Survey on Comparative Analysis of Queries over Historical Time Series

Suganya Devi R
Research Scholar
Dept of Computer Science and Engineering
Anna University Chennai

D Manjula, Ph.D.
Professor
Dept of Computer Science and Engineering
Anna University Chennai

ABSTRACT

Most search engine queries are time dependent in retrieving the results. 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. Generally they are broadly classified into two types of queries; exact match queries and pattern existence queries. Some applications need the existence of any one of the queries and while others may need both. Numerous methods have been employed to answer both the queries. Analyzing all these methods, 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

Search engine queries, survey, comparative analysis, and historical time series.

1. INTRODUCTION

The need for concentration on queries arises from the fact that numerous number of users search for information through many search engines like google, msn, yahoo, etc. everyday. All queries given by the users are not the same and vary from person to person and also from time to time and this poses as a difficulty for the researcher to concentrate on a particular type of queries. As for the researchers, this work gives an overview of methods and approaches employed in queries over time series. Users may be initially interested in retrieving all the time series data that have some specific patterns which can be quickly answered by pattern existence queries. Following this, they may want to retrieve all the time series which are similar to an interesting time series that they may find from the previous retrieval results. In this case, they can use exact match queries. In any search engine, a query is given through the user interface through which the user can retrieve information according to it. In this paper, a survey on each form of queries is made and the nature of type-queries is processed. It has made a comparative analysis of forms of random queries over time series data. Many forms of queries use various methodologies and approaches according to the distributions and calculations they perceive.

2. TYPES OF QUERIES

2.1 Top-k Queries on Temporal Data

Temporal data refers to objects that change over time. Time series (ex: sensor readings, daily closing stock prices, etc.) is an application of temporal databases. Attributes of those objects are temporal attributes [8]. 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 [12]. 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 [5]. 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.

2.2 Snapshot Queries

Snapshot queries refers to finding the top k objects at a given timestamp, on continuous time series with piecewise linear representation. R-trees work well for snapshot query indexing, in which Topk queries at every timestamp are considered as snapshots. The only existing but efficient technique for piecewise linear function is R-trees [6], which is a multi-dimensional spatial index. For KNN processing, to answer a query at any given time instance t, R-tree uses branch and bound technique [5]. The list of K potential nearest neighbors in a priority queue is maintained using this method, and employs top down approach which terminates if any object is identified to be larger than any other object. Numerous algorithms have been designed to optimize the R trees but the query performance is far from satisfactory due to its high cost.

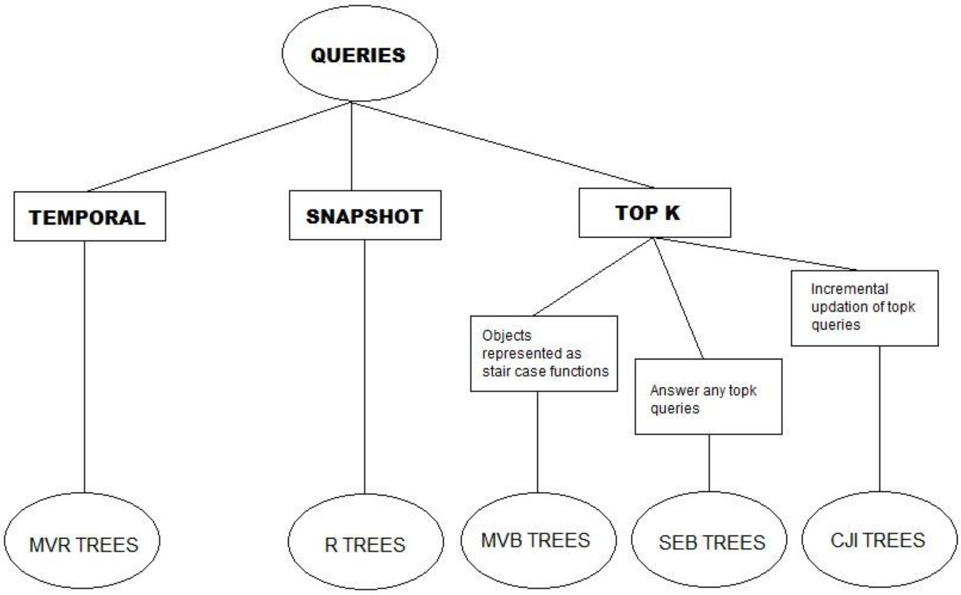

Fig 1: Tree Traversal Techniques in Query Processing

2.3 K-Skyline Queries

There exists time series which are not dominated by any other time series. The interval skyline queries return the above mentioned time series in a time interval [9]. There are two methods for computing interval skyline queries; On the fly method and View Materialization (VM) method. On the fly method keep maximum and minimum values of time series using radix priority search trees. It then sketches and computes skyline queries at query time.VM method maintains skyline over all intervals in a compact data structure. Both methods use linear space and are efficient for incremental maintainence.VM methods store only non-redundant interval skyline queries. Intervals of hundreds of timestamps (high dimension) are considered and equal weight is given to all time stamps. Radix priority search tree is a two dimensional data structure that allows efficient range queries, where the ranges on at least one dimension are un-bounded. It is a hybrid of a heap on one dimension and a binary search tree on another dimension. Optimal performance in query answering and update is obtained using radix priority search tree. The tree remains balanced after the updates. Dimensionality is extremely high and an attribute is generated at each new timestamp, resulting in the updates of all the time series.

2.4 Consistent Top-k Queries

The classes of queries that retrieve objects which exhibit consistent performance over time are called consistent queries [11]. 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.

2.5 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 [2][10]. 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 [13]. 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 [14]. 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].

3. COMPARATIVE STUDY OF DISTANCE MEASURE

Distance Measure is a very important feature when it comes to time series analysis. The process of identifying the nearest neighbor to a particular reference series is to be carried out by a specific measure. In the previous works, a number of distance measures have been used.

Table 1. Comparative Study of Distance Measures Used

Euclidean distance (ED)	Each time instance as a dimension between reference series and sequences.	sensitive to noise
Edit distance with real penalty (ERP)	Measure the similarity between time series data	sensitive to noise
Edit distance on real sequences (EDP)	Measure the distance between two multidimensional sequences.	robust to noise
Dynamic Time Warping (DTW)	Indexes the sequences of same length in the warping window.	robust to noise

However, the noteworthy ones are Euclidean distance and Dynamic Time Warping. Besides these two, there are two other measures, namely; edit distance with real penalty and edit distance on real distance which has been taken into account by many researchers. It should also be noted that the distance measures strive towards identifying the nearest object and thus cater to the same functionality. Therefore, it is identified that these distance measures are susceptible only to sensitivity. Table 1. Provides a comparative study on the above mentioned distance measures based on their sensitivity and robustness.

As depicted in Fig 2, it can be noticed that the distance measures applied on the query types vary with sensitivity to noise. The graph is split into two halves, where the first half implies more sensitivity whereas the second half is shown to be increasing in robustness to noise. The noise sensitivity also decreases as the graph proceeds along the horizontal axis.

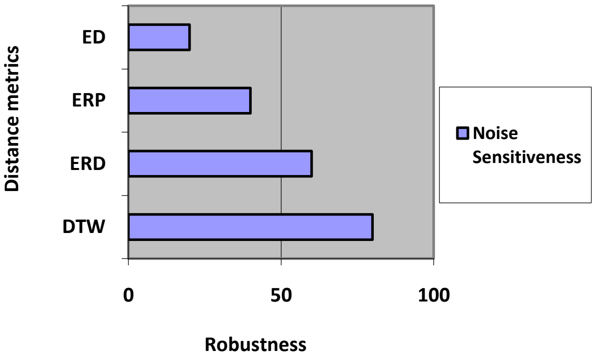

Fig 2: Distance Measure comparison based on noise sensitiveness

According to this survey, it can be concluded that Euclidean Distance and Edit Distance with real penalty are susceptible to noise sensitiveness whereas Edit Distance with real distance and Dynamic Time Warping are more robust to noise.

4. COMPARATIVE STUDY OF QUERY TYPES

The different types of queries vary over different attributes and satisfy a group of corresponding functions as depicted below in Table 2.

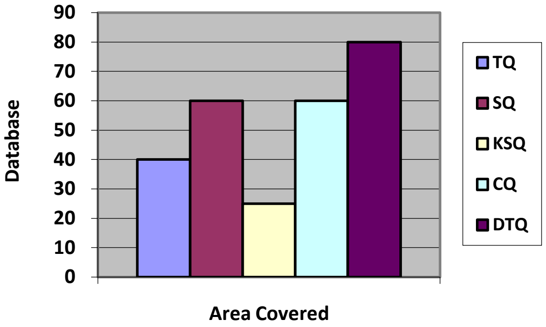

Fig 3: Query Type Comparison based on the area of database covered

Fig 3, shows the percentage of the vast database of queries covered by the different query types surveyed in this paper. One important factor to be noted is that, none of these queries satisfy hundred percentage of the entire database. This opens up a new door into the world of query types to reach the left out queries. Further findings of the above graph are mentioned below and also tabulated in Table 2.

Top-k queries and Snapshot queries attribute themselves to one dimension, however their query point customs itself to reach infinite values. Besides while Top-k queries over temporal data employ MVB trees and SEB trees for indexing, Snapshot queries imbibe R trees for the same purpose. However the performance of these queries is not entirely satisfactory and the reason points to its high cost of implementation. K Skyline queries correspond to only one set of time series queries and do not cover a vast database. However they employ two efficient algorithms which cater to the needs of this specific set of time series queries.

Consistent queries play a major role as it covers a large database and has been implemented in numerous real-life projects. It stands out with the fact that unlike the previously mentioned types of queries, consistent queries supports multidimensional data sets and work on results based on their consistency over a time interval.

Durable Top-k queries focus on the durability of the objects over historical time series. This holds importance as it incorporates the features of the foresaid queries and serves as an extension of these queries. These queries serve as the medium to overcome the triangular inequality which arises as a result of dynamic time warping.

However very little work has been carried out in this field, and this poses an open problem to work towards a more efficient method to handle durable queries.

Table 2. Comparative Study of Query types

Query Type	**Top-k Queries (TQ)**	**Snapshot Queries (SQ)**	**K-Skyline Queries (KSQ)**	**Consistent Queries (CQ)**	**Durable Top-k Queries (DTQ)**
Query Dimension	One Dimensional	One Dimensional	Two Dimensional	Two Dimensional	Multi-Dimensional
Distance Measure	Euclidean	Euclidean	Euclidean	Dynamic Time Warping	Dynamic Time Warping
Indexing Structure	MVB Trees, SEB Trees	R Trees	Radix Priority Search Tree	-	B+ trees, CJI trees
Algorithm	Brute Force	Branch and Bound Technique	On the Fly, View Materialization	Rank List, Bitmap Approach	Time Event Scanning, Query Space Indexing
Importance on Real Time Applications	Less	Least	Normal	More	Most
Limitations	Poor scalability of size, High construction cost	Unsatisfied query performance due to high cost	Extremely High Dimensionality	Predefined time series contents, pre-computation, equal weight time series	Slow updating of rank intervals, not efficient query indexing

5. CONCLUSION

This paper has surveyed all types of interval queries over time series and presented some of the important methodologies and tree traversals which yield efficient results in any of the top-k query processing. Retrieval of search engine results, stock market predictions, trend analysis, etc., are developed using top-k algorithms to get efficient results. Web query classification widely uses KNN algorithms to predict the best matching timestamps with the reference timestamps.

It has come to the conclusion that the selection of a particular query type is not carried out independently, however it depends upon the time series database it is being applied upon, the functionalities of the query type and the complexity of the algorithm used. It can be noted that based on the type of search engine and requirements, and the survey proposed by us, one can come up with the efficient query type for processing.

6. REFERENCES

[1]. Hao Wang, Yilun Cai, Yin Yang, Shiming Zhang, and Nikos Mamoulis, "Durable Queries over Historical Time Series," IEEE Computer Society 2014

[2]. 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.

[3]. 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.

[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]. 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.

[6]. A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," Proc. ACM SIGMOD Int'l Conf. Management of Data, 1984.

[7]. I.F. Ilyas, G. Beskales, and M.A. Soliman, "A Survey of Top-K Query Processing Techniques in Relational Database Systems," ACM Computing Surveys, vol. 40, no. 4, pp. 11:1-11:58, 2008.

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

[9]. B. Jiang and J. Pei, "Online Interval Skyline Queries on Time Series," Proc. IEEE Int'l Conf. Data Eng. (ICDE), 2009.

[10]. E. Keogh, "Exact Indexing of Dynamic Time Warping," Proc. 28th Int'l Conf. Very Large Data Bases (VLDB), 2002.

[11]. 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.

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

[13]. 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.

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

[15].