Dictionary Compression: An Optimal Answering for Continuous Top-k Dominating Queries

[1]Ramya.A, [2]Sankaran.L, [3]Kumaresan.A, [4]Manju Parkavi.R, [5]Kalaignanam.K, [5]Lokesh.V
[1,2,3,4,5]Department of Computer Science and Engineering
[1,2,3,4,5] SKP Engineering College, [1,2,3,4,5]Tiruvannamalai, [1,2,3,4,5]Tamil Nadu, [1,2,3,4,5]India

***ABSTRACT*- An online Query based searching involves mainly two methods of retrieving the data from the database. (1)Top-k query(2)Skyline query. The ranking rule methodology is used in Top-k query whereas dominance relationship is used in skyline query. Top-k dominating query used to search and retrieve the K records with highest dominance results either by ranking and dominance relationship method in static database. The Continuous Top-k Dominating query(cTKDQ) method has some limitations.In order to overcome the existing demerits, introduces a new indexing structure called close dominance graph(CDG) to support and maintain the relationship between dynamic data records. However, CDG as it takes more time to search results. In this paper introduce a dictionary based compression algorithm, which is efficient in answering a cTKDQ with minimum time and memory. The experimental results have shown that this scheme is able to better performance when compared with other prevailing scenarios.**

***INDEXTERMS* -Close dominance graph,Continuous query,Dictionary compression,Dominance relationship,Streaming model,Top-k dominating query**

I. INTRODUCTION

Effective processing of top-k queries is a crucial requirement in various interactive environments that involve excess amountof data. In particular, top-k processing is efficiently done in domains such as the Web, multimedia search, along with distributed systems have caused a greater impact on performance. Different types of information systems use various techniques for ranking query answers. Considering different application domains, users are considered with the important (top-*k*) query answers in the large answer space. Various recent applications supports a wider support for top-*k* queries. For instance, while considering the Web, the efficiency of Meta search engines, which combine rankings from various search engines are highly related to effective rank composition methods. One common way of identifying the top-*k* objects is scoring various objects with respect to some *scoring function*. An object score is the representation of differentiating that object from its characteristics. Data objects are evaluated by enormous scoring predicates which combines to evaluate the total object score. A scoring function is normally defined as an aggregation over partial scores. Top-*k* processing leads to various database research areas including query optimization, indexing methods, and query languages. As a consequence, the impact of effective top-*k* processing is becoming evident in a number of applications. One way to answer such a multi feature query is by scanning all database objects sequentially, computing the score of each object with respect to various features and combining the scores into a total score for each object. This approach has problems in scalability problems with respect to database size and the number of features. An alternative way is mappingthe query into a join query which joins the output of various single-feature queries, and then sorting the joined results with respect to the combined score. This approach does not scale with respect to both number of features and size of the database since all join results have to be computed followed by sorting. The main problem with sort-based approaches is that sorting is a blocking operation which requires full computation of the join results. Though the input to the join operation is sorted based on individual features, this order is not exploited by various join algorithms. Hence, sorting the join results becomes necessary for producing the top-*k* answers. Embedding rank-awareness in query processing techniques provides an efficient solution. In order to answer a scalable problem, we propose a Dictionary based Compression algorithm mainly for reducing the searching time and memory.

II. RELATED WORK

First introduced in the work of top-k query has been one of the most popular preference data retrieval schemes today [1]. Basically, top-k queries use a function to define a score for each record in the dataset. The records are then sorted according to these scores and the top k records are returned to the users. Top-k query has found application in many fields such as information retrieval [2]. A comprehensive survey about top-k query in database can be found in a data structure called Pareto-Based Dominant Graph (commonly known as DG) was proposed in consideration of the intrinsic connection between top-k problem with aggregate monotone functions and dominance relationship. As a result, DG can be used to reduce the search space in order to answer a top-k query. In a threshold-based technique was proposed for processing a top-k query in a highly distributed environment. The aim of the technique is to minimize the amount of data transferred between nodes[3]. The authors proposed the reverse top-k query which addresses the problem of deter- mining the weighting vectors in which a given record is on the top k list of a dataset [4]. Related research can also be found in skyline query provides users with a set of data that are prominent based on the concept of dominance relationship. Extensive research results have since been reported in the literature. In [5], two algorithms, called Bitmap and Index, were proposed to progressively provide skyline points. The authors proposed a nearest neighbor search-based technique to tackle the problem without having to read the entire dataset. Instead a of centralized system environment, the processing of a skyline query has also been adapted to peer-to-peer systems [6]. As an alternative proposed the concept of representative skyline that returns k skyline points that best describe tradeoffs among different dimensions offered by the full skyline [7]. Furthermore, [8] presented a technique to answer a skyline query by using the Z-order curve approach. TKDQ is a relatively new topic in the field of data management. Several algorithms that are based on the aggregate R-tree were proposed to process the query. In this scheme, a pruning technique was used to reduce the search space needed to obtain the query result [9]. In the problem of probabilistic top-k dominating query was introduced to address uncertain datasets. The introduction of streaming models presents new challenges for the processing of data queries, in which updated query results must be returned continuously [10]. The authors provided a review of some important preference queries over data streams [11]. The proposed method returns k records with highest domination scores over dynamic dataset with respect to any subset of available attributes. Assisted by the grid structure, the computation of a domination score is simplified because intensive checking is needed only for records existing in partially dominated cells [12]. Later, the authors proposed an event-based approach to deal with the problem in [13].The CDG answering for cTKDQ to reduce search space in unique indexing structure[14]. The CDG as it takes more time to search results although an enhanced algorithm with additional optimization techniques were proposed in this paper, reducing in time effort and memory remains a challenge.

III. PROCESSING FOR cTKDQ

The dominance relations gathered from a set of records is maintained by the methodology proposed by the authors based on the graph data structure. During the process of updating the answer to the query, we can greatly minimize the search space by making use of the unique characteristics of the data structure.

A. Close Dominance Graph

Definition: Consider a set of records R in a multidimensional space. The close dominance graph of R is a directed graph where R represents the vertex set and a direct link connects two vertices say s and r if and only if s→ r exists.

B.The Query Result

The authors have assumed a count based sliding window where inserting a new record leads to the removal of the remaining record which happens to be the oldest. The authors have obtained the query result as a result of updation process which is done with the help of CDG graph.

C.Updating Top-K Based On Anchor Set

With the help of CDG graph, the authors were able to obtain a set of records which could be termed as the candidate records. They were able to attain a subset of records from the candidate records which were named as the Anchor set. This anchor set does not dominate the records available in the Candidate set.

D.CDG Construction

CDG has to be constructed from scratch. Here, records would be deleted only when the sliding window gets filled up. The authors have focused on inserting and deleting records simultaneously. Two approaches were there for constructing CDG which are static and incremental. The static methodology is based on the fact that the query result would be formulated to a cTDKQ only after the sliding window gets completely filled. The static methodology would work based on the assumption that the collection of new records would be done at the starting operation of the system until the sliding window gets filled. The detailed algorithm used by the authors for the initialization of CDG is explained in [14].

E.The Anchor Based Algorithm

The authors have used a main algorithm called as the Anchor based algorithm (ABA)which was used in the consistent update process of the database.ABA not only look after the process of updating the query result but also takes care of the book keeping process. The ABA could be divided into three parts which are to be carried after one after the other. The first process involves updating the CDG which is followed by the generation of anchor set and the final process involves the computation of updated query result.

IV AN ANSWERING FOR CTKDQ

Dictionary based compression algorithm is mainly chosen for generating hash table for repeated data entries instead of creating a new record in dynamic database. The searching time delay for client to server will be greatly reduced due to the hash linking rather than searching entire database. The storing area and searching time will be optimized using dictionary compression. Dictionary-based compression algorithms normally creates a dictionary in the form of data which is scanned in order to find repeated information. With respect to the pattern recognition (a look-up in the dictionary), that string of information is replaced by a much shorter but uniquely identifiable string. This could result in a compression of that overall data. The size of the dictionary along with the speed at which the scan is done are implementation decisions from the hash table.

A. Dataset Acquisition

Visualizing a multidimensional dataset requires scatter and mesh tool sets and will be more complex with increased dimension. Image is one of the easily understandable multidimensional data with these advantage a dataset consists of thousand images input with various dimensional sets has been acquired.

B.Dynamic Database

Acquired input images are trained to a dynamic database with parallel addition and deletion data fields. Mat lab tool is used to create a access database, which has its unique data storing pattern as mat file.

C.Client network connection with timing token access

Multiport client handling is managed by creating priority threads in server to receive/send datas from/to client ports parallely. Congestion of network will be avoided by assigning timing tokens to client ports.

D.Parallel server port handling for client request

Single socket half duplex network connection is initialized with user interactable menus for Appending, Reading and deleting central server database. Each client users are allocated with a time frame to send Top-k queries to server, once the time-out occurs,client connection will cannot be received until the next time token from server. When all the client connected to the server is terminated, the server ports will remain in idle state and ‘waiting for new connection acquisition.

E.Close Dominance Graph Index Creation

Close dominance graph generation is specifically designed with streaming database as reference with CDG theorems its proven that based on dominance and data ranking priority, parallel indexing is possible in multidimensional datasets .Quick navigation or querying isacheived by indexing datasets by calculating that matching dominance and creating a ranking table based on the dominance.CDG is the reference graph for client quering and parallel updation.

F.Anchor Query Algorithm

Best dominance result r∈ A(r) can be achieved when r∈ r' with r' as top-k candidate. Anchor records are categorized into the dependent records of top-k candidate records which doesn't have dominance to candidate records. Anchor based querying ensure the adaption to CDG's dynamic updating and creating key candidates with single alteration process.

G.Dictionary compression Algorithm

Dictionary based compression are widely employed in the field of data mining. This project extends the idea into image mining application. The redundant data entries occurrence will be high in a streaming database. Dictionary Compression acts as a dynamic filter to avoid the reoccurrence of similar data pattern. Dictionary algorithm matches the input data pattern into in dictionary memory, if matching occurs it replaces the input data pattern designation with the dataset index_id of already existed data pattern. pattern. In which case memory and querying time will be reduced greatly

V EXPERIMENTAL RESULTS

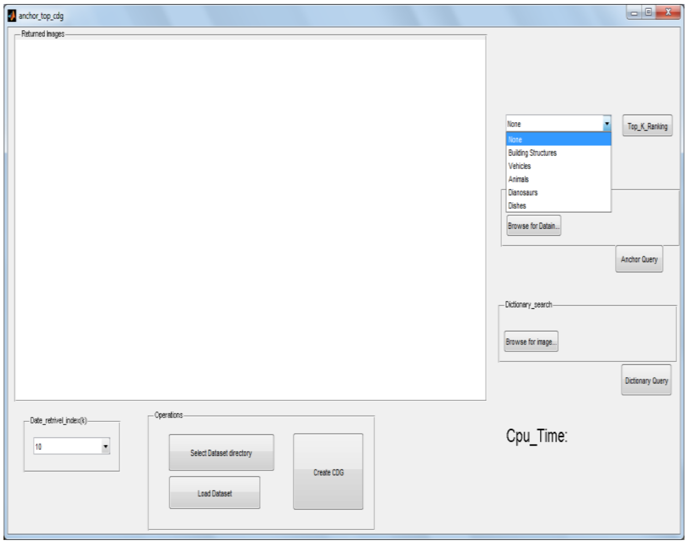


Fig.1. List of available datasets

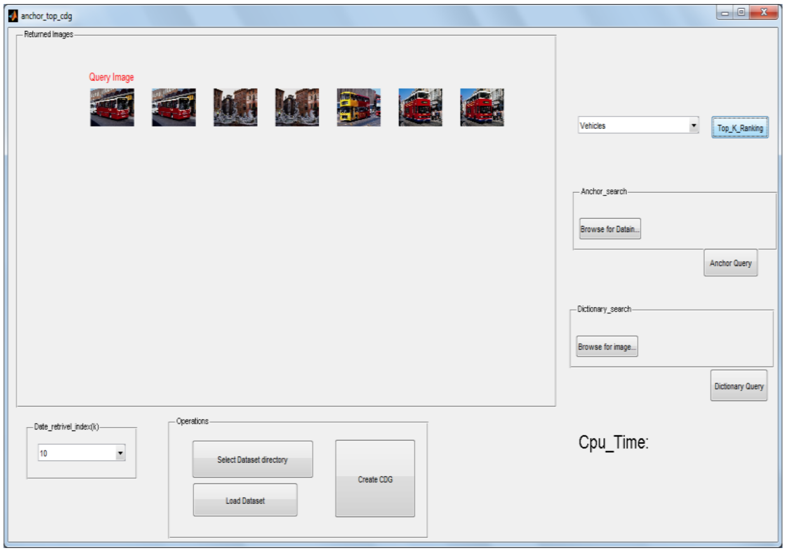


Fig.2.Static vehicle dataset is displayed

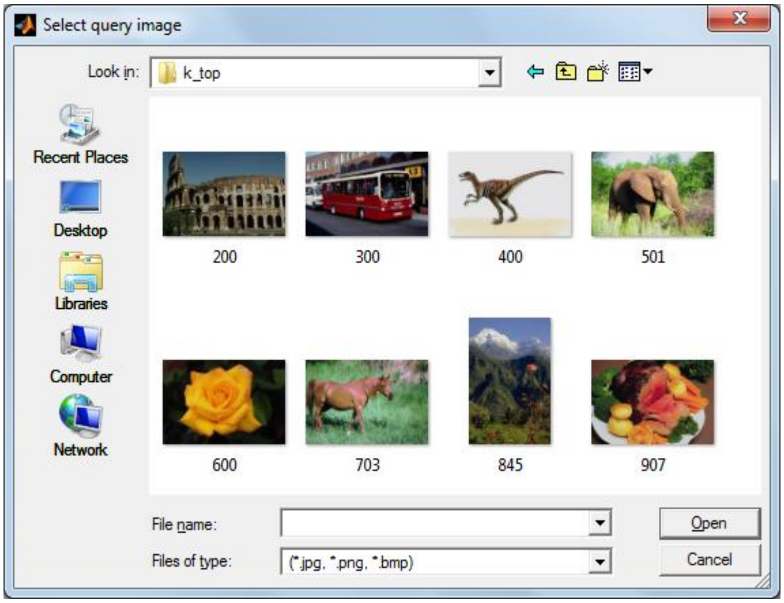
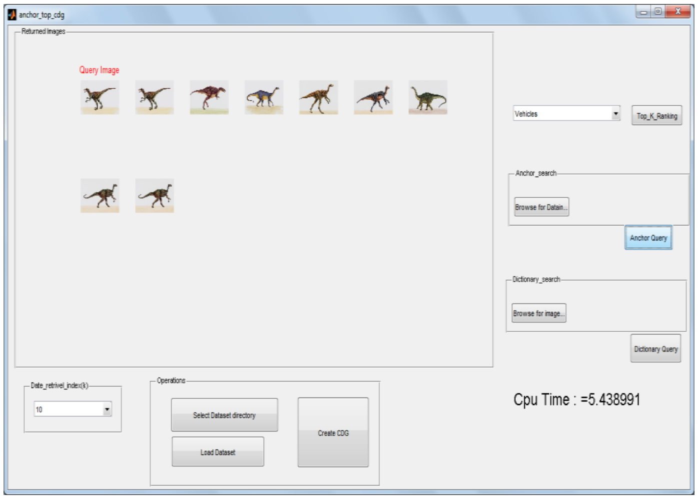
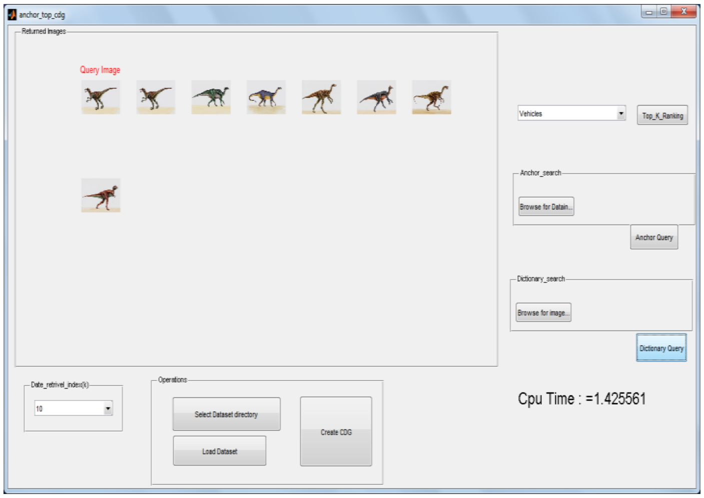


Fig.3.Choosing the query image from the system



Cpu Time : =5.438991

Fig.4Images related to the query image are retrieved from the dataset successfully with a higher CPU time



Cpu Time : =1.425561

Fig.5 Result of dictionary compression where redundant images are filtered and cpu time is reduced

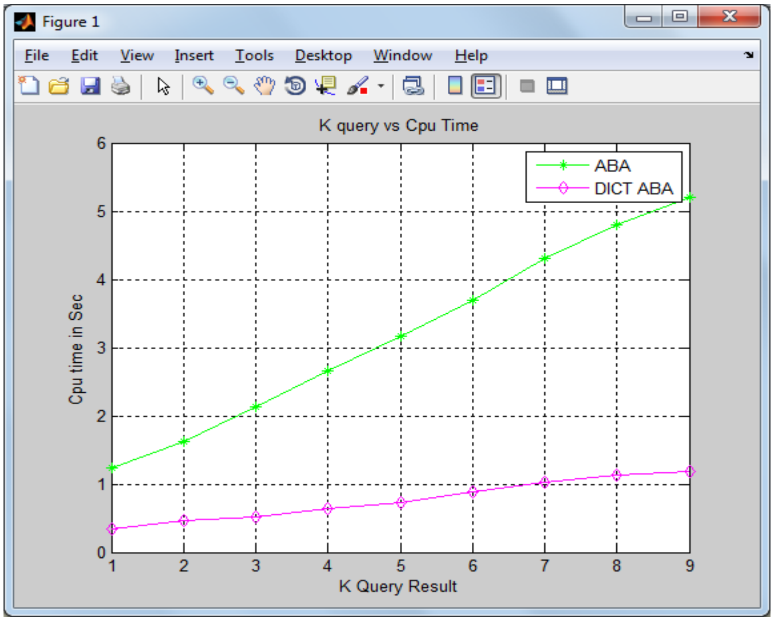


Fig.6 Graph plotted with results from different retrieval index

VI CONCLUSION

This paper introduces a Dictionary based Compression algorithm, which is efficient in answering a cTKDQ with minimum time and memory . The method of ranking rule is used in Top-k query and dominance relationship is used in skyline query. Top-k dominating query is used to search and retrieve the K records with highest dominance results by ranking and dominance relationship method in static database. cTKDQ has some limitations .In proposed dictionary based compression algorithm is able to overcome the existing demerits in CDG where Close Dominance Graph (CDG) serves as a framework for supporting the processing of a cTKDQ.

REFERENCES.

[1] R. Fagin, "Combining fuzzy information from multiple systems (extended abstract)," in *Proc. ACM SIGACT-SIGMOD-SIGARTSymp. PODS*, 1996, pp. 216Ð226.

[2] T. Tran, H. Wang, S. Rudolph, and P. Cimiano, "Top-k exploration of query candidates for efficient keyword search on graph-shaped (RDF) data" in *Proc. ICDE*, Shanghai, China, 2009, pp. 405Ð416.

[3] Vlachou, C. Doulkeridis, K. N¿rvŒg, and M.

Vazirgiannis, “On efficient Top-k query processing in highly distributed environments,” in *Proc. ACM SIGMOD*, 2008, pp. 753Ð764.

[4] Vlachou, C. Doulkeridis, Y. Kotidis, and K. Norvag, “Monochromatic and bichromatic reverse Top-k queries,” *IEEETrans. Knowl. Data Eng.*, vol. 23, no. 8, pp. 1215Ð1229, Aug. 2011.

[5] K.-L. Tan, P.-K. Eng, and B. C. Ooi, “Efficient progressive skyline computation,” in *Proc. 27th Int. Conf. VLDB*, San Francisco, CA, USA, 2001, pp. 301Ð310.

[6] D. Papadias, Y. Tao, G. Fu, and B. Seeger, “An optimal and progressive algorithm for skyline queries,” in *Proc. ACM SIGMOD*, San Diego, CA, USA, 2003, pp. 467Ð478.

[7] Y. Tao, L. Ding, X. Lin, and J. Pei, “Distance-based representative skyline,” in *Proc. ICDE*, Washington, DC, USA, 2009, pp. 892Ð903.

[8] K. C. Lee, W.-C. Lee, B. Zheng, H. Li, and Y. Tian, “Z-SKY: An efficient skyline query processing framework based on Z-order,” *VLDB J.*, vol. 19, no. 3, pp. 333Ð362, Jun. 2010.

[9] M. L. Yiu and N. Mamoulis, “Multi-dimensional Top-k dominating queries,” *VLDB J.*, vol. 18, no. 3, pp. 695Ð718, Jun. 2009.

[10] X. Lian and L. Chen, “Top-k dominating queries in uncertain databases,” in *Proc. 12th Int. Conf. EDBT*, 2009, pp. 660Ð671.

[11] M. Kontaki, A. N. Papadopoulos, and Y. Manolopoulos, “Continuous processing of preference queries in data streams,” in *Proc. 36th SOFSEM*, ŁpindleruvMlýn, Czech Republic, Jan. 2010, pp. 47Ð60.

[12] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, “Skyline with presorting,” in *Proc. 19th ICDE*, Bangalore, India, Mar. 2003, pp. 717Ð719.

[13] M. Kontaki, A. Papadopoulos, and Y. Manolopoulos, “Continuous Top-k dominating queries,” *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 840Ð853, May 2012.

[14] BagusJatiSantoso and Ge-Ming Chiu, ”Close Dominance Graph: An Efficient Framework Answering Continuous Top-*k* Dominating Queries”, Member”, *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1853,Aug 2014.