Record Matching : Improving Performance in Classification

Cyju Elizabeth Varghese
Dept. of Computer Science and Engineering
Karunya University
Coimbatore, India

G. Naveen Sundar
Dept. of Computer Science and Engineering
Karunya University
Coimbatore, India

Abstract— **Duplication detection identifies the records that represent the same real-world entity. This is a vital process in data integration. Record matching refers to the task of finding entries that refer to the same entity in two or more files. Performing record matching solves the duplication detection problems; hence the needs for identifying the suitable record matching technique follow. Supervised methods are the current techniques used for duplication detection. This requires the user to provide training data. These methods are not applicable for the Web database scenario, where the records to match are query results dynamically generated on-the-fly. To address the problem of record matching in the Web database scenario, we present a Fast Duplication Detection, FDD, which, for a given query, can effectively identify duplicates from the query result records of multiple Web databases. Starting from the non-duplicate set, we use two, a dynamic classification classifier and an SVM classifier, to iteratively identify duplicates in the query results from multiple Web databases. Performing clustering before giving vectors to classify should produce a better result. Moreover a nonlinear SVM produce a better result in case of noise document which improves overall performance of the system. Experimental results show that FDD performs better for web database scenario.**

Keywords- Record Matching; Duplication Detection; Record matching; SVM; Unsupervised

I. INTRODUCTION

Data mining tasks usually work on large data where the sources of data are from different sources. Despite the fact that records are not bit wise identical, they are strikingly similar. Potential duplicates posses minute difference and so are not regarded are exact duplicates. Typographical errors, versioned, mirrored, or plagiarized documents, multiple representations of the same physical objects are some of the chief causes for the occurrence of duplicate records. Such near duplicates contain similar content and vary only in minimal areas of the document like the advertisements, counters and timestamps. Web searches consider these differences as inappropriate. One problem that degrades the data results is the duplicated data among records from different sources.

This paper then proposes an efficient approach to detect duplicates in a web database scenario. The characteristics of relational data are analyzed from the perspective of duplicate detection. We define constraint rules that capture these characteristics. Since in a typical database the vast majority of randomly selected record pairs are non-duplicates, it is possible to populate the training set with negative examples based on such pairs, while filtering out likely pairs of duplicate records using similarity metrics such as vector-space cosine similarity. Rest of the paper is organized as follows. In the Web database scenario, the records to match are highly query-dependent, since they are results of the query.

Moreover, they are only a partial and biased portion of all the data in the source Web databases. Moreover, hand-coding or offline-learning approaches does not aid for two reasons [7]. First, the full data set is not available beforehand, and therefore, good representative data for training are hard to obtain. Second, and most importantly, even if good representative data are found and labeled for learning, the rules learned on the representatives of a full data set may not work well on a partial and biased part of that data set. Section II contains a generalized summary of various existing record matching techniques and brief description on what various context different techniques that have been taken for study. Section III gives a proposed architecture emphasizing its advantage over the existing approach.

II. BRIEF DISCUSSAION OF LITREATURE

In supervised learning, a number of labeled examples are usually required for training an initial predictor which is in turn used for exploiting the unlabeled examples. However, in many real-world applications there may exist very few labeled training examples, which makes the predictor difficult to generate, and therefore these supervised learning methods cannot be applied. In Unsupervised learning the users are not required to provide training data.

Thus for real time applications for example in this case to solve the problem of record matching from multiple web databases unsupervised learning methods are applicable. There are multiple algorithms for record matching. Some of them are supervised models which are based upon training data, DEPLHI technique[4] to eliminate fuzzy duplicates, methods using classifiers, approach using a weak classifier, Mapping – Convergence method [2]. The other technologies involved in record matching are unsupervised method where training data is not required. Negative training set, Positive training set, negative and positive training set can be provided as training data. In many machine learning we can understand that positive examples are difficult to collect while negative examples are abundant. This can be used in a web database scenario where the matching results depends could be query dependant. Similarity calculation among records and the proper selection of similarity function is an prominent part. The algorithms are run on various datasets and the performance is analyzed. UDD [1] approach performs record matching suitable for a web-database scenario. Though it has the advantage of utilizing dynamic allocation of weights to fields, the size of query result records, determines the time consumed by SVM classifier.

III. PROPOSED ARCHITECTURE

This paper proposes a method an efficient approach to solve duplication detection problem in web-database scenario where the records to match are query dependant and can dynamically change. In the existing record matching techniques of Chrsiten's method [2], PEBL [3] and clustering methods, it exhibit allocating static weights to fields of records. Especially in an online environment, the data type and nature of the query result is unpredictable hence static allocation of weight is inappropriate. Therefore an Unsupervised method is appropriate for record matching to solve duplication detection. Dynamic allocation of weights to different fields in a record provides efficient method of record matching [5]. Even though UDD [1] performs online duplication detection in an unsupervised manner, it considers all possible pair of records as potential duplicates, which is not necessary in reality.

Moreover duplication detection has to reduce record comparisons. For this many methods are already available like record clustering, bigram indexing [12] etc. FDD uses clustering in order to reduce record comparison. Thus, obtaining training data D and N' is less consuming which in turn increases the performance. The Support Vector Machine (SVM), this classifier using D and N' as train data further classifies the clustered potential duplicate vector set (P).

- Similarity Calculation is performed
- K-Means Clustering is performed on the obtained values and clustered
- Obtain Potential Duplicate Vector (P) and Non-duplicate vector (N)
- Use dynamic weight allocation algorithm to set the weights to each field and to obtain duplicate vector from P and N
- Train SVM
- Classify the potential duplicate vector to obtain actual duplicates
- Iteratively perform both the algorithms until Non-duplicate vector do not contain any further duplicates.

A. Similarity Calculation

Any similarity functions can be employed in FDD approach [6]. String transformations are adapted here [1]. Initially weight vector is initialized that sum of the weight components do not exceed 1. The Similarity vector Sim (v1,v2) is found among records. A similarity threshold value is set to identify initially a set potential duplicate vector and nonduplicate vector.

B. K-Means Clustering

Clustering is the unsupervised classification of patterns (or data items) into groups (or clusters). This method incorporates clustering before obtaining initial potential duplicate and non-duplicate vector. This has the advantage that classification of both vectors maybe more streamlined. The number of cluster to be generated is to be provided. Here we get two groups, mainly potential duplicate vector (P) and Non-duplicate vector (N). Moreover clustering can be applied to the potential duplicate vector set (P). This has the advantage of better

generation of training data and earlier classification of the potential duplicate vector by SVM. Thus the overall accuracy of the algorithm and less time consumed to detect duplicates makes it a better approach.

C. Vector Generation

After clustering, it generates Potential duplicate vector (P) and Non-duplicate vector (N) which is clustered more proficiently. Initial vector of weights are allocated for the fields for similarity calculation.

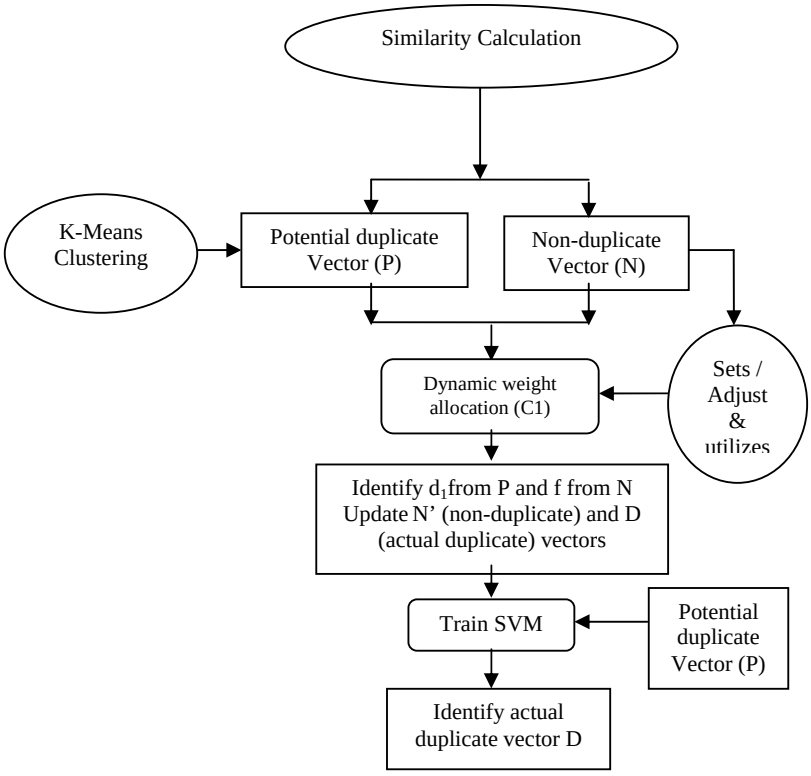

Figure 1. FDD (Faster Duplication Detection) algorithm.

D. Dynamic Weight Allocation [1]

The Dynamic allocation of weights to different fields in each record is performed by the Dynamic Allocation Algorithm. It considers both Duplicate Vector D and Non-duplicate vector N. The intuition for the weight assignment includes: [1]

1. Duplicate intuition: The similarity between two duplicate records should be close to 1. For a duplicate vector V12 that is formed by a pair of duplicate records r1 and r2, we need to assign large weights to the components with large similarity values and small weights to the components with small similarity values.
2. Non-duplicate intuition: The similarity for two non-duplicate records should be close to 0. Hence, for a non-duplicate vector V12 that is formed by a pair of non-duplicate records r1 and r2, we need to assign small weights to the components with large similarity values and large weights to the components with small similarity values.

E. Support Vector Machine

Support Vector Machine classifier is a useful technique for data classification. The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes. In this method classification is performed on Potential duplicate vector (P) and new actual duplicates are identified. The Classifier is trained using actual duplicate vector (D) and Non-duplicate vector as depicted in the below Fig 2. The performance of classification can be improved by using a better SVM. Utilizing a linear SVM on textual documents can lead to false positives. Instead of generating a general model to predict the test data, here the training data can be clustered and for each cluster a model could be generated. Therefore we go

for a nonlinear SVM where it will promote better classification of text documents giving better overall performance [15].

F. Identify Actual Duplicates.

Deduct the actual duplicates found from P and from N and update the new P and N. Iteratively perform C1 classifier and C2 classifier until there are no duplicates in the Non- duplicates.

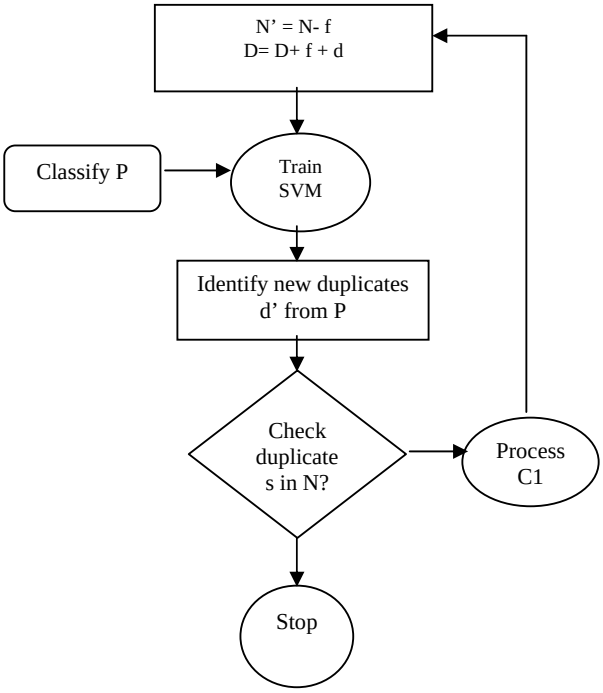

Figure 2. SVM process

IV. EXPERIMENTAL ANALYSIS

The proposed algorithm can be applied to various dataset. This is best applicable for web-database like Movie, Book and Hotel Booking etc. Moreover it can be applied on dataset Cora data set. Accuracy comparisons between different systems are also performed using variety of different measures used to evaluate individual approaches, such as:

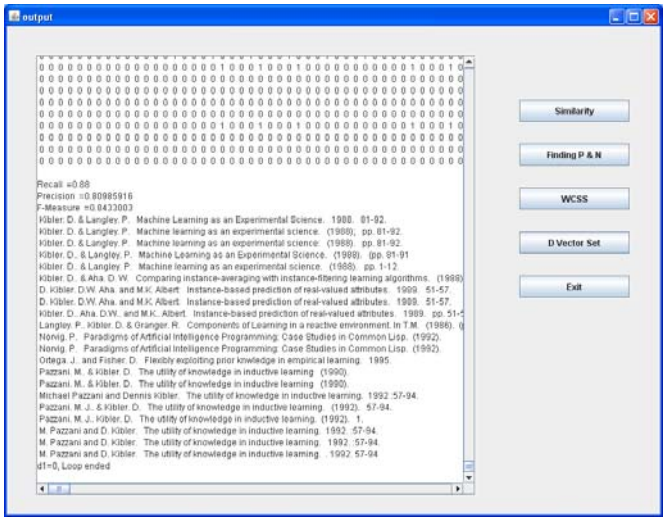

Figure 3. D Vector Set

•Maximum F-measure, which is the harmonic mean between pair-wise precision and recall.
•Pair-wise precision for the optimal number of pairs.
•Percentage of the correct equivalence classes for which an error exists in the grouping.
•Proportions of true matching pairs at fixed error levels

Figure 3 displays the actual duplicate vector set D, found by iteratively performing the two classifiers duplication detection and SVM classifier.

A. Evaluation Metric

The Overall performance can be evaluated using recall and precision. The large number of non-duplicates usually controls accuracy measure and yields results that are too optimistic.

$$\text{Pr}\,ecision = \frac{\#\,\text{of Correctly Identified Duplicate Pairs}}{\#\,\text{of All Identified Duplicate Pairs}} \quad (1)$$

$$\text{Re}\,call = \frac{\#\,\text{of Correctly Identified Duplicate Pairs}}{\#\,\text{of Actaul Duplicate Pairs}} \quad (2)$$

But due to unfair distribution of matched and non-matches in the weight vector set [13], we also use the F-measure, which is the harmonic mean of precision and recall.

$$\text{F-measure} = \frac{2 * \text{precision. Recall}}{\text{Precision + recall}} \quad (3)$$

Although all of these quantities characterize accuracy of duplicate detection systems, they avoid problem of selecting the threshold Tsim that separates duplicates from non-duplicates. This assumes, Tsim has been chosen by the user, or select a certain value implicitly, as maximum F- measure does. One problem with this approach is that the relative cost of false positives (non-duplicate pairs selected as duplicates) and false negatives (unidentified duplicate pairs) may vary, making the optimal value of Tsim situation specific.

TABLE I. PERFORMANCE OF FDD ON THE WEB-DATABASE SETS

Classification	***Precision***	***Recall***	***F-measure***	***Avg. Execution***
Book	0.954	0.925	0.939	0.85
Hotel	0.961	0.952	0.955	0.74
Movie	0.932	0.928	0.930	0.21

B. Cora Data Set

Although Cora is a noisy data set, our algorithm still performs well over it. Algorithm was tested on Cora dataset. FDD has a precision of 0.899, recall of 0.950, and F-measure of 0.933 over the Cora data set.
Table 1 shows the precision, recall, F-measure value, and actual execution time of the algorithm on the Web database data sets when the similarity threshold Tsim = 0.85. It can be seen that FDD can efficiently identify duplicates among records from multiple data sources with good precision and recall, on average.

V. CONCLUSION

In this paper, we propose a modified architecture to improve the performance of the record matching to solve duplication detection. Duplicate detection is an important step in data integration and most state-of-the-art methods are based on offline learning techniques, which require training data. The general in web database scenario is the records to match are greatly query-dependent; a pre-trained approach is not applicable as the set of records in each query's results is a biased subset of the full data set.
To overcome this problem, a better approach for an existing unsupervised, online approach, for detecting duplicates over the query results of multiple Web databases has been discussed. Two classifiers, dynamic classification classifier and SVM, are used cooperatively in the convergence step of record matching to identify the duplicate pairs from all potential duplicate pairs iteratively. The Accuracy of Support Vector Machine is

improved by using a nonlinear SVM to decrease the effect of noise documents. Prior to this, we have integrated clustering where a streamlined set of duplicate and non-duplicate vectors are generated. This aids in better performance of the FDD approach. Experimental results show that our approach is comparable to previous work UDD that identifies duplicates from the query results of multiple Web databases.

ACKNOWLEDGMENT

First and foremost, I praise and thank ALMIGHTY GOD whose blessings have bestowed in me the will power and confidence to carry out my work. I feel it a pleasure to be indebted to my guide, Mr. G. Naveen Sundar, M.Tech (Ph.D), Assistant Professor, Department of Computer Science and Engineering for his support and encouragement.

REFERENCES

[1] Weifeng Su, Jiying Wang, and Frederick H. Lochovsky, "Record Matching over Query Results from Multiple Web Databases," IEEE Transaction Knhe opowledge and Data Engineering, April 2010 (vol. 22 no. 4) pp. 578-589.

[2] P. Christen, "Automatic Record Linkage Using Seeded Nearest Neighbour and Support Vector Machine Classification," Proc.ACM SIGKDD, pp. 151-159, 2008

[3] H. Yu, J. Han, and C.C. Chang, "PEBL: Web Page Classification without Negative Examples," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 1, pp. 70-81, Jan. 2004.

[4] R. Ananthakrishna, S. Chaudhuri, and V. Ganti, "Eliminating Fuzzy Duplicates in Data Warehouses," Proc. 28th International Confernce Very Large Data Bases, pp. 586-597, 2002

[5] William Cohen, Pradeep Ravikumar, and Stephen Fienberg, "Adaptive Name matching in Information Intergration" vol. 18 no. 5 , pp. 16-23, 2003.

[6] N. Koudas, S. Sarawagi, and D. Srivastava, "Record Linkage: Similarity Measures and Algorithms (Tutorial)," Proc. ACM SIGMOD, pp. 802-803, 2006

[7] W. Su, J. Wang, and F.H. Lochovsky, "Holistic Schema Matching for Web Query Interfaces," Proc. 10th Int'l. Conf. Extending Database Technology, pp. 77-94, 2006.

[8] R. Baxter, P. Christen, and T. Churches, "A Comparison of Fast Blocking Methods for Record Linkage," Proc. KDD Workshop Data Cleaning, Record Linkage, and Object Consolidation, pp. 25-27, 2003.

[9] Bin He, Kevin Chen-Chuan Chang, "Automatic Complex Schema Matching across Web Query Interfaces: A Correlation Mining Approach" ACM Transactions on Database Systems, Vol. 31, No. 1, March 2006, Pages 1–45.

[10] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani, "Robust and, T Efficient Fuzzy Match for Online Data Cleaning," Proceedings ACM SIGMOD, pp. 313-324, 2003.

[11] P. Christen. Churches, and M. Hegland, "Febrl—A Parallel Open Source Data Linkage System," Advances in Knowledge Discovery and Data Mining, pp. 638-647, Springer, 2004.

[12] P. Christen and K. Goiser, "Quality and Complexity Measures for Data Linkage and Deduplication," Quality Measures in Data Mining, F. Guillet and H. Hamilton, eds., vol. 43, pp. 127-151, Springer, 2007.

[13] Mikhail Bilenko and Raymond J. Mooney, "On Evaluation and TrainingSet Construction for Duplicate Detection," Proceedings of the KDD-2003 Workshop on Data Cleaning, Record Linkage, and Object Consolidation,pp. 7-12, Washington DC, August, 2003.

[14] W.W. Cohen and J. Richman, "Learning to Match and Cluster Large High-Dimensional Datasets for Data Integration," Proc. ACM SIGKDD, pp. 475-480, 2002.

[15] Haibin Cheng, Pang-Ning Tan, Member, IEEE, and Rong Jin, "Efficient Algorithm for Localized Support Vector Machine," IEEE Transaction Knowledge and Data Eng., vol. 22, no. 4, April 2010.

AUTHORS PROFILE

Cyju Elizabeth Varghese received the B.E degree in Computer Science and Engineering from CSI Institute of Technology, Thovalai, India, in 2001. Currently she is doing M.Tech in Computer Science and Engineering in Karunya University, Coimbatore. Her research interests include Web Mining Data Mining and areas related to Database.

Naveen Sundar received the ME degree in computer science from Karunya University, Coimbatore, India, in 2006. Since 2009, He has been a PhD candidate in computer science at Anna University. His research interests include multimedia database, soft computing and fuzzy, and web image retrieval.