduces the size of candidate set by making use of upper bound prune scoring i.e. maxscore is calculated for each entity, but any how upper bound score may be loose, we have to derive the actual scores by pairwise comparisons and sometimes even for the whole dataset. Thus it lower the search performance, to overcome this problem an efficient algorithm is needed, BIG algorithm which makes use of bitmap catalogue, where each entity is presented in form of bit string, where each dimension of entity is presented in the form of $U_i$ + 1 bits, where $U_i$ represents unique terms for that dimension and one bit indicates as a missing bit. For example if dataset has four entities with three dimensions such as *e*1(2,_,5), *e*2(3,_,_), *e*3(1,_,3), *e*4(2,5,_) then for the first dimension unique terms are 2,3, and 1 now $U_i$ value is three for first dimension then one more bit is added to it *i.e*. a missing bit to give the value of $U_i$ + 1, as four ,similarly for all the dimensions $U_i$ + 1 is known then bitmap catalogue is constructed for all the entities in dataset. BIG is slightly improved,binning strategy is used in IBIG which efficiently minimizes the storage drawback. there are two types of compression techniques,WAH The Word Aligned Hybrid (WAH) bitmap compression trades some space to allow for bitwise operations without first decompressing bitmaps. WAH has been recognized as the most efficient scheme in terms of computation time. and CONCISE [6], (Compressed '*n*' Compassable Integer Set), a new scheme that enjoys significantly better performances than those of WAH andwhile compared to WAH CONCISE compression ratio is better. Thus while compared to BIG IBIG consumes less storage.

## 4. CONCLUSION

This paper is a overview on TKD queries on incomplete data with some missing dimensionsthis is mostly useful for applications like decision making applications, search pruning etc.as it returns top best entities which results in better outcome ESB UBB these two algorithms utilizes novel technique to prune the search space. Furthermore to increase the efficiency of the algorithm BIG, which uses bitmap or bit array and IBIG which uses binning strategy, thus storage space is reduced, *i.e*., efficiency is increased.

## 5. ACKNOWGLEGMENT

I hereby would like to express my deep sense of privilege to my guide Prof.V.Chandra Shekar Rao ,who motivated and guided me towards completion of my work.I feel extremely grateful and indebted to my guide for being a source of inspiration, to see things positively and felt honoured with their confidence and trust on my ability. I profoundly express my sincere thanks to Dr.P.Niranjan,Head of the department for providing all the facilities and also to staff of department of computer science and engineering.I shall ever be thankful to my parents and friends.

## 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] X. Lian and L. Chen, "Probabilistic top-*k* dominating queries inuncertain databases," Inf. Sci., vol. 226, pp. 23–46, 2013

[3] M. E. Khalefa, M. F. Mokbel, and J. J. Levandoski, "Skyline query processing for incomplete data," in Proc. IEEE 24th Int. Conf. Data Eng., 2008, pp. 556–565.

[4] D. Papadias, Y. Tao, G. Fu, and B. Seeger, "Progressive skyline computation in database systems," ACM Trans. Database Syst., vol. 30, no. 1, pp. 41–82, 2005.

[5] L. Antova, C. Koch, and D. Olteanu, "From complete to incomplete information and back," in Proc.

[6] A. Colantonio and R. Di Pietro, "Concise: Compressed 'n' composable integer set," Inf. Process. Lett., vol. 110, no. 16, pp. 644–650,2010.

[7] M. L. Yiu and N. Mamoulis, "Efficient processing of top-*k* dominating queries on multi dimensional data," in Proc. 33rd Int. Conf. Very Large Data Bases, 2007, pp. 483–494.

[8] Y. Gao, X. Miao, H. Cui, G. Chen, and Q. Li, "Processing k-skyband,constrained skyline, and group-by skyline queries on incomplete data," Expert Syst. Appl., vol. 41, no. 10, pp. 4959–4974,2014.

[9] 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.

[10] T. Imieli_nski and W. Lipski Jr, "Incomplete information in relational databases," J. ACM, vol. 31, no. 4, pp. 761–791, 1984.