Survey On Graph based Searching Techniques

Poonam S .Sonawane, Ranjana P. Dahake

ABSTRACT

Graph mining is important in variety of applications such as social network analysis, bibliography analysis, scientific graph databases etc. Keyword based searching is widely used technique in graph mining. The graph containing time interval information is called as temporal graph. Mining temporal graphs, get answers to the time based keyword queries and generate archived formatted result is required in variety of applications. In this literature work graph mining strategies, temporal graph mining strategies, graph storage and query techniques has been studied.

Keywords: temporal graph, graph keyword search, ranking functions, graph data archiving

I. INTRODUCTION

Summarized report generation and archived results extraction from large graph dataset is required in many domains such as social network analysis, bibliography analysis, scientific graph databases etc. A large dataset is generated in variety of domains. In collaborative projects the previous data can be kept in archived format and then extracted when required. Apart from time based archive data extraction, user may want to fire a query on temporal graphs to relate existing and current scenario. This is required to analyze how things are changed also helps to observe lifeline of an object/product. With the help of temporal information trend analysis can be done. This time wise data also helps in prediction theory.

In existing literature work lot of work has been done on keyword search on graph. This keyword search is not sufficient to answer the temporal queries.

Consider some examples of temporal queries:

1. Find the papers on graph theory published after 2016.
2. Find people employed in Google after 2016.

Temporal relation model[2] and temporal xml model[3] were used in literature generate to answer such type of temporal queries. The relational database can generate answer for such queries but many applications have graph structure data. The graph structure data cannot be efficiently handled by the relational database.

On graph dataset keyword based search technique is employed in many cases. For answer generation to temporal queries the graph snapshots are generated for defined time interval. The temporal aspect is covered using graph snapshot. The query is executed using keyword based searching on desired number of graph snapshots. Graph based temporal information storage require extensive storage space. For processing such temporal queries, huge number of traversal is required based on matching the number of snapshots with the desired temporal query requirement.

For result extraction from a graph using a query, requires some formatted query as an input to the system, so that system can parsed the query and fire it on dataset. The Sql type queries are one of the solutions. Sql queries are structured queries and widely used in database operations. The sql query has predefined syntax and It is studied by the database operators. Due to the structural complexity of sql queries, such queries are not suitable for any casual user. Casual users who are not computer experts should be able to express a query on temporal graph. Hence query format should be easy to understand to any casual user. Along with result extraction of matched keywords on dataset, the result should be in appropriateSstructure like in sorted format by considering order of the attribute like time or object name.

A query on temporal graph includes multiple aspects such as:ease in query, graph format, processing efficiency, storage efficiency, query facilities such as sorting, ranking, collaborative data analysis etc. In the following literature work, the various keyword search techniques on graph data set have been discussed. In the further section temporal search techniques, its strategies and limitations are discussed.

II. LITERATURE WORK

Lot of research work has been done on keyword search on graph[4]. Minimal tree semantics strategy is used to filter the graph and to find relevant query keyword matching. Sub trees are generated from entire graph by matching the query keywords. Top k results are generated as per the tree height. Small height tree has high matching relevance. But such type of graphs do not encodes the temporal information in graph and lacks in temporal result generation.

For keyword matching unlike tree formation minimum length, path traversal techniques such as Dijkstras algorithm[6] is used to find shortest path between matched keyword nodes. A technique named as BANK[5] supports keyword based search on temporal graphs. This algorithm runs Dijkstras algorithm multiple times for iterating path from every query keyword node. Hence this method takes longer time for result extraction.

Temporal databases were proposed to define various version or for trends identification. Relational Databases, graph and/or semi structured data format such as xml were proposed to preserve temporal information. A transaction time or a valid time is included as one more touple in dataset. Temporal database structure is proposed to preserve temporal information in database. This structures follows the relational database structure along with the transaction time or valid time included in every tuple. TQUEL[7], TSQL2[8] and SQL3[9] languages were proposed to fire query on such temporal dataset format. But these query syntax are quite complicated and not convenient for casual user. This temporal relational database format is also unable to handle graph data structures.

Temporal graphs are also called as dynamic graph or time evolving graph. Lots of techniques[10][11] have been proposed to fire specific type of queries on temporal graph. User can get a single node information from a graph in terms of its time varying history [10]. Reachability of nodes, sub-graph matching queries are again type of temporal graph processing. Distributed systems are used to store huge graph data,Graph processing on distributed environment is also proposed in literature[11].

Keyword based temporal graph mining is done in[12]. In this technique temporal graph snapshots are stored independently. The data clusters were generated in prior step as per the temporal information. This technique uses dijkstras algorithm for finding shortest path for marched nodes. This algorithm was executed distinct graph snapshot independently and collective results are generated at the end.

W. Huo and V. J. Tsotras.[13] proposes a solution for shortest path problem over temporal graphical data. For such temporal data storage, graph snapshots are used. A clustering is performed over graph snapshot and the collection of these graph snapshots stored independently. The Dijkstras algorithm is used to find to find shortest path on cluster result. If no match found with cluster result then individual graph search technique is applied. This process is time consuming process and each graph snapshot is stored independently on disk and hence high memory storage is required for such technique.

Road network or communication network also generates temporal graph. In such graphs, nodes connectivity varies with respect to time. Along with the connectivity edge weights are also varies. Processing of such dynamic graph is also proposed in literature[14].

Large graph those cannot be loaded in the memory for processing are processed using partitioning strategy. The issues and challenges of temporal graph partitioning and structural graph partitioning are also discussed in literature[15].

Ziyang Liu, Chong Wang and Yi Chen[1] proposes a method to overcome the problem of storage space and execution time. This technique combines more than one graph snapshot and generates a single graph structure. This is directed graph, it contains: nodes and edges denoted with one or more time interval. This system helps to support queries containing temporal aspect of data. A technique is proposed to answer user queries and provide archived information based on the relationship among 2 or more intervals. In this system graph nodes and edges represents one or more time intervals. Custom query structure is proposed to fire a query over temporal graph data. System successfully parses the query if appropriate syntax is provided. The query syntax includes query keywords, predicate and ranking functions. Predicate includes the search constraints and ranking function includes the temporal filtering criteria. Ranking provides sorting of data in ascending and descending order of time, relevance and duration.

III.ANALYSIS AND PROBLEM FORMULATION

Lot of applications generates graph dataset. Such graph dataset cannot be efficiently handled by the traditional relational database systems. Such dataset is stored in xml or graph format. User will not be able to fire structural query like sql on such graph dataset.

In literature work, lot of system were focuses on keyword based search on graph dataset. This keyword search systems were extract the results by matching graph node values. But such keyword based queries were unable to handle temporal aspects of graph dataset.

Temporal graph snapshots were proposed in literature to deal with temporal graph based queries. These graph snapshots were stored and processed separately. Such applications were require high storage space and retrieval time for keyword based temporal queries.

All graph snapshots are stored together to overcome these problems. This improves storage and execution efficiency. Custom query syntax provide sorting and ranking function. But these operations are not sufficient to answer user's aggregate queries. Along with the collecting temporal answers from entire graph dataset, system should provide sorted and formatted archived temporal result using ranking and aggregate functions such as sum ,count, avg, etc like sql queries.

IV.CONCLUSION

Keyword based result extraction is proposed in literature, is insufficient to answer temporal queries. Different mechanism needs to be applied, to deal with temporal queries. Graph should contain time information along with keyword or weight information with every nodes and edges. Lot of work has been done for efficient graph data storage and query processing. Relational query syntax is complicated and hence not easily useful for any casual user and hence custom syntax were proposed in literature. Very limited functions are incorporated with the custom syntax. There is need to enrich the custom query formats in a very simplest manner and provide more options to user to get desired results.

V.REFERENCES

[1] Ziyang Liu, Chong Wang, Yi Chen, "Keyword Search on Temporal Graphs," in IEEE Transactions on Knowledge and Data Engineering Vol. 29, No: 8, Aug. 2017

[2] C. S. Jensen, R. T. Snodgrass, and M. D. Soo. The TSQL2 Data Model. In The TSQL2 Temporal Query Language. 1995.

[3] F. Rizzolo and A. A. Vaisman. Temporal XML: Modeling, Indexing, and Query Processing. VLDB J., 17(5):1179–1212, 2008.

[4] K. Golenberg, B. Kimelfeld, and Y. Sagiv. Keyword Proximity Search in Complex Data Graphs. In SIGMOD Conference, pages 927–940, 2008.

[5] B. Kimelfeld and Y. Sagiv. Finding and Approximating Top-k Answers in Keyword Proximity Search. In PODS, 2006.

[6] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword Searching and Browsing in Databases using BANKS. In ICDE, pages 431–440, 2002.

[7] R. T. Snodgrass. The Temporal Query Language TQuel. ACM Trans. Database Syst., 12(2):247–298, 1987.

[8] C. S. Jensen, R. T. Snodgrass, and M. D. Soo. The TSQL2 Data Model. In The TSQL2 Temporal Query Language. 1995.

[9] Tsql2 and sql3 interactions. http://www.cs.arizona.edu/people/ rts/sql3.html.

[10] G. Koloniari, D. Souravlias, and E. Pitoura. On Graph Deltas for Historical Queries. CoRR, abs/1302.5549, 2013.

[11] A. Fard, A. Abdolrashidi, L. Ramaswamy, and J. A. Miller. Towards Efficient Query Processing on Massive Time-evolving Graphs. In CollaborateCom, pages 567–574, 2012.

[12] C. Ren, E. Lo, B. Kao, X. Zhu, and R. Cheng. On Querying Historical Evolving Graph Sequences. PVLDB, 4(11), 2011.

[13] W. Huo and V. J. Tsotras. Efficient Temporal Shortest Path Queries on Evolving Social Graphs. In SSDBM, pages 38:1–38:4, 2014.

[14] B. Ding, J. X. Yu, and L. Qin. Finding Time-Dependent Shortest Paths over Large Graphs. In EDBT, pages 205–216, 2008.

[15] M. Steinbauer and G. Anderst-Kotsis. DynamoGraph: A Distributed System for Large-scale, Temporal Graph Processing, its Implementation and First Observations. In WWW, 2016.