ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Network Pruning-Detecting Duplicate Efficiently in XML Data

Ms.M.Lakshmipriya[*1] (M.E/CSE), Mr.G.Loganathan[2],M.E(AP/IT)
[*1,2] Muthayammal Engineering College, India
lpmlakshmipriya@gmail.com

Abstract

Duplicate detection is a non-trivial task in which duplicates are not exactly equal due to error in the data and objects. The existing system uses a method called XMLDup. It considers only the XML data files to detect duplicate and non duplicate files. This method uses Bayesian network model to determine the probability of two XML elements being duplicate. It also uses network pruning algorithm to increase the BN evaluation time. This algorithm achieve high precision and recall scores in terms of both efficiency and effectiveness. In the proposed work aimed to extend the BN evaluation time using machine learning algorithm.

Keywords: XML, Duplicate detection, Bayesian networks, Network pruning

Introduction

Duplicate detection is the way of detecting different entries in a data source representing the same real world entity. Detecting duplicate in the XML is slightly more complex than detecting duplicate in the relational data. XML data is semi- structured and is organized hierarchically. DogmatiX is one of the method for duplicate detection, which compares XML elements based on the direct data values but also the similarity of the parent, children etc [5]. In the DogmatiX framework, the problem is divided in to three components namely: Candidate definition, duplicate definition and duplicate detection. Candidate Definition defines which objects are to be compared. Duplicate Definition defines when two duplicate are considered as duplicate. Duplicate Definition determines how to efficiently find those duplicates.

Detecting and eliminating duplicate data is major problem in the area like data cleaning and data quality. Data elimination is very hard because several types of errors may occur namely, typographical errors and equivalence errors.

In the existing system, the duplicate detection algorithm called XMLDup. This algorithm uses Bayesian Network model for XML duplicate detection. XMLDup considers both the similarity of attribute contents and the descendant elements depends on the overall similarity score. Network pruning algorithm extend the BN evaluation time which will produce efficiency and effectiveness in terms of high precision and recall scores.

Existing Methods

The method for detecting duplicate data is to construct the Bayesian network model then compute the similarity between XML objects. Using this similarity we can classify two XML objects as duplicates if it falls above a threshold.

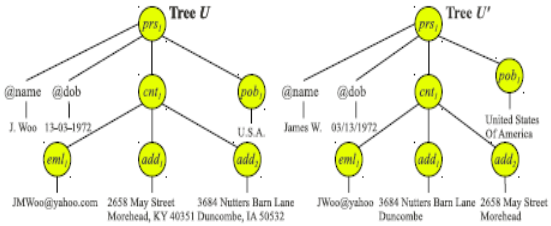

Fig. 1. Two XML elements that represent the same person. Nodes are labeled by their XML tag name and an index for future reference.

Bayesian Network Model

Bayesian Networks provide a specification of a joint probability distribution. It can be seen as a directed acyclic graph, where the nodes represent random variables and the edges represent dependencies between those variables. BAYESIAN

ISSN: 2277-9655
Impact Factor: 1.852

Network construction:

Basic assumption for XML duplicate detection ,the fact that two XML nodes are duplicates depends only on the fact that their values are duplicates and that their children nodes are duplicates [15]. Then, two XML trees are duplicates if their root nodes are duplicates. An XML tree is defined as a triple $U = (t, V, C)$ Where,• t is a root tag label, e.g., for tree U

• V is a set of (attribute, value) pairs. If the node itself has a value, we can consider it as a special (attribute, value) pair.

• C is a set of XML trees, i.e., the sub-trees of U. These subtrees are again each described by a triple.

First consider the XML nodes tagged prs. As illustrated in Fig. 2, the BN will have a node labeled prs11 representing the possibility of node prs1 in the XML tree U being a duplicate of node prs1 in the XML tree U0. Node prs11 is assigned a binary random variable. This variable takes the value 1 (active) to represent the fact that the XML prs nodes in trees U and U' are duplicates. It takes the value 0 (inactive) to represent the fact that the nodes are not duplicates. Then compute prior probability, conditional probability, final probability [6].

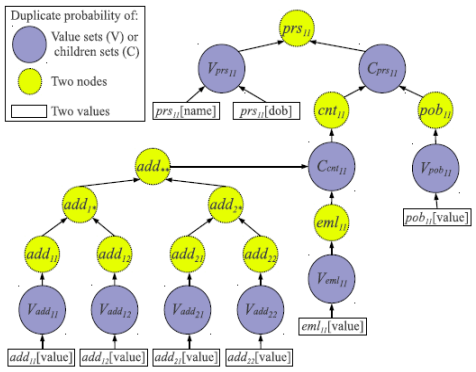

Fig. 2. BN to compute the similarity of the trees in Fig. 1.

Network Pruning Algorithm for BN

To increase BN evaluation time a lossless pruning strategy is used . This strategy is lossless in the sense that no duplicate objects are lost. Only object pairs incapable of reaching a given duplicate probability threshold are discarded. A network evaluation is performed by doing a propagation of the prior probabilities, in a bottom up fashion, until reaching the topmost node. The prior probabilities are obtained by applying a similarity measure to the pair of values represented by the content of the leaf nodes. Computing such similarities is the most expensive operation in the network evaluation and in the duplicate detection process in general. Therefore, the idea behind our pruning proposal lies in avoiding the calculation of prior probabilities, unless they are strictly necessary.The strategy is that before comparing two objects

All the similarities are assumed to be 1.

At every step of the process, maintain an upper bound on the final probability value.

At each step, whenever a new similarity is computed, the final probability is estimated taking into consideration the already known similarities and the unknown similarities that assume to be 1.

When verify that the network root node probability can no longer achieve a score higher than the defined duplicate threshold, the object pair is discarded and, thus, the remaining calculations are avoided.

The algorithm takes input as node N from the BN and a threshold T. It starts by gathering a list of all the parent nodes of N and assuming the duplicate probability score is 1. It then proceeds compute the actual probability value. If a parent node n is a value node then its probability score is simply the similarity of the value. If n also has parent node then its probability score depends on its own parent.

Algorithm 1. NetworkPruning(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
1: L ← getParentNodes(N) {Get the ordered list of parents}
2: parentScore[n] ← n L {Maximum probability of each parent node}
3: currentScore ← 0
4: for each node n in L do {Compute the duplicate probability}
5: if n is a value node then
6: score ← getSimilarityScore(n) {For value nodes, compute the similarities}
7: else
8: newThreshold ← getNewThreshold(T, parentScore)
9: score ← NetworkPruning(n, newThreshold)
10: end if
11: parentScore[n] ← score
12: currentScore ←computeProbability(parentScore)
13: if currentScore < T then
14: End network evaluation

ISSN: 2277-9655
Impact Factor: 1.852

15: end if
16: end for
17: return currentScore

Fig.3.Network Pruning Algorithm

Conclusion

Duplicate detection is a non-trival task in which duplicates are not exactly equal due to error in data and objects. The XMLDup uses a Bayesian network to determine the probability of two XML objects being duplicates. This model is composed from the structure of the objects. To improve the runtime efficiency of XMLDup, a network algorithm is used . This strategy is lossless in the sense that no duplicate objects are lost only object pairs incapable of reaching a given threshold are discarded. Every node should have the duplicate probability value as 1 before evaluation, this is called as pruning factor. Slightly lowering the pruning factor the system achieves high performance.The XMLDup does not consider different structure representation and the conditional probability is derived manually. So, in proposed the machine learning algorithm with support vector machine is used to extend the BN model construction.When compare to XML Dup, the machine learning algorithm is expected to calculate the objects with different structures and conditional probability is obtained automatically with the help of Support vector machine.

References

[1] *Ananthakrishna .R,Chaudhuri .S, and Ganti .V, "Eliminating Fuzzy Duplicates in Data Warehouses," Proc. Conf. Very Large Databases (VLDB), pp. 586-597, 2002.*

[2] *Carvalho J.C.P and da Silva A.S, "Finding Similar Identities among Objects from Multiple Web Sources," Proc. CIKM Workshop Web Information and Data Management (WIDM), pp. 90-93, 2003.*

[3] *Chen .L, Zhang .L, Jing F, Deng K.F, and Ma W.Y, "Ranking Web Objects from Multiple Communities," Proc. 15th ACM Int'l Conf. Information and Knowledge Management, pp. 377-386, 2006.*

[4] *Kalashnikov D.V and Mehrotra .S, "Domain-Independent Data Cleaning via Analysis of Entity-Relationship Graph." ACM Trans.Database Systems, vol. 31, no. 2, pp. 716-767, 2006.*

[5] *Kirkpatrick .S, Gelatt C.D, and Vecchi M.P, "Optimization by Simulated Annealing," Science, vol. 220, pp. 671-680, 1983.*

[6] *Leita˜o .L, Calado .P, and Weis .M, "Structure-Based Inference of XML Similarity for Fuzzy Duplicate Detection," Proc. 16th ACM Int'l Conf. Information and Knowledge Management, pp. 293-302,2007.*

[7] *Melanie Weis and Felix Naumann," Relationship-Based Duplicate Detection" Institut fur Informatik, Humboldt-Universit¨at zu Berlin Unter den Linden 6, D-10099 Berlin, Germany.*

[8] *Melanie Weis, Felix Naumann," Detecting Duplicates in Complex XML Data" Humboldt-University at zu Berlin Unter den Linden 6, 10099 nBerlin.*

[9] *Nie .Z, Zhang .Y, Wen J.R, and Ma W.Y, "Object-Level Ranking:Bringing Order to Web Objects," Proc. Int'l Conf. World Wide Web (WWW), pp. 567-574, 2005.*

[10] *Weis .M and Naumann .F, "Dogmatix Tracks Down Duplicates in XML," Proc. ACM SIGMOD Conf. Management of Data, pp. 431- 442,2005*