International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161
©2014 INPRESSCO®, All Rights Reserved
Available at http://inpressco.com/category/ijcet

Research Article

An Approach towards Record Linkage using Genetic Algorithm along with Hash Algorithm

J. R. Waykole[A*] and S. M. Shinde[B]

[A]Computer Engineering Department, Pune University, India

Accepted 15 June 2014, Available online 28 June 2014, **Vol.4, No.3 (June 2014)**

Abstract

Several systems that depends on the integrity of the data in order to offer high quality services, such as digital libraries and e-commerce brokers, may be affected due to the existence of duplicates in their warehouse. Due to this, more time is required to retrieve high quality data. Here deduplication or record linkage is computed by using hash algorithm i.e., MD5 and SHA-1 algorithm for finding similarity to detect duplicate records and eliminate them using evolutionary i.e., genetic algorithm. This approach removes the duplicate dataset samples in the system.

Keywords: *Cosine similarity, Dataset, genetic algorithm, MD5, SHA-1 and string distance.*

1. Introduction

As we know that, today increase in the volume of information created problem for duplicate records as we are collecting data from heterogeneous sources. So, to find duplicate records which are collected from several different sources, is necessary task. Due to this, more resources and time are required to find relevant data from repositories or warehouses. That's why we use deduplication technique to improve the data quality.

In a data repository or warehouse, a record that refers to the real world object is referred as duplicate records. And that duplicate record is also called as 'grubby or dirty data'. So many problems are occurred due to the presence of this dirty data in warehouse as follows:

1) *Performance ruin* — As we gathered the data from heterogeneous sources, it demands for more processing and more time is required to answer simple queries.

2) *Eminence failure* — Due to the presence of replicas or duplicates in repositories or warehouses and other inconsistencies, it leads to distortions in reports and misleading conclusions based on the existing data.

3) *Increased expenditure* — As we gathered more and more data from various sources, due to this additional volume of useless data, expensive investments are required on more storage media and extra computational processing power is required to keep the response time levels acceptable.

A major cause is the presence of duplicates or replicas in these repositories or warehouse is the *incorporation* of distinct data from heterogeneous sources. The problem of detecting and removing these duplicate records from a repository or warehouse is known as record deduplication) N. Koudas *et al*, S. Sarawagi *et al*, and D. Srivastava *et al* 2006). It is also referred as data cleaning (S. Chaudhuri *et al*, K. Ganjam *et al*, V. Ganti *et al*, and R. Motwani *et al* 2003), record linkage and record matching (V.S. Verykios *et al*, G.V. Moustakides *et al*, and M.G. Elfeky *et al* 2003). Also it is referred as merge-purge (M.A. Hernandez *et al* and S.J. Stolfo *et al* 1998) and instance identification (Y.R. Wang *et al* and S.E. Madnick *et al* 1989). In AI community, the same problem is described as database hardening (W.W. Cohen *et al*, H. Kautz *et al*, and D. McAllester *et al* 2000) and name matching (M. Bilenko *et al*, R.J. Mooney *et al*, W.W. Cohen *et al*, P. Ravikumar *et al*, and S.E. Fienberg *et al* 2003).

The rest of this paper is organized as follows. In Section 2, we discuss the literature survey i.e., different techniques had been applied for record linkage. In Section 3, we discussed the data preparation i.e., different similarity functions. In Section 4, we discussed how genetic algorithm alongwith similarity functions be applied to record deduplication problem. In Section 5, we describe the results of genetic algorithm for Deduplication problem. Finally, in Section 6, we present our conclusion to this technique.

2. Literature Survey

Record linkage is a research growing topic in databases as many duplicates or replicas are exists in repositories or warehouses. This problem can be solved by combining information available from repositories and identify whether a pair of record refers to the real world entity. As more strategies for extracting diverse pieces of data from various interpretations become available, many works have proposed new diverse approaches to combine and use them (A.K. Elmagarmid *et al*, P.G. Ipeirotis *et al*, and V.S.

*Corresponding author **J. R. Waykole** is a PG student and **S. M. Shinde** is working as Associate Professor

Verykios *et al* 2007) and classify these approaches into the following two categories:
1) Ad-Hoc or Domain Knowledge Approach—this approach usually relies on specific domain knowledge or specific string distance metrics. Those techniques where there is a use of declarative languages (A.K. Elmagarmid *et al*, P.G. Ipeirotis *et al*, and V.S. Verykios*et al* 2007) can also be classified under this approach.
2) Training-based Approach—this approach relies on some sort of training i.e., supervised or semi-supervised in order to identify the replicas. It includes probabilistic and machine learning approach.

A. Domain Knowledge Approaches

The idea of combining observations or facts or evidences to identify replicas has forced the researchers to look for methods that could get benefit from domain specific information found in actual gathered data as well as for different techniques based on general similarity metrics (A.K. Elmagarmid *et al*, P.G. Ipeirotis *et al*, and V.S. Verykios*et al* 2007). Here it uses matching algorithm. If we give a record from a file or repository or warehouse, then it looks for another record in a reference file. If it matches with the first record according to a given similarity function as threshold and if it returns more than one record that matches with that, then at that time the user is required to choose one record from that which is very close to the first one.

Records matching on high-weight tokens (strings) are more similar than those matching on low-weight tokens. The weights are calculated by the well-known IDF weighting method. In (V.R. Borkar *et al*, K. Deshmukh *et al*, and S. Sarawagi *et al* 2001), the authors use the vector space model for computing similarity among fields from different sources and evaluate four distinct strategies to assigning weights and combining the similarity scores of each field. As a result of their experiment, they found that using evidence extracted from individual attributes improves the results of the replica identification task. Here similarity is calculated by using distance metrics.

Limitations of Domain Knowledge Approach

- Here it is necessary to define the matching threshold (A.K. Elmagarmid *et al*, P.G. Ipeirotis *et al*, and V.S. Verykios*et al* 2007).
- Hence, the major advantage of ad-hoc method is nullified and it can operate without training data.

B. Probabilistic approach

Newcombe *et al*. were the first ones to address the record deduplication problem as a Bayesian inference problem i.e., a probabilistic problem and proposed the first approach to automatically handle duplicates. However, their approach was considered empirical since it lacks statistical ground as shown in paper (A.K. Elmagarmid *et al*, P.G. Ipeirotis *et al*, and V.S. Verykios*et al* 2007).

After Newcombe *et al*.'s work, Fellegi and Sunter proposed a more elaborated statistical approach to deal with this problem(I.P. Fellegi *et al* and A.B. Sunter *et al* 1969). It is implemented with Bayes's rule and Naive based classification. This method is usually works with two boundaries as follows:

1. *Positive identification boundary*— it means if the similarity value lies above this boundary, then the records are considered as duplicates.
2. *Negative identification boundary*— it means if the similarity value lies below this boundary, then the records are not considered as duplicates.

For the situation in which similarity values lies between the two boundaries, and according to him, the records are classified as possible matches or considered as there exists replicas or duplicates.

Limitations of Probabilistic Approach

- It relies on the two boundary values definition which is used to classify a pair of records as being duplicates or replicas or not.
- Identification errors are increased due to bad boundaries.
- In this case, a human judgment is necessary to identify the boundary values (S. Chaudhuri *et al*, K. Ganjam *et al*, V. Ganti *et al*, and R. Motwani *et al* 2003).

C. Machine Learning approach

By using machine learning techniques, we have derived record level similarity functions that combine field-level similarity functions, including the weights of records is mentioned in paper as (M. Bilenko *et al*, R. Mooney *et al*, W. Cohen *et al*, P. Ravikumar *et al*, and S. Fienberg *et al* 2003, W. Banzhaf *et al*, P. Nordin *et al*, R.E. Keller *et al*, and F.D. Francone *et al* 1998). It uses a small portion of the available data for training i.e., nothing but called as a test data. The main idea behind this approach is that, the similarity is calculated by using probability between these attributes, so higher the probability, the bigger the similarity between these attributes.

The adaptive approach is presented in (C. Sutton *et al*, K. Rohanimanesh *et al*, and A. McCallum *et al* 2004). This approach is applied to both clustering and pair-wise matching. During the learning phase, the mapping rule and the transformation weights are defined and combining them and executed using decision trees. The process involves two steps as follows:
1) First, it generates a mapping rule to find similarity between attributes.
2) Then, a mapping rule learner determines the duplicates and executed by a decision tree.

Limitations of Machine Learning Approach

- It requires large computation.
- It also requires high memory storage for mapping rules.
- This technique is data oriented i.e., they model the relationships between attributes of the training data set (C. Sutton *et al*, K. Rohanimanesh *et al*, and A. McCallum *et al* 2004).

D. Genetic Algorithm

Charles Darwin introduced that evolutionary computation is an area of computer science, which is inspired by the principles of natural selection. Genetic Programming is one of the evolutionary programming techniques which have the properties of natural selection or natural evolution. It is having mainly three operations such as selection, crossover and mutation (J.R. Koza *et al* 1992). All the operation has been incorporated in the algorithm. At each point during the search space we preserve a generation of individuals. In GP, each individual represents the possible solution for the problem. These individuals are represented by means of complex data structures such as trees, or graphs. After the initial population has been created, the actual evolutionary process starts.

The algorithm iteratively refines an initial population of potential solutions until a solution is found. An initial population is made up of number of solutions or problems. It not only creates new solutions or problems but also allows new combination of features (S. N. Sivanandam *et al* and S. N. Deepak *et al* 2008) into offspring.

Features of genetic programming

1) Genetic algorithm works with multi-objective problems.
2) Genetic algorithm has good performance on searching over very large search spaces, where the optimal solution in many cases is not known, but it can provide near-optimal solution.
3) Genetic algorithm can be applied to symbolic regression problems.
4) Genetic algorithm has been used for optimization problems.
5) Genetic algorithm is distinguish from other evolutionary techniques in the way that it represents the concepts and the interpretation of a problem as a computer program and even the data are viewed and manipulated .
6) Able to discover the independent variables and their relationships with each other and with any dependent variable.

3. Data Preparation

A record linking system contains several components, which includes data pre-processing, record pair comparison, record pair classification, and result evaluation. Among them, record pair classification has attracted most attention. In this task, the similarities of record pairs determine whether the pairs are matched or non-matched. Duplicate record detection is the process of identifying different or multiple records that refer to one unique real world entity or object (I. Bhattacharya *et al* and L. Getoor *et al* 2004).

Field Matching Techniques

One of the most common sources of mismatches in database entries is the variation of string data in which they are represented. Therefore, detection of duplicate is depends on string comparison techniques. Multiple methods have been developed for this task. In this, we describe techniques that have been applied for matching fields with string data in the duplicate record detection context.

String- based similarity

The string-based similarity method is designed to handle field matching in databases. In this section, we consider an example of book store, were divided into multiple attributes (author names, year, title, venue, and pages and other info) by an information extraction system. The string distance similarity function was applied to four out of five attributes. Only the attribute year was not used. This happened because the cosine similarity function, when applied to dates, is not able to properly measure the distance between them.

Cosine-based similarity

The cosine similarity between the two records name field Record 1 and Record 2 are calculated as follows:

1) The dimension of both the strings are obtained by taking the union of two string elements.
2) Then calculate the frequency of occurrence vectors of the two elements.
3) After that, obtain the dot product and magnitude of both strings.

For example, the dimension of both strings are obtained by taking the union of two string elements in the record 1 and record 2 as (word1 , word2, …….word N) and then the frequency of occurrence vectors of the two elements are calculated i.e., record 1 = (<vector value1>, <vector value2>,……<>) and record 2= (<vector value1>, <vector value2>,……<>).

Hashing-based similarity

Hash-based methods of redundancy elimination process each piece of data using a hash algorithm, such as SHA-1 or MD5. This method generates a unique number for each piece of data which is compared to an index of other existing hash numbers. If that hash number already exists on the index, the data need not be stored again. Otherwise, the new hash number is added to the index and the data stored. By using MD5 and SHA-1 algorithm we can eliminate duplicates from records or repositories. Hash collisions occur when two different record produces the same hash. The chances of this are very slim indeed, but SHA-1 is considered the more secure of the two algorithms.

4. Modelling the Record Deduplication with GA

A. Architecture Diagram

We proposed our work in the form of architectural diagram as shown in figure 1 below.

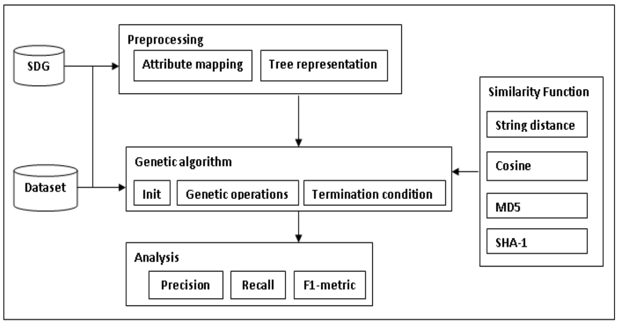

Figure 4.1 Architectural Diagram

B. Proposed Algorithm

1. First we generate synthetic data or take dataset as input to our system and then we convert input to an initial population for genetic algorithm.
2. Then in preprocessing module user can select similarity functions for different attributes of dataset according to their choice.
3. After selecting then it can be represented that problem in the form of data structure as tree.
4. From that tree representation, we compute the value of root as a de-duplication measure value i.e., nothing but a numeric value and this would be applied to genetic algorithm.
5. Then we evaluate all individuals and computed fitness value and compare with de-duplication measure value.
6. After this we apply roulette wheel selection criteria to choose m individuals to reproduce the next generation with the best parents.
7. After this apply genetic operations on that selected individuals like crossover and mutation to reproduce next generation.
8. Then replace the existing population with this new generated individuals and go back to step 5 until our termination condition is reached i.e., until all individuals are completed from dataset or for all attributes till we find the best similarity function.
9. Present the best individuals in the population as the output of the evolutionary process and considered it as duplicates or replicas and eliminate them.

The population size is one of the most important parameter that plays a significant role in the performance of the genetic algorithms. Here we use the tree representation for finding the de-duplication measure value as shown in figure 4.2. This can be represented as decision tree (Akshara k. *et al*, Soorya P. *et al* 2012) of the form as follows.

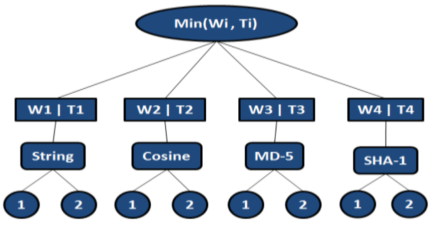

Figure 4.2 Tree Representation

When using this tree representation for GA then, a set of terminals represents the input as name of attribute along with function name as defined in (J.R. Koza *et al* 1992). Terminals are also called as leaf nodes or leaves of tree. In paper (Moises G. de Carvalho, Alberto H.F. Laender, Marcos Andre Goncalves, and Altigran S. da Silva 2012), the similarity function i.e., string distance similarity and cosine similarity is applied separately to each attribute and then compute the similarity as fitness value for the execution of genetic algorithm. In (Moises G. de Carvalho, Alberto H.F. Laender, Marcos Andre Goncalves, and Altigran S. da Silva 2012), each piece of evidence i.e., E is a pair of <attribute_name, similarity function>. Here we added two more similarity functions based on hash algorithm as SHA-1 and MD5 algorithm to detect duplicate records. For example, we consider the synthetic dataset with five attributes as (e.g. name, zipcode, age, sex and disease) using a similarity function (e.g string_sim, cosine_sim, SHA1_sim and MD5_sim). These pair of attributes with similarity function represents leaves of trees. The internal nodes represent operations that are applied to the leaves and modeled with arithmetic operators (e.g., +, -, *, /, exp or min or max operations)(Yinjin Fu, Hong Jiang, Nong Xiao, Lei Tian, Fang Liu, and Lei Xu 2013).

5. Results & Discussions

We generate three synthetic dataset i.e., 250, 650 and 1200 records. Each containing 100, 300 and 400 duplicate records respectively. Then we assign some parameters with values for representation of genetic algorithm as shown in table 5.1.

Table 5.1 Parameters for GA

Sr. No.	Parameter Name	Value
1.	Chromosome Size	20 chars
2.	Population Size	N chromosomes
3.	Crossover Probability	0.75
4.	Number of Generations	N
5.	Mutation Probability	0.02
6.	Gene Values	0 or 1
7.	Fitness Function Value	Between 0 and 1

Table 5.2 Results Using Both Methods

Method	Synthetic Dataset	Processing Time Required	Precision	Recall	F1-score
Automatically Selected Best Similarity Function	250 Records	0.0.3	0.0	1.0	0
	650 Records	0.0.26	0.230	0.769	0.35
	1200Records	0.1.14	0.167	0.833	0.27
Randomly Selected Similarity Function	250 Records	0.0.4	0.176	0.824	0.29
	650 Records	0.0.25	0.158	0.841	0.26
	1200Records	0.1.16	0.109	0.890	0.19

Here, we provide the results for this problem using finding the best suitable function from all(i.e., string, cosine, MD5 and SHA-1 similarity) and rendoomly selected function to all attributes of dataset. Here, we take synthetic data of 250, 650 and 1200 records and total attributes are fifteen. We given the table below as table 5.1 for the both methods with their precision, recall, f1-score and required time for processing and its corresponding graph shown in figure 5.1. And figure 5.2.

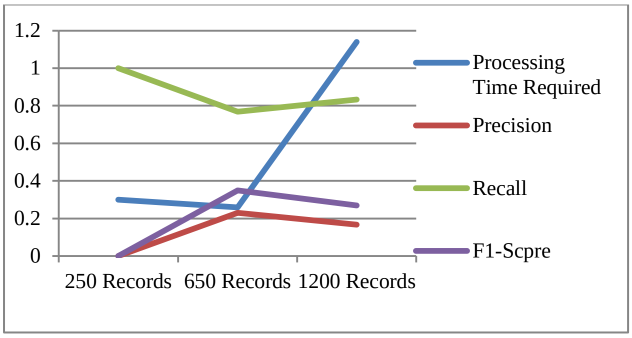

Figure 5.1 Graph for Automatically Selected Best Function

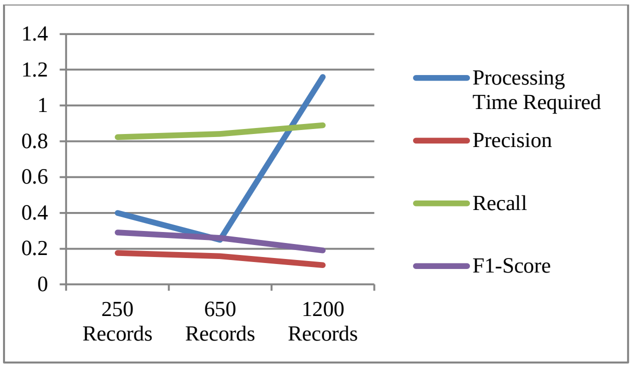

Figure 5.2 Graph for Randomly Selected Function

Conclusion

As there is duplicate information, so more space is required to store data. In order to identify and handle these replicas is an important task to guarantee the quality of information. So by detecting and removing such duplicates from repositories or warehouse is an important task. In this, we presented a GA approach with MD5 and SHA-1 for detecting duplicates and eliminate it. Also we proposed the our work as; the system can automatically select the best suitable similarity function for each attribute using tree representation technique in order to detect duplicates. Also we enhance our system that user can automatically select the similarity function whichever he/she wants to assign for particular attribute in order to achieve accuracy. Here we conclude that this record linkage or record deduplication problem belongs to NP-complete class as we get the best solution from all possible similarity functions, i.e., whichever is recommended. Also we can apply this technique for storing large data on cloud.

References

Moises G. de Carvalho, Alberto H.F. Laender, Marcos Andre Goncalves, and Altigran S. da Silva, A Genetic Programming Approach to Record Deduplication in IEEE Transactions on Knowledge and Data Engineering, Vol. 24, NO.3, March 2012.

Yinjin Fu, Hong Jiang, Nong Xiao, Lei Tian, Fang Liu, and Lei Xu, Application-Aware Local-Global Source Deduplication for Cloud Backup Services of Personal Storage in IEEE Transactions on Parallel and Distributed Systems, 2013.

N. Koudas, S. Sarawagi, and D. Srivastava, Record Linkage: Similarity Measures and Algorithms in Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 802-803, 2006.

S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani, Robust and Efficient Fuzzy Match for Online Data Cleaning in Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 313-324, 2003.

V.S. Verykios, G.V. Moustakides, and M.G. Elfeky, A Bayesian Decision Model for Cost Optimal Record Matchin, in The Very Large Databases J., vol. 12, no. 1, pp. 28-40, 2003.

A.K. Elmagarmid, P.G. Ipeirotis, and V.S. Verykios, Duplicate Record Detection: A Survey in IEEE Trans. Knowledge and Data Eng., vol. 19, no. 1, pp. 1-16, Jan. 2007.

V.R. Borkar, K. Deshmukh, and S. Sarawagi, Automatic Segmentation of Text into Structured Records in Proc. 2001 ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '01), pp. 175- 186, 2001.

C. Sutton, K. Rohanimanesh, and A. McCallum, Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data in Proc. 21st Int'l Conf. Machine Learning (ICML '04), 2004.

I. Bhattacharya and L. Getoor, Iterative Record Linkage for Cleaning and Integration in Proc. Ninth ACM SIGMOD Workshop Research Issues in Data Mining and Knowledge Discovery, pp. 11-18, 2004.

I.P. Fellegi and A.B. Sunter, A Theory for Record Linkage in J. Am. Statistical Assoc., vol. 66, no. 1, pp. 1183-1210, 1969.

M. Bilenko, R. Mooney, W. Cohen, P. Ravikumar, and S. Fienberg, Adaptive Name Matching in Information Integration in IEEE Intelligent Systems, vol. 18, no. 5, pp. 16-23, Sept./Oct. 2003.

M. Bilenko and R.J. Mooney, Adaptive Duplicate Detection Using Learnable String Similarity Measures in Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 39- 48, 2003.

J.R. Koza, Gentic Programming: On the Programming of Computers by Means of Natural Selection in MIT Press, 1992.

W. Banzhaf, P. Nordin, R.E. Keller, and F.D. Francone, Genetic Programming - An Introduction: On the Automatic Evolution of Computer Programs and Its Applications in Morgan Kaufmann Publishers, 1998.

S. N. Sivanandam and S. N. Deepak Introduction to genetic algorithms in Springer, NewYork, 2008.

Akshara k., Soorya P. Replica free repository using genetic programming with decision tree in International Journal of Advanced Engineering Applications, Vol.1, Iss.2, pp.62-66 (2012).

M.A. Hernandez and S.J. Stolfo, Real-World Data Is Dirty: Data Cleansing and the Merge/Purge Problem Data Mining and Knowledge Discovery, vol. 2, no. 1, pp. 9-37, Jan. 1998.

Y.R. Wang and S.E. Madnick, The Inter-Database Instance Identification Problem in Integrating Autonomous Systems in Proc. Fifth IEEE Int'l Conf. Data Eng. (ICDE '89), pp. 46-55, 1989.

W.W. Cohen, H. Kautz, and D. McAllester, Hardening Soft Information Sources in Proc. Sixth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '00), pp. 255-259, 2000.

M. Bilenko, R.J. Mooney, W.W. Cohen, P. Ravikumar, and S.E. Fienberg, Adaptive Name Matching in Information Integration in IEEE Intelligent Systems, vol. 18, no. 5, pp. 16-23, Sept./Oct. 2003.