IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727
PP 07-09
www.iosrjournals.org

Effctive Query Answering For Graph Patterns Using Views

Reshma Ravi[1], Remya R[2]
[1](Department of Computer Science, College of Engineering Perumon, Kollam, India)
[2](Department of Information Technology, College of Engineering Perumon, Kollam, India)

Abstract: *Relational and semi structured data can be effectively answered using Answering Queries Using Views. This paper focuses on the problem of solving graph pattern queries with the help of graph simulation. A pattern query can be answered using views if and only if the pattern query is contained in the views. This paper investigates efficient algorithms for determining the containment problem (minimal and minimum) of pattern queries and also the maximally contained rewriting algorithm for finding approximate answers. These methods are able to efficiently answer large real world graphs.*

Keywords: *Pattern containment problem, Views, Graph pattern query, Directed Graph.*

I. Introduction

Answer Generation using views has been studied for relational, XML and semi structured data. If a query Q and a set of views V= {V_1, V_2,…..,V_n}is given, the idea is to find another query A such that A is equivalent to Q and also A only refers to views in V. If such a query A exists then Q (D) can be answered using A without accessing database. Answering queries using views helps to find answer for real life social graphs easily because they are typically large and distributed. This method helps to query big data, irrespective of the size of underlying data. Graph pattern queries have been widely used in social network analysis. The most important problem related to the real word social graphs are large size and usually they are distributed. For e.g. Amazon has more than 2 million users with 140n billon links, here the data is distributed to different data centers across the world. The major challenge for the social network analysis is how to cope up with the large size of social graphs. Answering pattern queries using views provided an efficient solution for this problem.

In traditional approach answering pattern queries from graphs takes O ($|Q_S|^2$ + $|Q_S||G|$+$|G|^2$) time to compute Q_S (G).To identify a match set we have to perform a number of join operations. This increases the computational time .The advantage of our technique is that we only need to visit the views in V (G), without accessing the large graph all the time. The major aspect related to answering queries using views is to decide whether a given pattern query can be answered using a set of views. If a pattern query Q_s and a set of views V= {V_1,……..,V_n} is given, then Q_s can be answered if and only if Q_s is contained in V. For this a notion of pattern containment is proposed instead of the traditional query containment. This help to compute Q_s in O (|Qs||V (G)| +| V $(G)|^2$) time without accessing G. This helps to find Q_s in minimum computational time than in the traditional approach.

To decide which views in V to use we identified three fundamental problems. They are (1) Containment Problem (to decide whether a given query is contained in views). (2) Minimal containment (to find out a subset of V that minimally contains Q_s). (3) Minimum containment (To find out a minimum subset of V that contains Q_S). Maximally contained rewriting helps to find approximate answer for the query Qs, even if it is not contained in the views. If a query Q_s and a set of views V= {V_1,……,V_n} is given and Q_s is not contained in views. Then we have to find another Query Q_s^1 such that it is a sub query of Q_s and also it is contained in the set of views V. Query processing has two view based approaches. They are query rewriting and query answering. If a query Q and a set of views is given, the idea of query rewriting is to find another Q^1, if the query is not contained in the views. This helps to find approximate answer for the query. Query answering is to find Q_s using another query A which is equivalent to Q.

Graph simulation can be done in different ways like Distributed graph simulation, Graph compression, Incremental maintenance and bounded evaluation. In the case of Graph compression on big graphs, query summarization and query compression are used to reduce the search space. Real life graphs are updated frequently for this purpose we are using Incremental View Maintenance. This incremental view maintenance helps to maintain views when graph are updated. The view based approach can be combined with the existing distributed, incremental and compression techniques for effectively answer Queries.

II. Existing System

In [1] A.Y Halevy provided an efficient method of answering queries using a set of previously defined materialized views. In the case of query optimization, rewriting a query makes the execution more effective. The major challenge in this work is to extend this work to handle a large number of complex views and another challenge is to extend the data integration algorithm to effectively choose the best rewriting algorithm to answer the quey.

[2] X. Wu studied about answering queries for XML data has been studied .This paper tries to overcome the drawbacks of the previous papers, by setting the inverted list model for evaluating queries on a large persistent XML data. In this approach for materializing the views only those XML tree nodes that occur in the answer to the view is included in the inverted list model. A new time and space efficient algorithm is developed for answering queries. Optimization techniques are proposed to minimize the storage space and also the redundant views can be avoided using Bitmaps.

[3] This paper studied about doable and undoable for distributed graph simulation. This approach provides algorithm whose response time and data shipment are not a function of G. Experimental studies shows that these algorithms scale well with the large real world graphs. Also studies show that the Distributed simulation is Partition Bounded. This paper studied about the fundamental problems of the graph simulation. Proposed possibility and impossibility theorem. Scalability and efficiency of the algorithm is checked.

In [4] a new algorithm is proposed for rewriting the semi structured queries (Q) that access the semi structured views V and are equivalent to Q. In the first step the content mappings (which are used to produce the candidate rewriting) are used. And checks whether the composition is equivalent to the original query or not. This technique is complicated due to the lack of the schema and of structuring capabilities of TSL views. Our algorithm uses Containment mappings, the Chase and the query composition for efficient query answering.

In [5] the research has been done to revisit the Tree Pattern [TP] queries .The answer which is obtained by evaluating the annotated views is similar to the answer over the original View Analysis. The study proposed an algorithm for identifying the redundant view answers. In this approach V is divided into a finite set of sub views and finding MCR $_s$ of Q using each of the Sub views. In this approach a new technique is proposed for identifying the redundant view answers, which can be ignored while evaluating the maximally contained Rewriting.

In [6] a new method is proposed for incremental solutions for graph pattern matching based on simulation sub graph isomerism and bounded simulation. This approach developed incremental algorithms for batch updates and patterns. Incremental graph pattern matching helps to identify the answers to the queries even though the views are updated. This technique helps to apply graph simulation independent of the graph size. Incremental algorithms help to calculate changes in the matches when the graph is updated this helps to minimize unnecessary recommendation.

[7] Query preserving graph compression is developed for large real world graphs. In this approach efficient algorithms are developed for the reachability of queries and graph pattern queries. This method reduces the query processing time and the storage space. Incremental techniques are used for maintaining the compressed graphs .Developed compression for queries like reachability and graph pattern queries using graph simulation.

[8] This paper focuses on presenting a new algorithm for answering the queries using views and also implemented experimental evaluation for such algorithms. Existing algorithms like Bucket algorithm and inverse rules algorithm have some limitations, MiniCon algorithm overcomes this limitations. This technique can be used for large scale problems. Minicon algorithm efficiency is computed experimentally.

[9] View based query processing in semi structured data is studied in this approach. A framework is set between the query answering and query rewriting. Perfect rewriting is a co-NP function. Two methods are proposed for PTIME rewriting. In the first method rewriting are expressed in Datalog but in the second method rewritings are formulated as the unions of the conjunction of regular path queries. Query is answered directly from the view extensions.

In [10] we studied about Graph Structured Database (GSDB). A new algorithm is developed for GSDB views. To illustrate the fundamental problems developed an incremental algorithm for a graph simulated views. Due to space limitation, materialized views cannot be maintained. To achieve fast query response time, we have to materialize the views. This paper proposed incremental maintenance techniques for the semi structured data. It is efficient to use incremental maintenance algorithm to the view than to recomputed the query from the database. Several optimizations to the incremental maintenance algorithm are possible. Algorithm identifies the needed view changes from the information available on view specification and the update operation.

III. Proposed System

Answering pattern queries using views is an effective technique for querying large graphs. In this paper query answering using views has been done with the help of graph simulation. A notion of pattern containment is used instead of Query containment to characterize the pattern query problems. It is sufficient to determine whether a query Q_s is contained in the views.

Algorithm Matchjoin is to find a match set between the data graph (G) and the pattern query (Q_s). Approximation algorithms like contain, minimal and minimum are used to find out whether a query is contained in the view or not. Minimum containment helps to effectively reduce the redundant views. When the query is not contained in the views our approximation algorithm helps to find the answer at reasonable accuracy. Optimization strategy in this paper makes the view base matching up to 1.66 times faster.

Minimal algorithm is defined as follows; the algorithm returns either a non-empty subset V^1 of V that minimally contains Q_s or 0 to indicate that the Q_s is not contained in the views. Minimum algorithm identifies a subset V^1 of V such that (1) Q_s is contained in V^1 if Q_s is contained in V and card $(v^1) \leq \log(|E_p|)$, card $V_{opt.}$

IV. Conclusion

A notion of pattern containment is to characterize which pattern queries can be answered using views. Also this proposes three fundamental problems like minimal, minimum and pattern containment. When the pattern query is not contained in the views an efficient algorithm for computing maximally contained rewriting is proposed to find out the approximate answers. The experimental results have showed the efficiency and accuracy of the system. To find a practical method to query big graphs, one needs to combine different techniques like view-based, distributed, incremental and compression method. In our proposed methodology we are trying to include artificial intelligence. This helps to answer queries for biological community detection and social network analysis.

References

[1]. A.Y. Halevy’s, “Answering queries using views: A Survey, “VLDBJ, vol.10, no.4, pp-270-294 2001.

[2]. X. Wu, Theodoratos and W .H. Wang, “Answering XML queries using materialized views revisited”, in proc.18th ACM conf .inf. Knowl. Manage. , 2009, pp.475-484.

[3]. W. Fan, X. Wang, and Y. Wu, “Distributed graph simulation: impossibility and possibility”, proc. VLDB Endowment, vol.7, no.12, pp.1083-1094, 2014.

[4]. Y. Papakonstantinou and V. Vassalos , “Query rewriting for the semi structured data,” in the ACM SIGMOD int. conf. Manag. Data, 1999, pp. 455-466.

[5]. J. Wang, J, J, X. Yu’ and J.Li’s paper, “Answering tree pattern queries using views: A revisit,” in the 14 th Int. conf. Extending Database Technol., 2011, pp. 153-164.

[6]. W. Fan, X. Wang, and Y. Wu, “Incremental graph pattern matching ,” ACM Trans. Database syst., vol.38, no. 3,2013.

[7]. W. Fan, J. Li, X. Wang, and Y. Wu, “Query preserving graph compression,” in proc. ACM SIGMOD Int. conf. Manag . Data, 2012, pp. 157-168.

[8]. R . Pottinger and A. Y. Levy, “A scalable algorithm for the answering using views,” Very Large Data Bases, of 2000, pp. 485.

[9]. D. Calvanese, G. D. Giacomo, , M. Y. Vardi and M. Lenzerini, : “View based query processing and the constraint satisfaction,” in Proc. 15th Annu . IEEE Symp. Logic Comput. Sci., 2000, pp. 475-484.

[10]. Y. Zhuge and H. Garcia-Molina , “ Graph structured views and their Incremental maintenance algorithm ,” in 14th Int .Conf, held at 1998, pp. 116-125.