A Novel Approach for the Effective Detection of Duplicates in XML Data

Anju Ann Abraham[1,], S. Deepa Kanmani[2]
1PG Student, Department of Computer Science and Engineering, Karunya University.
2 Assistant Professor, Department of Computer Science and Engineering, Karunya University.

Abstract:
eXtensible Markup Language is widely used for data exchange between networks and it is also used for publishing data on web. Identifying and eliminating the duplicates has become one of the challenging tasks in the area of Customer Relationship Management and catalogue integration. In this paper a hybrid technique is used for detecting duplicates in hierarchically structured XML data. Most aggressive machine learning techniques is used to derive the conditional probabilities for all new structure entered. A method known as binning technique is used to convert the outputs of support vector machine classifiers into accurate posterior probabilities. To improve the rate of duplicate detection network pruning is also employed. Through experimental analysis it is shown that the proposed work yields a high rate of duplicates thereby achieving an improvement in the value of precision. This method outperforms other duplicate detection solution in terms of effectiveness.

Keywords: *Binning, duplicate detection, heterogeneous structure, network pruning, posterior probability, SVM, XML*

1. Introduction

eXtensible Markup Language (XML) is widely used in most of the business applications to exchange data within the network and it is also used for publishing data on web. Most of the time XML data comes with errors and inconsistencies inherent in the real world data. It is necessary to ensure the quality of data published on web as it is created from distributed and heterogeneous data source. Data quality however can be compromised by different types of errors, which can have various origins [1].

Identifying and eliminating duplicates has become one of the challenging tasks as the data may not look exactly similar. Now a day's data's are given different representation as a result identifying different representation of the same entity turn out to be a problem in the field of duplicate detection. It is essential to use a correct matching strategy for identifying if they refer to the same real world entity or not.

Due to the extensive applications in various fields, duplicate detection has been studied profoundly for data stored in relational tables. While detecting duplicates in relational structure the tuples are compared and their similarity scores are computed based on their attribute values. In most cases, it omits some of the data's as foreign keys are used to connect tables. Many algorithms where developed which considered hierarchical and semi structured data [2] [3] [4] [5].

Both the figures represent the same publication details. The main difference between figures is the way in which they are represented. In both the figures nodes are labeled by their XML tag name. Leaf elements have text node which stores the actual data. Ven1 and aut1 are child nodes which in turn act as parent node for vname1, vname2, vol1 and name1, id respectively. The objective of duplicate detection is to efficiently prove that both the publication depicted in the figures in different format is identical despite of their structure.

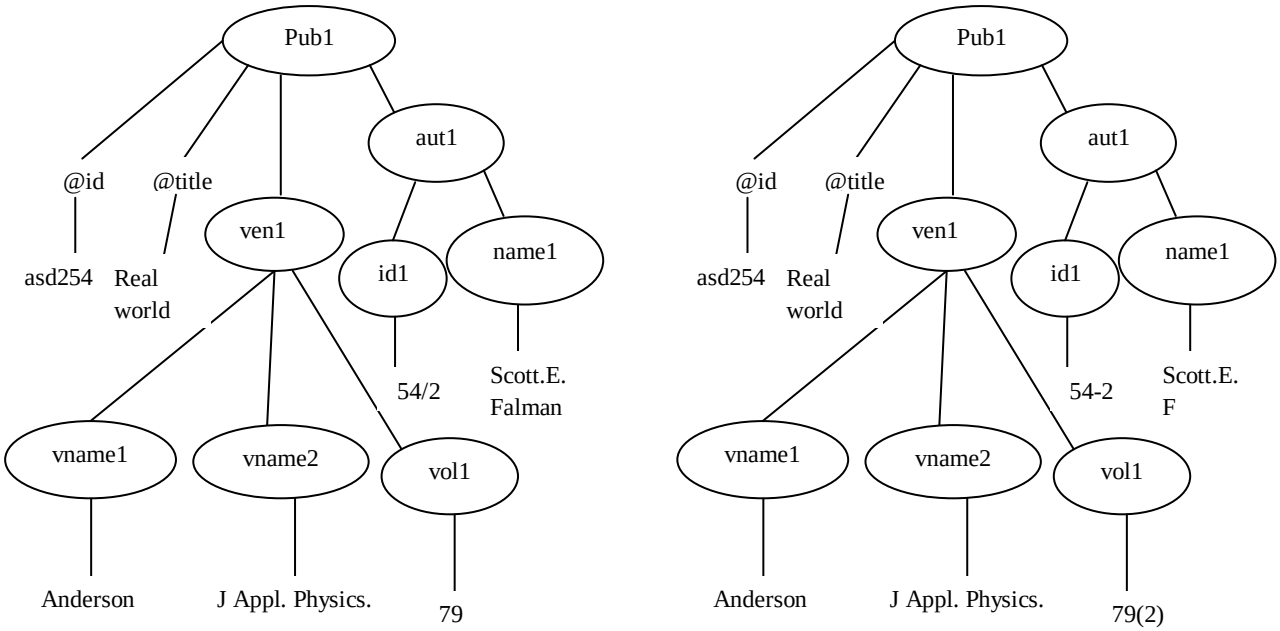

Figure1. XML element that represents the same publication detail in format.

A hierarchical representation of XML data is shown in the Fig. 1. The elements are represented in different format and they differ from each other in the way they are structured.

Contribution-In this paper a supervised machine learning algorithm known as SVM is used for deriving conditional probabilities. A method known as binning is used to convert the output of SVM into an accurate posterior probability [6] [7]. To obtain a better result in terms of effectiveness a network pruning [8] is applied. Unlike other works the main aim is to improve the performance of detecting duplicates of differently structured hierarchical data. In this paper performance is evaluated by comparing the recall and precision values of XMLDup and Dogmatix with the proposed hybrid method using SVM.

Layout -This paper is organized as follows: Section 2 presents the related work. Section 3 summarizes the hybrid methodology. Section 4 describes the experimental setup and result analysis. Finally, section 5 concludes the work and presents a suggestion for future work.

2. Related Works

In this section various duplicate detection algorithms and techniques are explained.

Delphi [9] is used to identify duplicates in data warehouse which is hierarchically organized in a table. It doesn't compare all pairs of tuples in the hierarchy as it evaluates the outermost layer first and then proceeds to the innermost layer.

D. Milano et.al, [5] suggested a method for measuring the distance of each XML data with one another, known as structure aware XML distance. Using the edit distance measure, similarity measure can be evaluated. This method compares only a portion of XML data tree whose structure is similar nature.

M. Weis et.al [2] proposed Dogmatix framework which comprises of three main steps: candidate definition, duplicate definition and duplicate detection. Dogmatix compares XML elements based on the similarity of their parents, children and structure. It also takes into account difference of the compared elements.

A novel method, XMLDup [3] is used for detecting fuzzy duplicates in hierarchical and semi structured XML data which considers the duplicate status of children and probability of the descendants being duplicates. Probabilities are derived efficiently using a Bayesian Network, which helps in determining the duplicates in XML data.

Network pruning [8] was the extension of XMLDup which was proposed to accelerate the Bayesian network evaluation time. In network pruning, a threshold is given and only those object pairs incapable of reaching the threshold was discarded. Proposed paper made a difference with this paper in the sense different structures are considered and conditional probability is estimated using SVM.

3. XML Duplicate Detection

The main goal of duplicate detection in XML data is to identify the XML element which represents the same real world entity. An XML document is considered as duplicates not only based on their structure but also on the way in which contents are represented. Each node is considered as duplicates based on their probability values. If estimated probability is above a given threshold value then it is considered as duplicate.

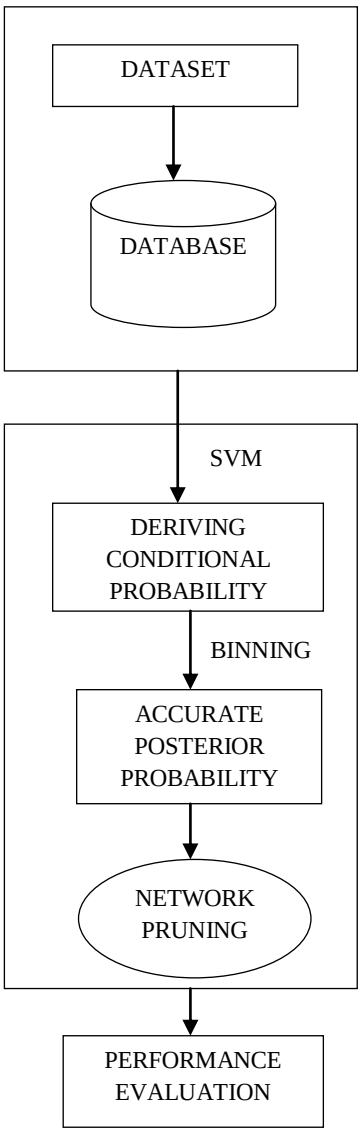

Figure 2 Overview of duplicate detection

Figure. 2 show the overview of detecting duplicates in XML data using the support vector machine classifier and network pruning.

3.1 Deriving Conditional Probabilities Using SVM

A XML document is considered as duplicates based on the conditional probability. Using SVM probabilities of different structure can be calculated efficiently. While applying SVM, conditional probability is obtained as the output which is converted into an accurate posterior probability using binning [10], [11].

While using SVM, the dataset was divided into two: training and testing sets. SVMs learn a decision boundary between two classes by mapping the training examples onto a higher dimensional space and then determining the best separating hyper plane between those spaces [10]. Given a test example 'x', the SVM outputs a score that measures the distance of 'x' from the separating hyper plane. The sign of the score indicates

to which class 'j' example 'x' belongs, where j = {1, 0}. The obtained score has to be converted into an accurate posterior conditional probability. For the conversion purpose a histogram technique known as binning is used.

In the figures given above node pub1 is considered as duplicates only when its child node and value node are duplicates. A child node is considered as duplicates only when their value nodes are duplicates. In short a node is considered as duplicates only when all its child node and value nodes are duplicates.

Four conditional probabilities are stated below are from [8]:

Conditional Probability 1 (CP1): the probability of values of a node being duplicates depends on three factors: 1) all the individual value nodes are duplicates 2) all the individual value nodes are not duplicates and 3) some of the value nodes are duplicates based on the threshold value . From the examples if id, title, vname's, vol, name are duplicates then the pub1 nodes are duplicates otherwise if all the nodes are non-duplicates then pub1 is considered as non-duplicates. If only some nodes are duplicates, then duplicate probability is based on a given value 'a'.

Conditional Probability 2 (CP2): the probability of a children node being duplicates, given that each individual pair of children is duplicates. Aut1 and ven1 are duplicates only if their child nodes are duplicates.

Conditional Probability 3 (CP3): the probabilities of two nodes are duplicates given that their values and children are duplicates. Pub1 are duplicates only if both the children node ven1 and aut1 are duplicates and the value node id and title are duplicates.

Conditional Probability 4 (CP4): the probability of a set of nodes of the same type being duplicates given that each pair of individual nodes in the sets is duplicates.

Using SVM probability of a node being duplicate or non – duplicate can be determined easily. SVM first evaluate the probability of the parent node being duplicates. If there is a probability for parent node to be duplicates then the corresponding probability of the child node and value node being duplicates are evaluated. If there is probability of the child and value node to be duplicates then then the parent node is considered as duplicates.

After all the conditional probabilities have been derived binning is used to convert the conditional probability to accurate posterior conditional probability. In binning method the training instances are first ranked according to their scores. Then the process continues by dividing the instance into 'n' subsets of equal size. The corresponding estimated probability P (j|x) is the portion of training instances that actually fit to the class that has been predicted for the test example [6].

3.2 Network Pruning

An algorithm proposed in [8] is used to prune the non-duplicate node that doesn't cross the threshold value. The algorithm takes a node N and if the node probability score falls above a given threshold value then it is considered as duplicates otherwise that node is discarded. Higher the similarity score there are more chances of missing out the duplicate pairs. By lowering the similarity score network can be evaluated faster as there are more chances of crossing the threshold value easily. Even though network can be evaluated faster there are chances of missing out duplicates which can be considered as a disadvantage.

4. Experimental Setup and Result Discussion

This section gives a brief description about the experiments performed using the dataset. The same dataset which was used [8] are taken for the process of duplicate detection. The experimental evaluation was performed on Cora dataset.

Experiments were performed to compare the effectiveness of the hybrid method using SVM with XMLDup and Dogmatix. To measure the effectiveness of the proposed method two parameters are used: recall and precision.

$$\text{Precision} = tp/(tp+fp) \qquad (4.1)$$

$$\text{Recall} = tp/(tp+fn) \qquad (4.2)$$

where, tp(true positive) is correctly identified duplicates, fp(false positive) is number of non-duplicate nodes which are identified as duplicates and fn(false negative) are the number of duplicates nodes identified as non-duplicates.

The experimental assessment was performed on an intel core i5 CPU at 2.67GHz with a 4Gb RAM. It is fully implemented in Microsoft Visual Studio.

To measure the effectiveness of the proposed method it is compared with XMLDup and Dogmatix.

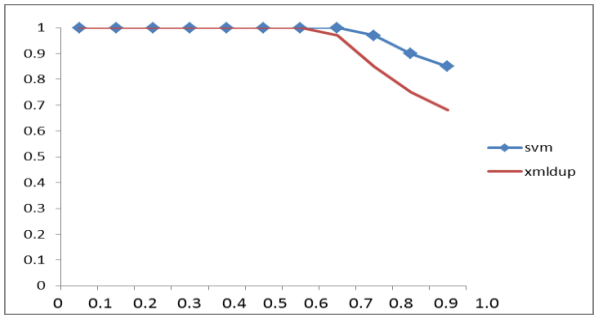

Figure 3 Comparison of results of XMLDup and hybrid method using SVM using Cora dataset.

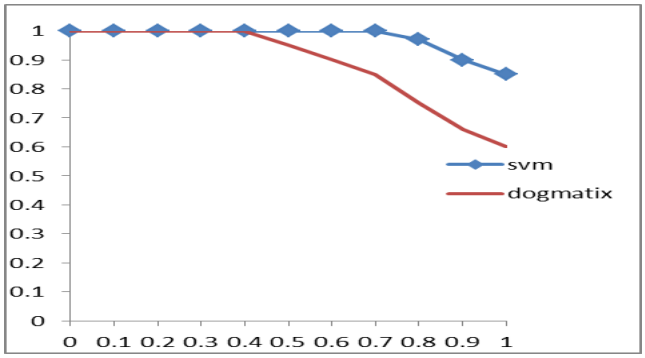

Figure 4 Comparison of results of Dogmatix and hybrid method using SVM using Cora dataset.

Figure 3 and Figure 4 represents the comparison result for XMLDup, Dogmatix and proposed method using SVM representing the recall and precision values for the dataset Cora. The proposed approach shows a better result compared to other algorithms.

Table 1 Recall and precision values for various pruning factor

Pruning factor	SVM		XMLDUP		DOGMATIX	
	r	p	r	p	r	p
0.4	75	99	98	83	92	73
0.5	75	99	98	83	92	73
0.6	75	99	99	87	93	81
0.7	82	100	99	87	93	81
0.8	82	100	99	95	95	83
0.9	82	100	99	95	95	83
1.0	82	100	99	95	95	83

The table which is shown above gives a detailed view about the performance of various algorithms based on different pruning factor. While using SVM it shows a high value of precision compared to other algorithms even though the recall value is low.

5. Conclusion

In this paper, a machine learning algorithm known as SVM is proposed for deriving conditional probabilities for the detection of duplicates and a technique known as binning is used to convert the output of SVM to an accurate posterior probability. Estimating the probability using SVM increases the rate of duplicate detection. SVM not only consider contents but it also takes into account XML objects with different structures. The proposed method achieves an improvement in the value of precision on different structured data.

References

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

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

[3] L. Leitao, P. Calado, and M. Weis, "Structure-Based Inference of XML Similarity for Fuzzy Duplicate Detection," Proc. 16th ACM International Conf. Information and Knowledge Management, pp. 293-302,2007.

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

[5] D. Milano, M. Scannapieco, and T. Catarci, "Structure Aware XML Object Identification," Proc. VLDB Workshop Clean Databases (CleanDB), 2006.

[6] J. Drish "Obtaining calibrated probability estimates from support vector classifiers: project proposal"

[7] B. Zadrozny and C. Elkan. "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers". Proceedings of the Eighteenth International Conference on Machine Learning, 2001.

[8] M. Weis, L. Leitao and P. Calado "Efficient and Effective Duplicate Detection in Hierarchical Data," IEEE Transactions on Knowledge and Data Engineering, Vol. 25, May 2013.

[9] R .Ananthakrishna, S. Chaudhuri, and V. Ganti. "Eliminating fuzzy duplicates in data warehouses," In International Conference on Very Large Databases, Hong Kong, China, 2002.

[10] J. Drish "Obtaining calibrated probability estimates from support vector classifiers"

[11] J.T. Kwok "Moderating the Outputs of Support Vector Machine Classifiers". In IEEE -NN,1995