STUDY ON TOP K QUERIES AND DURABLE QUERIES OVER TEMPORAL TIME SERIES

B.Saranya,
M.Phil Research Scholar,
Department of Computer Science,
Siri PSG College of Arts and Science for Women,
Sankari, Tamilnadu, India.

K.Sumathi,
Assistant Professor,
Department of Computer Science,
Siri PSG College of Arts and Science for Women,
Sankari, Tamilnadu, India.

Abstract: Durable query scheme is used to find objects in historical time series databases. Durable queries can be extended for multidimensional time series analysis. Aggregate scoring function is used to rank the series at every time instance in top-k search. Euclidean distance measure is used in KNN queries at each timestamp between the reference series. Time series analysis plays an important role in predicting the status of the query, at every time stamp to retrieve efficiently. Studies have shown that different approaches are used for different queries over time series. The paper tries to survey some improved methods and have experimentally tested their effectiveness. Besides, it also studies some future directions on historical time series queries.

Keywords: Top K Queries, Temporal Time Serious, Durable Query

I.INTRODUCTION

Temporal data refers the sequence of events with time stamp details. The goal of temporal data mining is to discover hidden relations between sequences and sub-sequences of events. In temporal mining data clustering, data classification and relationship finding are carried out with temporal dependencies. Time series data analysis is used in engineering, science and financial domains. Time series (ex: sensor readings, daily closing stock prices, etc.) is an application of temporal databases. Attributes of those objects are temporal attributes [1]. Various efforts have been taken for indexing and querying temporal data, mostly on similarity search queries, aggregate queries, nearest neighbors, range queries, temporal pattern queries, top-k queries and interval skyline queries [2]. Top-k query is a one dimensional nearest neighbor query where query point is infinite. KNN(K-nearest Neighbor) is used to answer Top-k queries both in small and large databases [3]. In top-k query processing, Euclidean distance measure is used for indexing KNN. However more novel approaches exist [4]. The contents of the time series are pre-defined. Therefore the values on which the topk function is applied are pre-computed and can be indexed. Meagre amount of work has been carried out on top-k processing and no previous work has been done on KNN queries. As a result, the query performance is far from satisfactory. KNN queries use R-trees(minimum bounding rectangle tree) to support Topk, but query performance is not guaranteed. Hence the special case of MVB(Multi version B trees)trees and SEB(Sample Envelope B-trees) trees are applied which employ index solution. MVB trees provide moderate query performance compared to SEB trees due to its poor scalability of size and construction cost for general piecewise linear functions. Due to this, MVB trees do not support general updates, whereas SEB trees support updates of historic data. SEB trees are also a collection of B-trees which have a simple query algorithm and it can be easily integrated with DBMS. This is limited in both R trees and MVB trees. But if the temporal attribute of object is piece wise constant (i.e., staircase), then the only solution is MVB trees, which supports queries on any version of the B trees as efficiently as if each version is stored individually. MVB trees keep all versions of the B trees.

II.RELATED WORKS

Numerous solutions [5] have been proposed for indexing time series to support similarity search. Such queries retrieve time series that are closest to a reference series, according to a certain distance measure. Two popular distance measures are 1) the euclidean distance in the space defined by considering each time instance as a dimension and 2) dynamic time warping (DTW) which improves robustness over the Euclidean distance by allowing mapping of shifted sequence elements. The Gemini framework addresses the dimensionality curse in time-series indexing and search using dimensionality reduction; popular methods in this direction

include Chebyshev polynomials, piecewise linear approximation [6], APCA and so on. These methods do not apply to durable queries, as they focus on an object's overall similarity to a query, rather than their properties at individual timestamps. There is also a vast amount of existing work in indexing and searching trajectories, where each record contains the locations of a moving object at different timestamps.

Guting et al. [7] study a similar problem, termed TCkNN, but focus on retrieving the nearest neighbors during a historical period rather than the current ones. Specifically, a TCkNN query finds, at each timestamp during the given period, the NN of a reference trajectory. The technical focus is to organize trajectories into an R-tree-like structure and then take advantage of some pruning heuristics. Compared to similarity queries, there is little work on top k queries for time series data, despite the importance of such queries. Although it is possible to simulate a top-k query by a k-NN query with an imaginary reference time series that has the largest domain value at each time moment, such a reduction is often "far from satisfactory"; methods designed for similarity search do not capture well the unique properties of top-k search.

Li et al. [8] conducted a thorough study on the evaluation of snapshot top-k queries on continuous time series with a piecewise linear representation. The focus is clearly different from ours, both in terms of the query nature and the data model used.

Jestes et al. [9] study aggregate top-k queries on temporal data with a piecewise linear representation. The goal is to find the top-k objects with the highest aggregation scores in a given time interval. The focus and the data model are clearly different from ours, and their solutions do not apply to durable queries.

Lee et al. [10] were the first to study the consistent top-k query, which is the special case of DTop-k with the durability threshold r fixed to 100 percent. In the example of consistent top-2 query with time period (0, 5) retrieves only object s2. The basic idea of the solution is to exhaustively verify every object in the data set against the query definition. For each object s, LHL first checks whether s belongs to the top-k set at timestamp t *b* . If so, LHL continues to check if s is a top-k object at t *b*1 ; otherwise, it discards s and starts with another object. The process continues, until either s is eliminated, or after checking the rank of s at every timestamp in the query window. To accelerate snapshot top-k membership checking, LHL pre computes the rank of each object at every timestamp, and organizes this information into a sorted list, stored on disk in a compressed format. For instance, LHL associates the list (1, 2, 2, 3, 3) to object s1, signifying that s1 ranks first at timestamp 0, second at time 1-2, and third at time 3-4. During query processing, LHL scans the rank list of an object linearly from t *b* , until reaching either te or a value larger than k. LHL does not support DTop-k queries with r < 100%. To handle such cases, we extend LHL as follows: For each object s, we scan the part of its rank list from timestamp tb to te, and count the number of times that s is in the snapshot top-k sets. During the scan, if we find that s is outside the top-k set for more than (1-r). (t *e* t *b*) timestamps, we drop s since it cannot possibly reach the durability threshold r. The set of objects that pass the verification are reported as results. The main drawback of LHL is that it scales poorly with the number of objects, as each object initiates a list scan with at least one I/O read.

U et al. [10] studied durable top-k queries in the context of keyword search in web archives, where each object is a web document that gets edited or replaced over time. In addition to the parameters k, (tb, te), and r, a durable query also involves a keyword list K *W* . The score of a document version is calculated based on its relevance to the keywords in K *W* with an IR model. There are important differences between our work. First, computing the relevance of a document to an arbitrary K *W* is both hard and expensive. Therefore, preprocessing methods cannot be used to accelerate search in our work. Second, the data domain, i.e., versional documents, is quite special: the relevance of keywords to documents remains relatively constant in adjacent timestamps. When this assumption does not hold, for example, if all objects change values at every timestamp methods reduce to brute-force search. Hence, the solutions are tailored to a specific domain, and are not suitable for DTop-k queries in the general case.

Another piece of related work is the interval skyline query [3]. An object si dominates another sj, if an only if si is better than sj in at least one timestamp, and no worse in all other timestamps. The set of objects that are not dominated is then reported as the interval skyline. The interval skyline and the durable top-k, however, retrieve very different results. The former's result set includes objects with high values in a small number of timestamps, whereas the latter identifies objects with durable quality. For instance, s 1 is on the interval skyline as long as the query window contains timestamp 0 (where s1 is the best object), regardless of its scores in other time instances. Consequently, the solutions are inapplicable to our problems. The probabilistic top-k query finds objects with high probability to be in the top-k set, is also remotely related to this work, since one can view each timestamp as a possible world, and calculate the probability for each object. On the

other hand, the focus is clearly different from ours, and their methods do not apply to durable queries.

Finally, recent-biased time series have been studied in the context of online analysis of streaming data. The focus of this work is different, however, since we focus on offline queries over historical data. For instance, in the various application scenarios, it is generally more natural to consider timestamps within the query window as equally important, than giving higher weights to more recent time instances.

III. CONSISTENT TOP-K QUERIES

The classes of queries that retrieve objects which exhibit consistent performance over time are called consistent queries [10]. It holds importance in many applications (weather forecasting, stock exchange, traffic management and e-cops management) to maintain historical records of data, which the users need in time with persistent behavior. To focus on consistent queries, previous works have studied to have a query with durability threshold fixed to 100%. The contents of the time series are also predefined. Therefore the values on which the topk function is applied, is pre-computed and indexed. However pre-processing methods are not used to accelerate the search. Each time stamp is equally important and it employs equal weight time series model. Consistent topk queries are different from topk queries and skyline queries. In both these methods one may not able to retrieve the desired objects in a multidimensional data set whereas using consistent topk queries one can retrieve desired set of objects which is consistent over time. Two methods are used to evaluate consistent topk queries. Rank list method captures the rank information of the time series data. Bitmap approach is used for leveraging bitwise operations to answer consistent topk queries. Among these two methods, bitmap approach is more efficient and scalable than rank list method.

IV. DURABLE TOP-K QUERIES

Queries that retrieve objects with durable quality over time in historical time series databases are referred to as Durable Topk Queries. Durable top k queries are an extension of snapshot topk queries and Nearest Neighbor queries. These queries employ Euclidean measure for indexing nearest neighbors so as to tackle the problem of triangular inequality as in the case of dynamic time warping. Most of the studies prefer normal interval tree which is constructed to map rank change intervals. However, this fails in case of IO cost for long query windows. The paper finds that usage of CJI(Conceptual join Intervals) trees is a better solution as it reduces unnecessary KNN intervals and provides incremental updating of time stamp results at every time series. In CJI, set of nodes are fixed throughout the topk evaluation [11]. This means that, B+ trees of the files is first used correspondingly to find the smallest time point in them and then they are scanned linearly and concurrently from these points. In all the existing methods and state of the art approaches, studies finds that the updating of rank intervals is very slow, since KNN sets may not change at every time intervals and also some retrieval may give best result for object indexing but not query indexing. In all the methods for Dtopk processing, results obtained for 2D is smaller than 1D series and it is an open problem. Many works have been finding it difficult to answer whether the k value is equal or varies for both KNN and top K processing [12]. Methods coming under GEMINI framework address the dimensionality curse in indexing and searching using dimensionality reduction, but they focus on object's overall similarity to a query, rather than individual timestamps. The methods existing for durable Topk processing, like TES framework exploits time series smoothness to reduce query cost. Also, QSI(Query Space Indexing) indexes the query space and avoids unnecessary snapshot KNN queries compared to value domain partitioning and brute force techniques [1].

V. PROBLEM DESCRIPTION

a) Existing System

The top-k query selects k best objects based on their ranking scores, is a common approach to obtaining a small set of desirable objects from a large database. Recently, top-k search has been extended to databases that contain multiple versions of data objects, for example, web archives, trajectory data, time series and so on. Ranked retrieval in such applications may need to consider not only an object's value at one particular time instance, but also its overall quality during a time period.

In the system in depth the problem of finding objects of consistent quality during a time interval. The system first analyzes the durable top-k (DTop-k) query that operates on a historical database where each object is a 1D time series, i.e., at each time instance, every series carries a single scalar. Given k, time interval $[t_b, t_e]$ and percentage $0 < r \leq 1$ (called the durability threshold), a DTop-k query retrieves objects that appear in the snapshot top-k sets for at least $[r \cdot (t_e = t_b)]$ timestamps during $[t_b, t_e]$. Fig. 3.1a shows an example with four series s_1- s_4. Assuming higher scores are preferred, a durable top-2 query with $[t_b, t_e) = [0, 4)$, r = 70% retrieves s_1 and s_2, since they appear in the top-2 set in at least 70 percent timestamps during (0, 4).

The system also identifies a natural extension of the DTop-k query: the durable k nearest neighbor (DkNN) query, which considers at each time moment the k nearest neighbors of a reference series s_{ref}. Consider again the example of Fig. 3.1a and the DkNN query with $s_{ref} = s_4$, k = 1, $[t_b, t_e) = [0, 4)$, r = 70%; i.e., related to the sequences that are the nearest neighbor of s_4 on at least 70 percent of the timestamps 0-3. The only series that qualifies this query is s3, since it is the NN of s_4 75 percent of the time in (0, 4). The DkNN query is much more challenging compared to DTop-k, since the former is rather resistant to materialization and indexing.

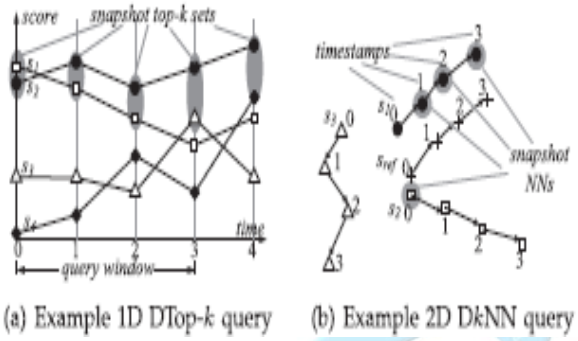


Fig. 3.1. Example Durable Queries.

Durable queries are useful in many real-world applications. For example, consider Google Zeitgeist, which presents weekly statistics of search keywords, each of which is associated with a time series of its search volumes. A DTop-k (resp. DkNN) query can be used to identify keywords that are frequently searched (resp. most related) during some time period, which may be further used by sociologists to understand the impact of certain historical events. A similar application is Twitter Trendsmap, which tracks frequently mentioned phrases and hashtags. In SciScope, a geospatial search engine built upon a wide-area sensor network, durable queries may be used by meteorologists to identify regions with consistently high environmental indices in particular time windows. In general, durable queries may serve as fundamental tools in time series analysis; domain experts can use their results to better understand their data and trigger further investigation.

Durable queries can naturally be extended for multidimensional time series, where at each time moment every series carries an array of values. For top-k search, to rank the series at every time instance, An aggregate scoring function is defined on the values in the individual dimensions. For NN queries, the system use a distance measure at each timestamp between the reference series s_{ref} and the sequences; for example, a police officer may investigate on vehicles consistently moving close to a pivot a suspect or a witness. Fig. 3.1b illustrates an example 2D time series data set containing the positions of three moving objects s_1-s_3 at timestamps 0-3. Considering a DkNN query with k = 1 and a period (0, 4), object s1 satisfies the query for r $\leq$ 75%, since it is the snapshot NN of s_{ref} for timestamps 1-3.

The only existing solutions for DTop-k employ either brute-force search, or techniques that are limited to specific domains. To fill this gap, the system uses an efficient method called top-k event scanning (TES). TES exploits the fact that real-world time series typically exhibit a certain degree of smoothness, meaning that the changes in the top-k set at adjacent timestamps are usually small, if at all. TES indexes these changes and incrementally computes the snapshot top-k sets at each timestamp of the query window. To efficiently support DkNN queries on 1D time series, query space indexing (QSI), method indexes the query space. Going one step further, QSI is extended to handle multidimensional top-k and k-NN queries. Extensive experiments using real and synthetic data confirm that the proposed methods significantly outperform previous ones, often by large margins.

B) Drawbacks of The Existing System

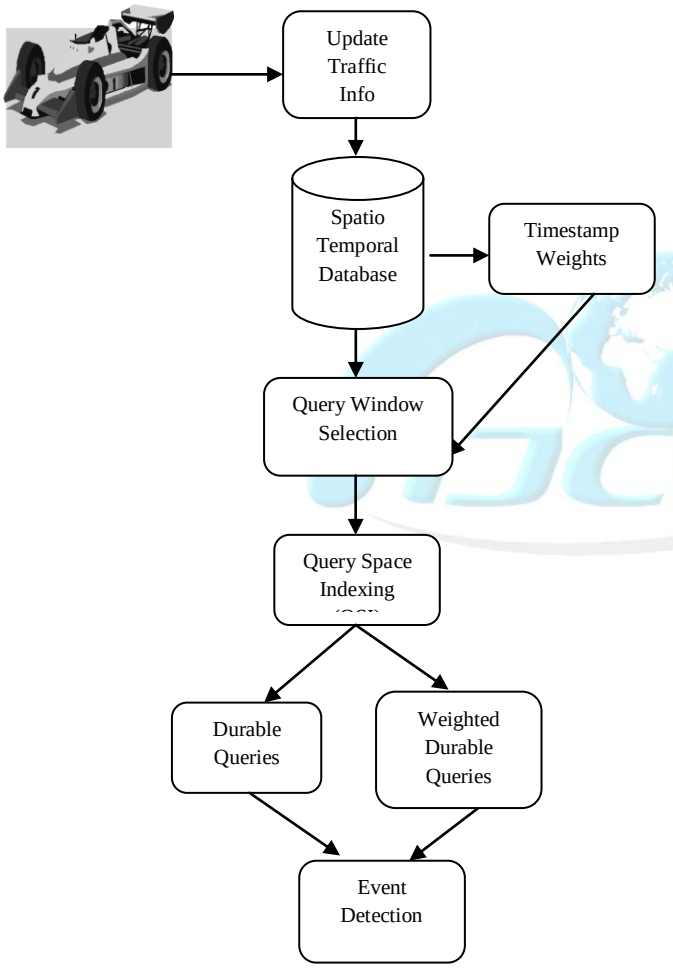
Durable query scheme is used to find objects in historical time series databases. Durable queries can be extended for multidimensional time series analysis. Aggregate scoring function is used to rank the series at every time instance in top-k search. Euclidean distance measure is used in KNN queries at each timestamp between the reference series. The durable top-k (DTop-k) query operates on a historical database of 1D time series. Query window and durability threshold are used to retrieve objects in Dtop-K query model. Durable k nearest neighbor (DkNN) query considers at each time moment the k nearest neighbors of a reference series. Top-k event scanning (TES) is used to detect actions that are related to the time limit. TES indexes the timestamp and incrementally computes the snapshot top-k sets at each timestamp of the query window. TES scheme is extended with Query Space Indexing (QSI) model to support DKNN queries on 1D time series. Query space indexing (QSI) indexes the query space. QSI is extended to handle multidimensional top-k and k-NN queries. The following drawbacks are identified from the existing system.

- Timestamp weights are not considered
- Pruning strategies are not used for timestamp analysis
- Time weight analysis is not carried out in DKNN query
- Noise elimination is not performed

C) Proposed System

The time series data analysis scheme is improved with noise elimination tasks. The durable query scheme is enhanced with weight based timestamp analysis process. The system is enhanced with pruning strategies to fetch data in required levels. Durable KNN query model is extended for time weight analysis mechanism.

System Architecture Diagram



VI.CONCLUSION

Durable queries are used for historical time series analysis. Durable top-k (DTop-k) and nearest neighbor (DkNN) queries are used analyze time stamped sequences of values or locations. The durable query scheme is enhanced with noise elimination and weight based timestamp analysis mechanism. The system supports high scalability in spatio temporal analysis. Query retrieval accuracy is improved in the system. One dimensional and multi dimensional data query are supported by the system. Weight based timestamp analysis mechanism is used to improve the event detection accuracy.In Future Enhancement will be spatio temporal query system can be enhanced to analyze the descriptive data values such as share market reviews and sport data reviews.The time stamp weight scheme can be integrated with spatial distance based weight scheme.

REFERENCES

[1]. J. Jestes, J.M. Phillips, F. Li, and M. Tang, "Ranking Large Temporal Data," Proc. VLDB Endowment, vol. 5, pp. 1412-1423, 2012.

[2]. F. Li, K. Yi, and W. Le, "Top-k Queries on Temporal Data," VLDB J., vol. 19, pp. 715-733, 2010.

[3]. R.H. Guting, T. Behr, and J. Xu, "Efficient K-Nearest Neighbor Search on Moving Object Trajectories," VLDB J., vol. 19, pp. 687- 714, 2010.

[4]. C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast Subsequence Matching in Time-Series Databases," Proc. ACM SIGMOD Int'l Conf. Management of Data, 1994.

[5] V. Athitsos, P. Papapetrou, M. Potamias, G. Kollios, and D. Gunopulos, "Approximate Embedding-Based Subsequence Matching of Time Series," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2008.

[6] Q. Chen, L. Chen, X. Lian, Y. Liu and J.X. Yu, "Indexable PLA for Efficient Similarity Search," Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB), 2007.

[7] R.H. Gu¨ ting, T. Behr, and J. Xu, "Efficient K Nearest Neighbor Search on Moving Object Trajectories," VLDB J., vol. 19, pp. 687-714, 2010.

[8] F. Li, K. Yi, and W. Le, "Top-k Queries on Temporal Data," VLDB J., vol. 19, pp. 715-733, 2010.

[9] J. Jestes, J.M. Phillips, F. Li, and M. Tang, "Ranking Large Temporal Data," Proc. VLDB Endowment, vol. 5, pp. 1412-1423, 2012.

[10] M.L. Lee, W. Hsu, L. Li, and W.H. Tok, "Consistent Top-K Queries over Time," Proc. 14th Int'l Conf. Database Systems for Advanced Applications (DASFAA), 2009.

[11]. L.H. U, N. Mamoulis, K. Berberich, and S. Bedathur, "Durable Top-K Search in Document Archives," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2010.

[12]. X. Yu, K.Q. Pu, and N. Koudas, "Monitoring K-Nearest Neighbor Queries over Moving Objects," Proc. Int'l Conf. Data Eng. (ICDE), 2005.