Volume 4, Issue 11, November 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Mitigating Duplicity in Hierarchical Data Using XML Mining

Gajanan DnyanobaTemgire, Bharati Kale

Computer Department & University Pune,

Maharashtra, India

Abstract— Relational data has many variants like SQL server, Oracle, Mysql etc. Each relational database system has primary and foreign key concept by which the matching of records can be done. There are lot of services like OLAP have features to identify the duplicity of records. There are few models addressed and surveyed on the duplicate detection of hierarchical data like eXtensible Markup Language (XML). An algorithm XMLDup can be addressed a Bayesian network to determine the XML elements probability. The heterogeneous data duplication method can also be used for XMLDup to develop and guarantee hierarchical duplication using XML mining. This work can establish key stone in mitigating the duplication in large data sets. Here Bayesian network will be used for duplicate detection, and by experimenting on both artificial and real world datasets the XMLDup method will be able to perform duplicate detection with high efficiency and effectiveness. This method will compare each level of XML tree from root to the leaves. The first step is to go through the structure of tree comparing each descendant of both datasets and find duplicates despite difference in data.

Keywords— XML, Bayesian algorithms, optimization, deduplication, Heterogeneous data.

I. INTRODUCTION

As software world evolve, the electronic data usage, importance and security increased largely. Due to the extent and huge data usage the capacity and storage of data plays a vital role in representation and selection of data. There are many ways to represent and optimization of data. For relational database the normalization forms are used to optimize the data. Main challenge in the data storage is redundant storage of data. There is no benefit to keep the duplicate and redundant data. There are not key concepts of keys (like primary key, foreign key etc.) for Hierarchical database (XML) and that why it's a difficult to identify the similarity and duplication in XML. This leads to think in such direction which identifies the duplication data in the Hierarchical database. As popular and interoperable universal hierarchical data format is XML, our research revolves around the XML data duplication.

Relational data which stored in Relational database like SQL Server, Oracle has the keys like primary key, foreign key, unique and identity key are used to detect the duplication of records. Primary key is used to detect the exact match but our aim is to find duplicated which has similarity between records which might not be exact match but these entities are same. Our research interest and aim is of the hierarchical data system eXtensible Markup Language (XML). As XML is interoperable data format which is universally accepted data for storage and data transfer. It is not easy and straightforward in the hierarchical data to check and find the duplicate records as structure, semantic and culture of the XML data vary depends on the system to system. In XML we have hierarchical data using nodes, elements and attributes. Node has combination of values and child nodes. Child nodes again can have multiple values and again its child nodes.

We present here probabilistic duplicate detection algorithm for Hierarchical data called as XMLDup. This algorithm considers both similarity of attribute contents and the relative importance of descendent elements with respect to overall similarity score. We propose the Bayesian network for duplicate detection. We first construct Bayesian network model for duplicate detection and then show how this model is used to compute the similarity between XML object representations. Given this similarity, we classify two XML objects as duplicate if it is above the given threshold value. Schema matching is the task of identifying and discovering semantic relationship between elements of two or more schemas. It plays important roles for many database applications, such as data integration to identify and characterize inter-schema relationships between multiple (heterogeneous) schemas, data warehousing to map data sources to a warehouse schema, E-business to help map messages between different XML formats. Heterogeneous data matching and duplicate detection between relational data and hierarchical data is also vital part in future duplication detection of heterogeneous data.

II. RELATED WORK

Surveyed the state of the art for duplicate detection in hierarchical data, which is the focus of this paper. Among studies that deal with hierarchical data, we mainly find works focusing on the XML data model. The only exception is which focuses on hierarchical tables in a data warehouse. Early work in XML duplicate detection was mostly concerned with the efficient implementation of XML join operations.

Ahmed K. Elmagarmid, Senior Member, IEEE, Panagiotis G. Ipeirotis (2007) presented the survey on Duplicate record detection. In their study they cover similarity metrics that are commonly used to detect similar field entries, and presented an extensive set of duplicate detection algorithms that can detect approximately duplicate records in a database.

Also they covered multiple techniques for improving the efficiency and scalability of approximate duplicate detection algorithms. Ahmed K. Elmagarmid at [3].Thandar Lwin & Thi Thi Soe Nyunt(2010) presented project work on An Efficient Duplicate Detection System for XML Documents. They presented the process of detecting duplicate includes three modules, such as selector, preprocessor and duplicate identifier which uses XML documents and candidate definition as input and produces duplicate objects as output. The aim of this research was to develop an efficient algorithm for detecting duplicate in complex XML documents and to reduce number of false positive by using MD5 algorithm. They illustrated the efficiency of this approach on several real-world datasets. Thandar Lwin & Thi Thi Soe Nyunt [4].

Different approaches have been proposed for storing and querying XML documents using relational database systems. Analyzes the XML data and expected query workload to obtain a set of Schemas. Any data that can't be accommodated in these schemas are stored in overflow graphs. Decomposed the tree structure of XML documents into nodes and stored them in relational tables according to their types. Later, transformed the inverted index into relational tables and converted containment queries into SQL queries. XML often needs to be combined with structural relational data within a relational database that includes varying levels of support for XML data. It is possible to receive data in XML format and we need to process it together with existing relational data. A relational database provides a convenient archival repository for persistent XML data. Fethi Abduljwad, Wang Ning, Xu [5].

There are two levels of data definition; one level defines physical structure of data and the other is characterized by logical level commonly known as schema. A schema implies a plan. A relation schema is the logical definition of an entity that defines the entity name and its attributes with data types. The collection of these relational schemas is called database. Database schema means a Structure of a database that describes how its concepts, their relationships and constraints are arranged. The application of database schema is useful when there is a requirement to integrate different application specified databases. Concepts of various database schemas are defined according to specific domain and requirements at a particular time. Therefore, they possess strong differences from each other. This arise heterogeneity as a highlighted issue. Heterogeneity may be structural or semantic. Structural heterogeneity includes conflicts like type conflicts, dependency conflicts, key conflicts, behavioral conflicts or semantic conflicts, the differences among the databases that are related to the meaning, interpretation, and intended use of data. To overcome this heterogeneity problem, schema matching is performed. Schema matching is a process in which semantic correspondences are identified between elements of many database schemas. Saira Gillani, Muhammad Naeem, Raja Habibullah, Amir Qayyum[9]

Schema matching is the problem of generating correspondences between elements of two schemas. A schema is a formal structure that represents an engineered artifact, such as a SQL schema, XML schema, entity-relationship diagram, ontology description, interface definition, or form definition. A correspondence is a relationship between one or more elements of one schema and one or more elements of the other. There are many applications that require schema matching. In the database field, it is usually the first step in generating a program or view definition that maps instances of one schema into instances of another. For example, it arises in object-to-relational mappings, data warehouse loading, data exchange, and mediated schemas for data integration. Philip A. Bernstein, Jayant Madhavan, Erhard Rahm [10].

III. METHODOLOGY

Proposed research work has three different modules which will be presented here. We will have the three modules like Homogeneous hierarchical database schema structure, Heterogeneous hierarchical database schema structure, Heterogeneous database (Hierarchical and Relational database) schema. Below subsections depicts the system architecture of the modules. Hierarchical database in the research project will be eXtensible Markup Language (XML).

A. Homogeneous hierarchical database schema structure

XML Schema will have the homogeneous structure. XML schema in this module is going to be the identical schema and identical structure of the XML elements, attributes and sequence. Architecture of given module is described as below figure.

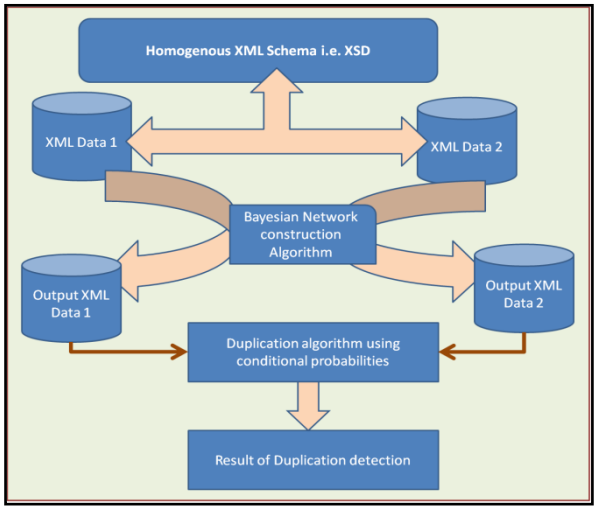

Figure 1 : Architecture for: Homogeneous hierarchical database schema structure

B. Heterogeneous hierarchical database schema structure

XML Schema will have the heterogeneous i.e. different structure. XML schema in this module is going to be the different schema and different structure of the XML elements, attributes and sequence. Architecture of given module is described as below figure.

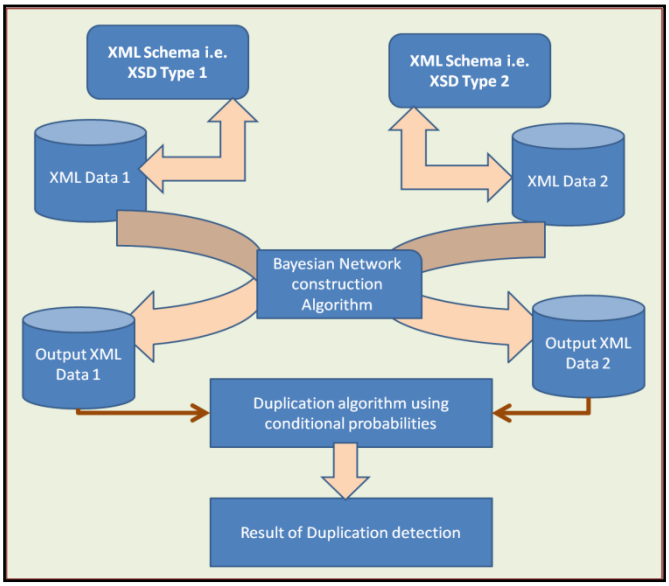

Figure 2 : Architecture for: Heterogeneous hierarchical database schema structure

C. Heterogeneous database (Hierarchical and Relational database) schema

In this module of research the duplicate detection would be between XML (hierarchical database) and relational database. There will be mapping mechanism by which the table structure and XML schema structure need to be matched. Architecture of given module is described as below figure.

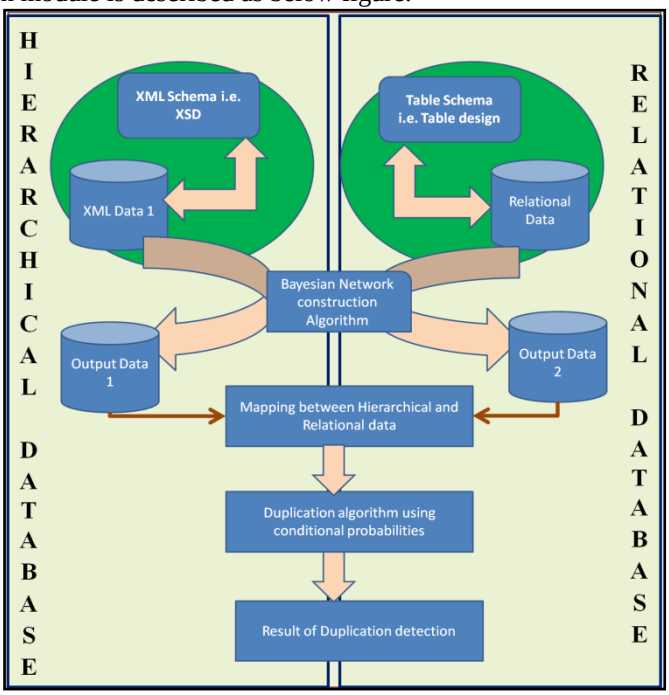

Figure 3 : Architecture for: Heterogeneous database (Hierarchical and Relational database) schema

IV. CONCLUSION

The shortcomings of exact matching methods introduced in literature work are overcome by a proposed new method XMLDup. This method presents a novel procedure for XML duplicate detection which contains various type of XML Schema. Using a Bayesian network model, this method is able to accurately determine the probability of two XML objects in a given database being duplicates. This model is derived from the structure of the XML objects being compared and all probabilities are computed taking into account not only the values contained in the objects but also their internal structure. To improve the runtime efficiency of XMLDup, a network pruning strategy is also used as basis. The heterogeneous database duplication system is the new forthcoming feature in a duplicate detection system. Introduction of the primary key, foreign key concepts in the XML data would be invention in the hierarchical database (XML).

REFERENCES

[1] Luis Leitao, PavelCalado, and Melanie Herschel, "An Efficient and Effective Duplicate Detection in Hierarchical Data" IEEE *TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, VOL. 25, NO. 5, pp. 1028-104 MAY 2013.

[2] Faten A. Elshwimy, Alsayed Algergawy, Amany Sarhan∗, Elsayed A. Sallam, "Aggregation of Similarity Measures in Schema Matching based on Generalized Mean" in *ICDE Workshops 2014,* pp. 74-79, 2014

[3] Ahmed K. Elmagarmid, Senior Member, IEEE, Panagiotis G. Ipeirotis, Member, IEEE Computer Society, and Vassilios S. Verykios, Member, IEEE Computer Society , Duplicate Record Detection: A Survey, *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, VOL. 19, NO. 1, pp. 1-16 JANUARY 2007.

[4] Thandar Lwin & Thi Thi Soe Nyunt University of Computer Studies, Yangon, Myanmar, 2010 Second International Conference on Computer Engineering and ApplicationsAn Efficient Duplicate Detection System for XML Documents, *Computer Engineering and Applications,* pp. 178-182 2010.

[5] Fethi Abduljwad, Wang Ning, Xu De School of Computer & Information Technology Beijing Jiaotong University, SMXIR: Efficient way of Storing and Managing XML Documents Using RDBMSs Based on Paths, *2010 2nd International Conference on Computer Engineering and Technology*VOL.1 , pp. 143-147 2010

[6] F. Naumann and M. Herschel, "an Introduction to DuplicateDetection" Morgan and Claypool, 2010

[7] A.M. Kade and C.A. Heuser, "Matching XML Documents in Highly Dynamic Applications," Proc. ACM Symp. Document Eng. (DocEng), pp. 191-198, 2008.

[8] L. Leita˜o and P. Calado, "Duplicate Detection through Structure Optimization," *Proc. 20th ACM Int'l Conf. Information and Knowledge Management*, pp. 443-452, 2011.

[9] Saira Gillani, Muhammad Naeem, Raja Habibullah, Amir Qayyum, " Semantic Schema Matching Using DBpedia", in *I.J. Intelligent Systems and Applications*, Vol No. 04, pp. 72-80, 2013

[10] Philip A. Bernstein, Jayant Madhavan, Erhard Rahm, "Generic Schema Matching, Ten Years Later", in *Proceedings of the VLDB Endowment,* Vol No4/11, pp 695-701, 2011.

[11] Pavel Shvaiko and J´er ome Euzenat, "Ontology matching: state of the art and future challenges", *in IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, pp. 158-176, JANUARY 2013

