Duplicate Detection in XML Data Using Probabilistic Duplicate Detection Algortihm

Nithya. P
IInd Year – M.E. CSE
Srinivasan Engineering College
Peramabalur
Tamil Nadu, India

Vinothini. K
Asst. Professor / CSE
Srinivasan Engineering College
Peramabalur
Tamil Nadu, India

Abstract

Duplication is the process of determine irreverent, incomplete, inaccurate data in XML database. To identify duplicates is the intricate problem in XML data, due to its hierarchical structure of data. A big defiance is to identify duplicate in XML data rather than the relation data by using foreign keys. In an existing system the core part is focus on discerning duplicates on whole objects in single table by using Dogmatix and fuzzy duplicate algorithm .To overcome this situation XMLDup is used, XMLDup use the Bayesian network to determine the probabilities of two XML element by including information about the structure. In addition to improve the aegis and efficiency of network pruning, a novel Decision tree induction and pruning strategy has used. The combination of three algorithms will discern the duplicates in complex structure both effective and efficiently.

Keywords

Duplication Detection, XML, Bayesian Networks, DogMatix, Fuzzy, Decision Tree Induction, Data Cleaning.

1. INTRODUCTION

Duplicates are multiple representation of same real world entities that can be differ from each other. Data quality depends on different category of recent error in origin data. In various applications such as, numerous business processes and decision are done by using Electronic data. Duplication detection is a nontrivial task because of duplicate are not exactly equal, due to error in the data. Therefore, we cannot use the common algorithm to detect exact duplicates. With the ever increasing volume of data and the ever improving ability of information systems to gather data from many, distributed, and heterogeneous sources and data quality problem abound. One of the most intriguing data quality problem in that of multiple, yet different representations of the same real-world object in the data. An individual might be represented multiple times in a customer database, a single product might be listed many times in an online catalog, and data about a single type protein might be stored in many different scientific databases. Such so-called duplicates are difficult to detect in the case of large volume of data.

Simultaneously, it decreases the usability of data and cause unnecessary expenses and also customer dissatisfaction. Such duplicates called fuzzy duplicates, in database management systems duplicate are exact copy of records.

For examples, consider the two XML elements describe as tree. Both are correspond to person object and are labeled *prs.* These elements have two attribute, namely date of birth and name. Advance XML element representing place of birth (pob) and contact (cnt). A contact consist of several address (add) and an email (eml), represented as a children of XML element of cnt. Each leaf element has text node which store actual data.

The objective of duplicate discovery is to detect the both persons are duplicates, regardless of the variation in the data. By comparing the corresponding leaf node values of both objects. Hierarchical association of XML data helps to detecting duplicate prs element, since successor elements can be detected to be similar. The goal is to reduce the number of pair wise comparison and to increase the efficiency of pair wise comparison. To compare two candidates, an overlay between their two sub trees is computed. It is not possible to match the same XML elements in different contexts. The weight is assigned to a match is based on a distance measure, e.g., edit distance for string values in leaves. The goal is to determine an overlay with minimal cost and not a proper substructure of any other possible overlay. To construct the Bayesian network, taking two XML elements as input, each rooted in the candidate element and having a sub tree that corresponds to the description. Nodes in the Bayesian network represent duplicate probabilities of a set of simple XML element, a set of complex XML elements and a pair of complex elements or pair of simple elements.

Probabilities are propagated from the leaves of the Bayesian network to the root and can be interpreted as similarities. As nodes either represent pairs or set of elements, the different semantics of a missing elements and as NULL values cannot be captured because the lack of an element results in the probability node not being created at all. The DogmatiX similarity measures Is aware of the three challenges that arise when devising a similarity measures for XML data. DogmatiX does not distinguish between the different semantics the both element optionality and element context allow. DogmatiX distinguishes between XML element types and real-world types so that all candidates of same type. A similar description pair is defined as descriptions whose pair wise string similarity. None of the similarity measures distinguishes between possibly different semantics caused by alternative representations of missing data or by different element context when computing a similarity cost. Another issue is infeasibility of tree edit distance measures for unordered tree.

2. ALGORITHM USED FOR IDENTIFYING DUPLICATE:

2.1. Probabilistic duplicate detection algorithm:

Probabilistic duplicate detection algorithm for hierarchical data called XML Dup. It considers both the resemblance of attribute content and the relative importance of descendant elements, with respect to similarity score. This algorithm is used to improve the efficiency and effective of run time performance.

2.2. Network Pruning Algorithm:

Pruning is a technique in machine learning that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances. To improve the Bayesian Network evaluation time, a loss less pruning strategy is proposed. This approach is loss less in the intellect of no duplicate objects are lost. Only object pairs incompetent of conquest a given duplicate probability threshold are redundant. Network evaluation is performed by doing a propagation of the prior probabilities in a bottom up fashion, imminent realization the top down node.To improve the Bayesian Network evaluation time, a loss less pruning strategy is proposed. This approach is loss less in the intellect of no duplicate objects are lost. Only object pairs incompetent of conquest a given duplicate probability threshold are redundant. Network evaluation is performed by doing a propagation of the prior probabilities in a bottom up fashion, imminent realization the top down node.

Algorithm 1. *Network Pruning(N,T)*

Require: The node N, for which we intend to compute the probability score; threshold value T, below which the XML nodes are considered non-duplicates

Ensure: Duplicate probability of the XML nodes represented by *N*

```
L ⟵ getParentNodes (N) {Get the ordered list of parents}
parentScore [n] ⟵ 1,Vn e L {Maximum probability of each parent node}
currentScore ⟵ 0
for each node n in L do {Compute the duplicate probability}
    if n is a value node then
        score ⟵ getSimilarityScore (N) {For value nodes, compute the similarities}
    else
        newThreshold⟵ getNewThreshold(T;parentScore)
        score ⟵ NetworkPruningðn;newThresholdÞ
    end if
    parentScore [n] ⟵ score
    currentScore ⟵ computeProbability(parentScore)
    if currentScore < T then
        End network evaluation
    end if
end for
return currentScore
```

2.3. Decision Tree Induction Algorithm

This algorithm is one of the machine learning approach used for maintaining security at the time of identify duplicate data in XML database. It identifies duplicate data and also removes noisy data, data cleaning, and incomplete data. Decision tree induction algorithm to the probability duplicate detection and network pruning algorithm ,in order to improve the performance of identify duplicate in the Bayesian network efficient and efficiently. Probabilistic duplicate detection algorithm for hierarchical data called XML Dup. It considers both the resemblance of attribute content and the relative importance of descendant elements, with respect to similarity score. This algorithm is used to improve the efficiency and effective of run time performance

3.RELATED WORKS:

In an existing, XML duplication detection was concerned with efficient implementation of XML join operation. A pioneering approach was anxiety on how to join the two set of similar element, and not on accurate the joining process of two similar object. The hierarchical structure of object representation is ignored; linear amalgamation of weighted correspondence is used to explanation for the relative consequence of the different field within the vector. Evaluate the algorithm both in terms of effectiveness and efficiency. First, to evaluate effectiveness by comparing it to a up to date duplicate detection system, called DogmatiX, that proved to be the most aggressive so far previous version. To appraise the efficiency of XML Dup when using planned pruning optimization of node and node ordering heuristics, varying the pruning factor of the nodes and automatically selecting the most adequate pruning factors on the nodes. The experiments are concluded with a discussion of the results. Testing the impact of data quality on duplicate detection is important to confirm the effectiveness of a given algorithm. To have shown that XML Dup manages to cope with error like duplicate erroneous elements without any significant degradation of the results and even performs effectively when dealing with reasonable amounts of missing data. Pioneering approaches is concerned only with joining the two set of the similar object not on the accurately the joining was done.

3.1 Dogmatix Algorithm:

The DogmatiX agenda aspire at both efficiency an effective in duplicate detection. These frame consist of three steps are *candidate definition, duplicate detection, and duplication detection.* It is used to identify duplicate object in whole object not in the separate object, and also does not provide the functionalities of the Bayesian network. It compares XML element based not only on their direct data values of the tree, but also on the similarity of their parents nodes, children nodes, structure, etc. An evaluation of algorithm using several heuristics validates approach.

3.2 Fuzzy Duplicate Detection Algorithm:

The algorithm is used to identify duplicates in relational database systems, data stored on single relational table using the foreign keys. Duplicate detection in a single relation does not straightforwardly apply to XML data, suitable to difference between the two models. For example, occasion of a same object type may have a variety structure at the prospect level, tuples within the relation have same structures. Algorithm for fuzzy duplicate detection is more complex structures, hierarchies in data warehousing, XML data, and graph data have recently emerged. Similarity measures that consider the duplicate data status of their direct neighbors.

DISADVANTAGES

- Duplicate detection is more complex in hierarchical structures.
- Common algorithm that cannot detect exact duplicate.
- Duplication detection in single relation that do not directly apply in XML data.

4. SYSTEM ARCHITECTURE:

The architecture shows the how to find the duplicate in XML data by using information regarding Bayesian network, network pruning and decision tree knowledge. A new XML data can be passed through the filtering module, by using some knowledge about the XML data. After filtering the XML data a noisy or inaccurate data can be removed and stored in a duplicate database.

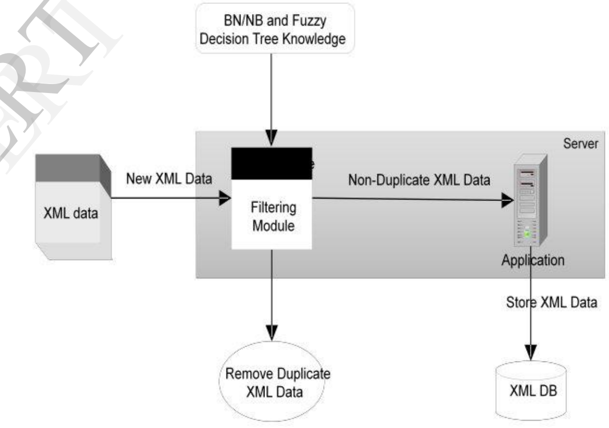

Figure 1: System Architecture

A non duplicate data can be stored in a original XML database by the administrator process or server. By using the knowledge of decision tree and network pruning, Bayesian network can find the duplicate in the XML data with efficiently and effectively.

SFinally, non duplicate data application can be stored in a XML database. To improve the run time performance of system by network pruning strategy.

5. PROPOSED SYSTEM:

To construct Bayesian network model for duplicate detection, is used to compute the similarity between XML object depiction. XML objects are duplicates based on the threshold values of two XML elements in the database. First present a probabilistic duplicate detection algorithm for hierarchical data called XML Duplication as XMLDup. This algorithm considers both the similarity of attribute contents and the relative importance of the elements, with respect to the overall similarity scores. Address the issue of efficiency of the initial solution by introducing a novel pruning algorithm and studying how the order in which nodes are processed affects runtime of their process. A major result is in that XMLDup now outperforms for existing algorithm, a previously more efficient state-of-the-art algorithm for XML duplicate detection.

It provides a more extensive evaluation of algorithms than in previous work. Propose a distance measure between two XML object representations that is defined based on the concept of overlays, this algorithm is more effective, its result must first be validated to guarantee a elevated superiority mapping. They can be seen as a directed acyclic graph, where the nodes characterize accidental variables of the tree and the edges represent dependencies between those variables of the tree. The Bayesian Network for XML duplicate detection is constructed for identify duplicate.

Contribution:

Probabilistic duplicate detection algorithm for hierarchical data called XMLDup. It considers the both similarity of attribute content and generation element, with respect to similarity score. 1) To address the issue of efficiency of initial solution by using novel pruning algorithm.2) The no of identified duplicates in increased, can be performed manually using known duplicate objects from databases.3) Extensive evaluation on large number of data sets, from different data domain.

The goal is to reduce the number of pair wise comparison is performed, and concerned with efficiently perform each pair wise comparison

5.1 Bayesian Network Constructions

Bayesian network provide a succinct requirement of a joint probability distribution. It is directed acyclic graph, the node represents random variables and edges represent dependencies between those variables. Two XML nodes are duplicates depends only on their values are duplicates and their children nodes are duplicates.

XML nodes being duplicates depend on 1) Whether or not their assessment is duplicates. 2) Whether or not their children are duplicates.

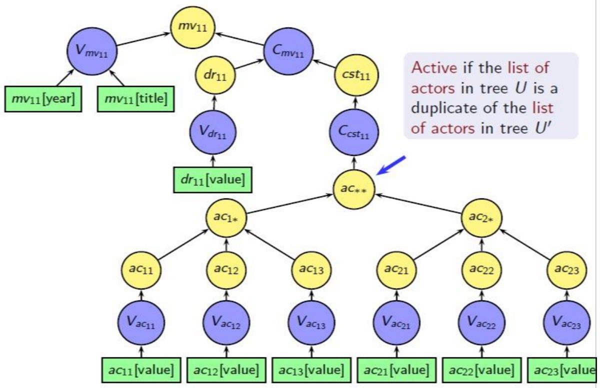

Figure 2: Bayesian Network Formation

A binary random variable can be active or inactive that can be assigned to each node, representing the values and children node is duplicate or non duplicates.

5.2 Contributing the Probabilities:

Assign a binary random variable to each node which takes the value 1 to represent the corresponding data in tree U and U' are duplicates and the value 0 to represent opposite. To decide if two XML tree are duplicates, the desire algorithm has to compute the probability of the root nodes being duplicates. To obtain the probabilities associated with the Bayesian Network leaf nodes, which will set the intermediate node probabilities, until the root probability is found between the nodes.

5.3 Network Pruning:

To improve the BN evaluation time by using lossless pruning strategy. By using lossless approach in the sense of no duplicate object are lost. Network evaluation is performed by doing a propagation of the prior probabilities, in bottom up fashion. Prefer the suitable order by which to evaluate the nodes, it makes the minimal number of estimate before choose if a pair of object is to be superfluous.

5.4 Duplicate Detection:

Evaluate the algorithm both in terms of efficiency and effectiveness. To evaluate the efficiency of XMLDup using node ordering heuristics, varying the different pruning factor, pruning optimization, selecting pruning factors. Data quality on duplicate detection is essential.

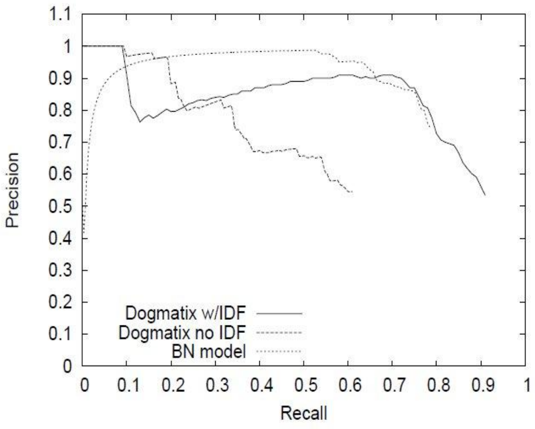

Figure 3: Duplicate Detection Rate

5.5 Evaluation

To evaluate effectiveness by comparing the XMLDup and Dogmatix algorithm. Determined the best node sorting strategy, using sorting strategy that can be applied to XMLDup on the real world entity, to improve the efficiency of the Duplication Data. Tested several variation of pruning factor and evaluate the actual impact on evoke. The reason its involuntary pruning factor optimization provided reliable enhancement in performance. The higher loss of recall for the artificial dataset, is same as original data sets.

6. CONCLUSION:

In this paper, the algorithm uses Bayesian network, network pruning, decision tree induction for improving the efficiency and also finding the probabilities of two elements being replica in secure manner .Bayesian network model is flexible, allowing the uses of different similarities measures and combining probabilities. To improve the efficiency of XMLDup by using pruning strategy. It can be applied in both ways, lossless approach is no impact on the accuracy of final result and a lossy approach is reduces the evoke. In the future work is to extend the BN model construction algorithm to compare. It is aimed to provide a complete solution for identifying the duplicate in hierarchical and complex structure by using the combination of probabilistic duplicate detection, network pruning and decision tree induction algorithm. Network pruning is used to improve the run time efficiency, at the same time it identify duplicates in XML elements. This duplicate in XML elements can be determined based on their probability of two XML elements by using network pruning and decision tree algorithm in Bayesian networks. It can applied in different structure.

6.1. REFERENCES

[1] Efficient and Effective duplicate detection in hierarchical data - Luı´s Leita˜ o, Pa´ vel Calado, and Melanie Herschel, VOL. 25, NO. 5, MAY 2013

[2] An Introduction to Duplicate Detection - F. Naumann and M. Herschel , 2010.

[3] An Overview of XML Duplicate Detection Algorithms - P. Calado, M. Herschel, and L. Leita˜o, vol. 255, pp. 193-224, 2010

[4] Data Cleaning: Problems and Current Approaches - E. Rahm and H.H. Do , vol. 23, no. 4, pp. 3-13, Dec. 2000.

[5] Domain-Independent Data Cleaning via Analysis of Entity-Relationship Graph - D.V. Kalashnikov and S. Mehrotra , vol. 31, no. 2, pp. 716-767, 2006.

[6] Structure-Based Inference of XML Similarity for Fuzzy Duplicate Detection - L. Leita`o, P. Calado, and M. Weis , 2007.

[7] Structure Aware XML Object Identification - D. Milano, M. Scannapieco, and T. Catarci, 2006.

[8] XML Duplicate Detection Using Sorted Neighborhoods - S. Puhlmann, M. Weis, and F. Naumann, pp. 773-791, 2006.

[9] Finding Similar Identities among Objects from Multiple Web Sources - J.C.P. Carvalho and A.S. da Silva, 2003

[10] Approximate XML Joins - S. Guha, H.V. Jagadish, N. Koudas, D. Srivastava, and T. Yu, 2002

[11] The Merge/Purge Problem for Large Databases - M.A. Hera ´ndez and S.J. Stolfo, pp. 127-138, 1995

[12] Duplicate Detection through Structure Optimization - L. Leita ˜o and P. Calado, pp. 443-452, 2011.

[13] Support Vector Regression Machines - H. Drucker, C.J. Burges, L. Kaufman, A. Smola, and V. Vapnik, vol. 9, pp. 155-161, 1996.

[14] Object-Level Ranking: Bringing Order to Web Objects Z. Nie, Y. Zhang, J.-R. Wen, and W.-Y. Ma, pp. 567-574, 2005

[15] Ranking Web Objects from Multiple Communities - L. Chen, L. Zhang, F. Jing, K.-F. Deng, and W.-Y. Ma, pp. 377-386, 2006.