X-Warehouse: Building Query Pattern-driven Data Warehouse For XML Data

Ji Zhang
Department of Computer Science
University of Toronto
Toronto, Canada, M5S3G4
jzhang@cs.toronto.edu

Wei Wang
Nanjing Normal University,
Nanjing, China & PAMI Lab,
University of Waterloo, Waterloo,
Canada
wwang@pami.uwaterloo.ca

Han Liu
Department of Computer Science
University of Toronto
Toronto, Canada, M5S3G4
hanliu@cs.toronto.edu

ABSTRACT

In this paper, we propose an approach to materialize XML data warehouses based on the frequent query patterns discovered from historical queries issued by users. The schemas of integrated XML documents in the warehouse are built using these frequent query patterns represented as Frequent Query Pattern Trees (FreqQPTs). Using hierarchical clustering technique, FreqQPTs are clustered and merged to produce a specified number of integrated XML documents for actual data feeding. Maintenance issue of the data warehouse is also treated in this paper.

Keywords

Data Warehouse, XML Data, Query Patterns, Data Integration

1. Introduction

Building a XML data warehouse is appealing in that it provides users with a collection of semantically consistent, clean and concrete XML-based data that are suitable for efficient query and analysis purposes. However, building an enterprise wide XML data warehouse system is usually extremely time and cost consuming that is unlikely to be successful. Without a proper guidance on which information is to be warehoused, the resulting data warehouse cannot really well cater for user's needs in XML data acquirement.

To overcome these problems, we propose a novel XML data warehouse system, called X-Warehouse, by taking advantage of the underlying frequent patterns existing in the historical user queries. The general idea of our approach is: Given multiple distributed XML data sources and their global integrated schema represented as a DTD Tree, we will build an XML data warehouse based on the frequent query patterns. In doing so, the frequent query patterns, each represented as a Frequent Query Pattern Tree (FreqQPT), are discovered by applying a rule-mining algorithm. According to integration specification, FreqQPTs are clustered and merged to produce a specified number of integrated XML documents. The integrated XML documents in the data warehouse are updated when there are significant changes in the document content or user query patterns.

Related work in the filed of building and managing XML data warehouse in literature include [2, 3, 4, 5, 7]. In [2], a semi-automated approach to building conceptual schema for a data mart starting from XML sources is proposed. [3] uses XML to establish the Internet-based data warehouse system to solve the unavoidable defects of client/server data warehouse systems. [4] presents a framework for supporting interoperability among data warehouse islands in federated environment based on XML. [5] presents a change-centric method to manage versions in a web warehouse of XML data. [7] introduces a dynamic warehouse of XML data of the web supporting evaluation, change control and data integration. However, none of these works explore the historical user's query patterns as a useful guidance in the process of warehouse construction and maintenance.

2. X-Warehouse

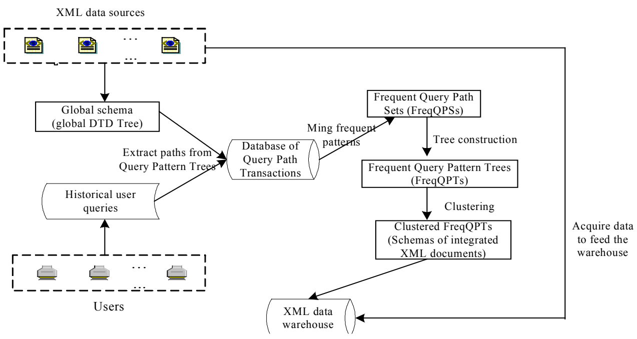
XML data sources
Global schema (global DTD Tree)
Extract paths from Query Pattern Trees
Historical user queries
Users
Database of Query Path Transactions
Ming frequent patterns
Frequent Query Path Sets (FreqQPSs)
Tree construction
Frequent Query Pattern Trees (FreqQPTs)
Clustering
Clustered FreqQPTs (Schemas of integrated XML documents)
Acquire data to feed the warehouse
XML data warehouse

Fig.1 An overview of X-Warehouse

The building of X-Warehouse based on frequent query patterns in users' queries involves the following four steps (See Figure 1):

(1) Transform user's queries into query path transactions;

(2) Discover Frequent Query Path Sets (FreqQPSs) in database of query path transactions;

(3) Produce schemas of integrated XML documents;

(4) Acquire data to feed the warehouse.

2.1 Transform User's Queries into Query Path Transactions

We express user's historical queries using XQuery-formatted statements. A *Query Path* is a path expression of a DTD tree that starts from the root of the tree. QPs can be obtained from the query script expressed using XQuery Statements. A query issued by the user can be transformed into a number of QPs, called a *QP transaction*. Having transforming a set of queries of into QP transactions, we now obtain a database containing all these QP transactions, denoted as D_{QPTra}. Like market-basket databases, this well-formatted database of query transactions makes it easy for us to apply rule-mining technique to discover significant rules revealing the users' query patterns.

2.2 Discover Frequent Query Path Set in Database of Query Path Transactions

We apply rule mining technique in D_{QPTra} to discover Frequent Query Path Sets (FreqQPSs) in D_{QPTra}. A FreqQPS contains the frequent QPs that jointly occur in D_{QPTra}. Frequent Query Pattern Trees (FreqQPTs) are built based on these FreqQPSs and serve as the building blocks of schemas of the integrated XML documents in the warehouse.

A *Frequent Query Path Set* (FreqQPS) is a set of QPs: $\{QP_1, QP_2, \ldots, QP_n\}$ that satisfies the following two requirements:

(1) Support requirement: *Support* $(QP_1, QP_2, \ldots, QP_n) \geqslant minsup;$

(2)Confidence requirement: *For each* QP_i, $Freq(QP_1, QP_2, \ldots, QP_n)/Freq(QP_i) \geqslant minconf.$

minsup and *minconf* are the minimum support and confidence thresholds specified by users. The FreqQPS mining algorithm is presented in Figure 2.

```
Algorithm MineFreq ($D_{QPTra}$, minsup, minconf)
$FreqQPS^1=\{QP|QP \in D_{QPS}, SatisfySup(QP)=true\}$;
$i$=2;
WHILE (1) DO {
    $CanQPS^i$=CanQPSGen($FreqQPS^{i-1}$);
    $CanQPS^i = CanQPS^i - \{QPS^i|$ SubSetNumber($QPS^i, FreqQPS^{i-1}) < i\}$;
    $FreqQPS^i=\{QPS^i \mid QPS^i \in$ CanQPS$^i$,SatisfySup($QPS^i$)=true AND SatisfyConf($QPS^i$)=true$\}$;
    $FreqQPS^{i-1} = FreqQPS^{i-1} - \{QPS^{i-1}| QPS^{i-1} \subseteq QPS^i, QPS^{i-1} \in FreqQPS^{i-1}, QPS^i \in FreqQPS^i\}$;
    $i$++;
    IF ($CanFreqQPS^{i-1} = \phi$ ) THEN Break;}
MaxItemset=i-2;
IF (MaxItemset$\neq$0) THEN
    FOR ($i$=1;$i \leqslant$MaxItemset; $i$++)
        Return ($FreqQPS^i$);
```

Fig.2 Algorithm of mining Frequent Query Path Sets

We employ the algorithm presented in Figure 2 to generate FreqQPSs from D_{QPTra}. The *n*-itemset QPS candidates are generated by joining (*n*-1)–itemset FreqQPSs. A pruning mechanism is devised to prune away from the candidates the *n*-itemset QPSs that do not have *n* (*n*-1)-itemset subsets in the (*n*-1)-itemset FreqQPS list. In other words, if one or more (*n*-1)-subsets of an *n*-itemset QPS candidate are missing in the (*n*-1)-itemset FreqQPS list, this *n*-itemset QPS cannot become a FreqQPS.

The *n*-itemset QPS candidates after the pruning are evaluated in terms of the support and confidence requirements to decide whether or not it is a FreqQPS. The (*n*-1)–itemset FreqQPSs are finally deleted if they are subsets of some *n*–itemset FreqQPSs.

After we have obtained a number of FreqQPSs, their corresponding Frequent Query Pattern Trees (FreqQPTs) will be built. Given a FreqQPS, its corresponding *Frequent Query Pattern Tree* (FreqQPT) is a rooted tree *FreqQPT=<V, E>*, where *V* and *E* are its vertex and edge sets, which are the union of the vertexes and edges of QPs in this FreqQPS, respectively. The root of FreqQPT is the root of its constituting QPs.

2.3 Produce Schemas of Integrated XML Documents

When all the FreqQPTs have been mined, the schema of the integrated XML document will be built. We choose to build a few, rather than one, integrated XML documents from the FreqQPTs mined, making the integration more flexible. We use agglomerative hierarchical clustering paradigm for this end. The basic idea of agglomerative hierarchical clustering is to begin with each FreqQPT as a distinct cluster and merges two closest clusters in each subsequent step until a stopping criterion is met. There are three possible scenarios of FreqQPT merging: (i) The two FreqQPTs have the same root; (ii) The root of one FreqQPT is ancestor node of the other FreqQPT' root; (iii) Cases other than Case 1 and 2. We have devised efficient strategies to cope with each of the cases.

2.4 Acquire Data to Feed the Warehouse

The last step of building X-Warehouse is to read data from XML data sources when the schemas of the integrated XML documents are ready. Processing efforts such as standardization, data cleaning and conflict solving need to be performed in this step to make the data in warehouse more consistent, clean and concrete.

3 Updates of XML Data Integration

Maintenance of the integration of XML data is an important issue after the integration has been finished. To make the integrated XML document applicable over time, the regular update of the integration is required to cope with two changes: (1) the changes of the original XML data sources and (2) the change of query patterns of users, which are termed the *Content Change* and *Pattern Change*, respectively. We adopt the regular/periodical updating scheme: the updating process is triggered only when the content or pattern has been changed significantly over time in terms of two thresholds T_c and T_p (the degree of content change exceeds T_c or degree of pattern change exceeds T_p). In the content-driven updating process, the system has to replace the old data of the affected element or attribute in the integrated XML documents with the new ones. In the pattern-driven updating process, the new frequent pattern tree will be re-constructed.

4. Conclusions

A new method for building the query pattern-driven data warehouse for XML data, called X-Warehouse, is proposed in this paper. A rule-mining technique is employed to discover these frequent query patterns, based on which the schemas of integrated XML documents are built. Frequent query patterns are represented using Frequent Pattern Trees (FreqQPTs) that are clustered using a hierarchical clustering technique according to the integration specification to build the schemas of integrated XML documents in X-Warehouse. The updating scheme proposed ensures that X-warehouse is applicable as time evolves, allowing for changes in data content in X-warehouse and user's query patterns.

References

[1] M. Golfarelli, S. Rizzi, and B. Vrdoljak. Data Warehouse Design from XML Sources. In Proceedings of *DOLAP'01*, 2001.

[2] S. M. Huang and C.H. Su. The Development of an XML-based Data Warehouse System. In Proceedings of *IDEAL'02*, 2002.

[3] O. Mangisengi, J. Huber, C. Hawel and W. Essmayr. A Framework for Supporting Interoperability of Data Warehouse Islands using XML. In Proceedings of *the DaWaK'01,* 2001.

[4] A. Marian, S. Abiteboul, G. Cobena, L. Mignet. Change-centric Management of Versions in an XML Warehouse. In Proceedings of *VLDB'01,* 2001.

[5] L. Xyleme. A Dynamic Warehouse for XML Data of the Web. *IEEE Data Engineering Bulletin*, 2001.