DERIVING THE PROBABILITY WITH MACHINE LEARNING FOR EFFICIENT DUPLICATE DETECTION IN HIERARCHICAL OBJECTS

SASIKALA.K
Department of Computer Science and Engineering
SNS College of Technology
Coimbatore, India
sasi7675@gmail.com

SATHISH BALAJI.R
Department of Computer Science and Engineering
SNS College of Technology
Coimbatore, India
sathishbalajir@gmail.com

SALMA.A
Department of Computer Science and Engineering
SNS College of Technology
Coimbatore, India
mailsalma92@gmail.com

MAHESWARI.B
Assistant Professor
Department of Computer Science and Engineering
SNS College of Technology
Coimbatore
maheswari.bk@gmail.com

Abstract— XML data is mostly created from distributed and heterogeneous data sources. Most of the Duplicate detection techniques for xml data were consuming more time and memory, because of comparison of each node with all other nodes. The time consumption and memory utilization is reduced by Bayesian Network model. In this method conditional probabilities are used to find out the duplicate elements. However the fixed number of conditional probabilities is not applicable for comparing XML objects with different structures. In this paper machine learning techniques is used to derive the conditional probabilities for new structure entered. SVM machine learning which one of the most aggressive machine learning methods is used for deriving conditional probability. We used a method known as binning technique to convert the outputs of support vector machine classifiers into accurate posterior probabilities in which we can get the better performance and effective duplication result.

Keywords—Duplicate detection, Bayesian networks, XML, SVM, Binning

I. INTRODUCTION

The data sets to be integrated may contain data on the same real-world entities. In order to integrate two or more data sets in a meaningful way, it is necessary to identify representations belonging to the same real-world entity. Therefore, duplicate detection is an important component in an integration process. Duplicate detection is the problem of identifying multiple representations of a same real-world object [5]. With the popularity of XML, there is a growing need for duplicate detection algorithms specifically geared towards the XML data model. Indeed, most algorithms developed for relational data, such as those presented apply to a single relation with sufficient attributes. However, in the case of XML data, we observe that XML elements representing objects have few attributes and instead have related XML elements describing them. We call XML data complex, because we have to consider a schema with complex XML elements that is more complex than a single relation for duplicate detection [5].

A similar approach for XML data also proposed as, the top-down approach, as well as bottom-up

approaches. These methods are, rely on the fact that parent and child elements are in a 1:n relationship, meaning that a parent can have several different children but a child is associated to a unique parent. This is for example the case for <movie> elements nesting <title> elements; because a single title can only belong to a single movie, but a movie can have alternative titles. However, the assumption is not valid for movies nesting actors, because an actor can star in different movies.

II. MACHINE LEARNING

A branch of artificial intelligence, concerns the construction and study of systems that can learn from data. For example, a machine learning system could be trained on email messages to learn to distinguish between spam and non-spam messages. After learning, it can then be used to classify new email messages into spam and non-spam folders. The core of machine learning deals with representation and generalization. Representation of data instances and functions evaluated on these instances are part of all machine learning systems. Generalization is the property that the system will perform well on unseen data instances; the conditions under which this can be guaranteed are a key object of study in the subfield of computational learning theory. There is a wide variety of machine learning tasks and successful applications. Optical character recognition, in which printed characters are recognized automatically based on previous examples, is a classic example of machine learning.

We use the SVM for new structure. Because we are not able to obtain the conditional probabilities for the different structure in the Bayesian Network method. To overcome this problem we are using SVM for the different structure. Here we get the conditional probability as an output of the SVM method for the new structure. In this proposed work, the conditional probability is derived from the SVM [20]. We used a method known as binning to convert the outputs of support vector machine classifiers into accurate posterior probabilities.

SVMs learn a decision boundary between two classes by mapping the training examples onto a higher dimensional space and then determining the optimal separating hyper plane between those spaces [20]. Given a test example *x*, the SVM outputs a score that provides the distance of *x* from the separating hyper plane. The sign of the score indicates to which class *j* example *x* belongs, where $j = \{1,-1\}$. The problem of interest is how to calibrate that score into an accurate class conditional posterior probability, or $P(j|x)$. Our solution is to use a histogram technique known as binning. The binning method proceeds by first ranking the training examples according to their scores, then dividing them into *b* subsets of equal size, called bins. The value of *b* is typically chosen experimentally such that the variance is reduced in the binned probability estimates. Given a test example *x*, it is placed in the bin according to the score produced by the SVM. The corresponding estimated probability $P(j|x)$ is the fraction of training examples that actually belong to the class that has been predicted for the test example.

III.BAYSEIAN NETWORK CONSTRUCTION

A Bayesian network could represent the probabilistic relationships between Xml nodes with their values. The network can be used to compute the probabilities of the presence of similar data in xml elements [6].

Probability of trees U and U′ being duplicates $P(mv_{ij})$,To compute this we need Prior probabilities of leaf nodes and conditional probabilities of inner nodes Represent the likelihood that two values in the XML trees are the same [6] E.g., $P(mv_{ij}$ [year]) □ probability of the two years being the same

Define as

$P(mv_{ij}$[year])=Vi [Year],Vj [Year] If similarity is measured.

Conditional probability 1 (CP1): The probability of the values of the nodes being duplicates, given that each individual pair of values contains duplicates.

Conditional probability 2 (CP2): The probability of the children nodes being duplicates, given that each individual pair of children are duplicates.

Conditional probability 3 (CP3): The probability of two nodes being duplicates given that their values and their children 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 set is duplicates.

IV. NETWORK PRUNING FOR BN

After the calculation of the conditional probability values in the Bayesian network then we remove the duplicate node using the network pruning step. It follows a propose a lossless pruning strategy.

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. As stated before, 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 follows the premise that, before comparing two objects, all the similarities are assumed to be 1 (i.e., the maximum possible score). The idea is to, 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 we assume to be 1. When we 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

V. AUTOMATIC PRUNING FACTOR SELECTION

Attributes in an XML object have different characteristics, they could also have different pruning Factors. A manual fine tuning of pruning factors is complex task. Because all user has more knowledge about the data base. So we compute all pruning factor automatically. We propose a method that automatically determines which pruning factor to use for each attribute, in order to optimize efficiency, while minimizing the loss in effectiveness. We use approximate search using the method of simulated annealing (SA) [19]. SA is an algorithm that is used to determine pruning

factor which is searched from the maximum or minimum value of a function with several independent variables.

VI. XML DUPLICATES DETECTION WITH BN AND AUTOMATIC PRUNING FACTOR

In this the actual xml duplicate detection is achieved by using the BNN with pruning optimization, node ordering heuristics, varying the pruning factor, and automatically selecting the most adequate pruning factors. The experiments are concluded with a discussion of the results**.** The different attributes in an XML object have different characteristics; they could also have different pruning factors. Automatic tuning is very advantage [5]. Because several, pruning factors manually can be a complex task, especially if the user has little knowledge of the database, thus we should be able to compute all pruning factors automatically.

VII. DERIVING THE CONDITIONAL PROBABILITIES OF NEW STRUCTURE USING SVM

For the different structure SVM classifier is used in the proposed work. After classify the objects i.e., determining the output of the SVM, we then transforms that output into the probability. To transform the scores of the SVM classifiers into accurate well-calibrated probabilities, we use a technique known as binning, which is recommended for naive Bayes classifiers in previous systems. The binning method proceeds by first sorting the training examples according to their scores, and then dividing them into *b* equal sized sets, or bins, each having an upper and lower bound. Given a test example *x*, it is placed in a bin according to its score. The corresponding probability $P(j=1|x)$ is the fraction of positive training examples that fall within the bin.

Using all the training examples from the training set sometimes results in over fitting the probability estimates. To solve this problem we use a method, where 70% of the training examples are used to learn the classifier and 30% are used for the binning process [8]. These subsets are stratified, meaning the proportion of positive examples in both of them are fixed. There is no imposed lower or upper bound on SVM scores. Therefore, when using this method it is possible for some scores from the 30% subset to fall below or above the low and high scores, respectively, of the 70% training subset. If this happens the corresponding probability $P(j = 1|x)$ for example *x* is that of the nearest bin to the score of *x*.

VIII. XML DUPLICATES DETECTION WITH BN AND AUTOMATIC PRUNING FACTOR BY USING THE DERIVED CONDITIONAL PROBABILITIES

In this the same xml duplicate detection is achieved with the derived conditional probabilities. The derived conditional probabilities are used to find more efficient duplicate detection for the different structure of data element in the same xml data. In this work, first the SVM is trained with various known structures of xml object and corresponding conditional probabilities [16].

The SVM is calculated the similarity between structures with the condition probabilities and adjust its support vectors boundary value for each structures. Then the adapted boundary values are used for unknown structure in the testing stage, the conditional probability for a given structure is derived by boundary margin of matched structures which are trained already

IX.RESULT

Experiments are performed to compare the effectiveness and efficiency of the tested algorithms. To assess effectiveness, we applied the commonly used precision, recall, and precision measures. Precision measures the percentage of correctly identified duplicates, over the total set of objects determined as duplicates by the system. Recall measures the percentage of duplicates correctly identified by the system, over the total set of duplicate objects. R-precision measures the precision at cut-off R, when R is the number of duplicates in the data set. To assess efficiency, we measured the runtime and number of comparisons of the duplicate detection process.

In this graph shows the performance evaluation between the existing and proposed system. In this graph x axis will be recall and y axis will be precision. In the existing system precision will be decreased according to the increased recall rate. In the proposed system is also precision will be decreased according to recall rate. But in this proposed the rate of precision decreased according to increased recall rate is lower compared to the existing system. We can obtain the conclude that the proposed system has more efficient and effective i.e., maintain higher precision scores until later recall values. Based on this graph the proposed system has more effective than the existing system.

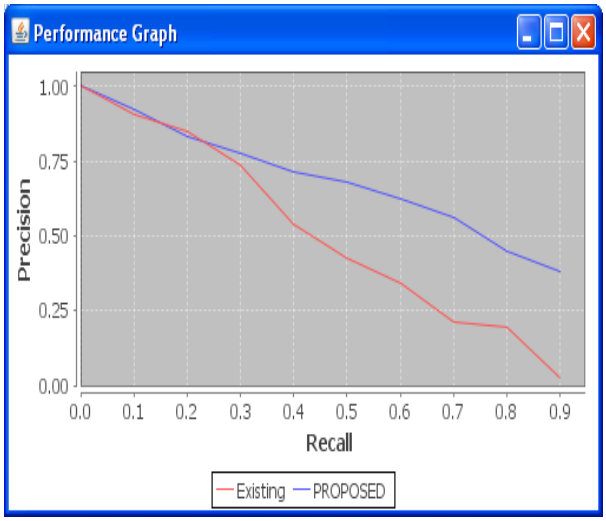

Fig 9.1 Performance Graph

X.CONCLUSION

In this research we presented a novel approach called hierarchical duplicate detection of data with XML Data and XML data object. Both XML and XML data objects are efficient to find duplicate detection of data in structure .Existing network pruning method derive condition probabilities are derived based on the general probabilities values ,it becomes less when compare to machine learning algorithm to derive probability values for XML duplicate detection called XMLDup. SVM machine learning algorithm derives condition probability values automatically instead of general probabilities .It is also performs in two ways: First the probability values are derived automatically by using SVM. Bayesian network pruning was performed to remove duplicate detection of XML data and XML data objects. The Bayesian Network model is composed from the structure of the objects being compared, thus all probabilities are computed considering not only the information the objects contain, but also the way such information is structured. XMLDup requires little user intervention, since the user only needs to provide the attributes to

be considered, their respective default probability parameter, and a similarity threshold. However, the model is also very flexible, allowing the use of different similarity measures and different ways of combining probabilities. To improve the runtime efficiency of XMLDup, a network pruning strategy with SVM is also presented. Furthermore, the second approach can be performed automatically, without needing user intervention. Both strategies produce significant gains in efficiency over the unoptimized version of the algorithm.

XI.FUTURE WORK

Among other tasks we intend to extend the BN model construction algorithm to other types of machine learning and optimization algorithm such as bee, artificial immune system, and BAT algorithm to derive conditional probability values and compare them to existing methods based on the existing 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] F. Naumann and M. Herschel, An Introduction to Duplicate Detection. Morgan and Claypool, 2010.

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

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

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

[6] L. Leita˜o, P. Calado, and M. Weis, "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] 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] D. Milano, M. Scannapieco, and T. Catarci, "Structure Aware XML Object Identification," Proc. VLDB Workshop Clean Databases (CleanDB), 2006.

[9] P. Calado, M. Herschel, and L. Leita˜o, "An Overview of XML Duplicate Detection Algorithms," Soft Computing in XML Data Management, Studies in Fuzziness and Soft Computing, vol. 255, pp. 193-224, 2010.

[10] S. Puhlmann, M. Weis, and F. Naumann, "XML Duplicate Detection Using Sorted Neighborhoods," Proc. Conf. Extending Database Technology (EDBT), pp. 773-791, 2006.

[11] S. Guha, H.V. Jagadish, N. Koudas, D. Srivastava, and T. Yu, "Approximate XML Joins," Proc. ACM SIGMOD Conf. Management of Data, 2002.

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

[13] R.A. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., 1999.

[14] M.A. Herna´ndez and S.J. Stolfo, "The Merge/Purge Problem for Large Databases," Proc. ACM SIGMOD Conf. Management of Data, pp. 127-138, 1995.

[15] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, second ed. Morgan Kaufmann Publishers, 1988.

[16] 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.

[17] E.H. Simpson, "Measurement of Diversity," Nature, vol. 163, p. 688, 1949.

[18] H. Drucker, C.J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support Vector Regression Machines," Proc. Advances in Neural Information Processing Systems (NIPS), vol. 9, pp. 155-161, 1996.

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

[20] T. Joachims, Making Large-Scale Support Vector Machine Learning Practical, pp. 169-184. MIT Press, 1999.

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

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

[23] W. Huang, Y. Nakamori, and S. Wang, "Forecasting stock market movement direction with support vector machine," Computers & Operations Research, vol. 32, 2005, pp. 2513-2522.

