IGAEM: Improved Genetic Algorithm based Entity Matching

Y. AASSEM, I. HAFIDI, N. ABOUTABIT
National School of Applied Sciences Khouribga, Sultan Moulay Slimane University.
E-mail: aassem.youssef@gmail.com

Abstract. The presence of duplicate records is a major data quality concern in huge datasets. To detect duplicates, entity matching is used as an essential step of the data cleaning process to map records that refer to the same real-world entity. Most of proposed algorithms require labeled data in order to train a classifier. However, we cannot always obtain labeled data. In our paper we propose an unsupervised approach for entity matching problem using an improved version of genetic algorithm. We explain the main improvements added to genetic algorithm and the encoding strategy to encode partitions in the form of a chromosome. Different similarity functions are used to compute similarities between records. The obtained results prove that our proposition stands as a powerful approach in the entity matching field where it outperforms the traditional genetic algorithm based approach.

1. Introduction
Nowadays, the data increases exponentially in huge amount. The challenge is to extract meaningful information from it. Knowing its structure may help in utilizing the data in a better way. For that purpose, user may provide the name to get an entity’s related information. Issues with multiple entries having same name create confusion about the information of the real-world entity. Entity resolution [1] is a critical task in data cleaning and data integration. The goal is to process data and map different entries to distinct real-world entities.

Naive algorithm consists of enumerating all possible pairs of comparisons which is tedious and has a quadratic complexity. Consequently it is necessary to reduce the search space and thus the execution time.

The first algorithm family that were proposed are blocking [2] and Sorted Neighborhood Method [3]. Both families rely on the concept of dividing data to blocks so that pairwise comparisons within each block are performed.

Entity matching can be considered as a partitioning problem where to goal is to partition the record so that all records in one partition should belong to one entity. Some of the suggested techniques for entity matching [2, 3, 4] perform the partitioning over the set of records automatically without knowing the number of clusters, while some existing techniques require the number of clusters as input [5].
Researchers have utilized diverse techniques [6, 7] using genetic programming for record linkage problems. Nevertheless, these approaches are supervised. Our proposed approach is an unsupervised approach that utilizes the search capabilities of genetic algorithm (GA) to solve the entity matching problem. We have modeled the partitioning problem as an optimization problem and applied GA to solve it.

In the current paper, we have suggested a genetic algorithm based framework for solving the record linkage problem. For the encoding of the partitions, we used medoid based representation. Various distance measures are explored for calculating similarity between records.

2. Related work

The problem of entity is famous issue in a data driven community. Industries and researchers are trying to solve this problem. A myriad of techniques were proposed to solve that problem [2, 3, 4].

For the machine learning category, they were some active works. Distinct [8] was proposed. Authors utilize SVM classifier to identify various linkage types. The classifier is trained using a set of distinct objects. The case of author with single record is excluded. Authors assert that the exclusion would not impact the accuracy, but there are many datasets having entities with one single record.

Another approach was proposed, which is based on a probabilistic framework on a probabilistic framework [9] the drawback of that method is that it requires the number of entities to be specified in advance which is not practical. A may to overcome this limitations is by estimating the number of entities. It was suggested in one of their recent studies [10].

Mishra et al [11] have considered the problem of automatic partitioning of records as an optimization problem. authors have proposed a genetic algorithm based automatic technique to solve the entity matching problem.

The approach determines automatically the number of partitions available. Despite the good performance compared to DBLP and ArnetMiner, the approach is designed only for bibliographic dataset and wasn’t applied on other types.

Another recent work [12] in which Mishra et al utilize another algorithm of genetic algorithm for multi-objective optimization. They have employed NSGA-II algorithm in order to optimize the partitioning following multiple objective functions. The study has shown that for certain combinations of objective functions the proposed technique can produce better results in many cases. But still has the same drawback of being designed only for bibliographic dataset.

3. The proposed algorithm

In this section we present our proposed approach. We remind the steps of genetic algorithm as illustrated in the following steps.
input : Records R = \{r_1, r_2, ..., r_n\}, Objective function

output: Clustering arrangement of R

1 Initialize population P;
2 Evaluate the fitness of all individuals;
3 Select fitter individuals for reproduction;
4 Apply recombination among individuals;
5 Evaluate the fitness of generated solutions;
6 Generate new population;

Algorithm 1: Genetic Algorithm

The first step is population initialization in which a population of n chromosome is generated. A chromosome is an individual that represent a solution for our problem. The $i^{th}$ gene represents one or multiple records from the $i^{th}$ cluster. The length of a chromosome is randomly chosen. After randomly choosing the genes, the record assignment step takes place in order to assign the remaining records to the closest clusters represented by genes.

Next, the generated chromosome must have an assigned score in order to differentiate fitter individuals for reproduction. The evaluation metrics are objective function that are developed for the purpose of measuring the quality of clustering. An objective function is a combination of separation and compactness. Separation measures the distance between clusters. Hence it should be maximized. While the distance between records within the same cluster must be minimized, and it is referred to as compactness.

The following step is the selection of fitter individuals for reproduction based on their score. A number of random re-combinations are applied on those chromosomes in order to generate more powerful individuals. A recombination can be performed by applying mutation and/or crossover operators to chromosomes. Based on fitter individuals from both parents and their modified version, we generate a new population which is more powerful.

The process is repeated in each iteration until a number of iteration is achieved. At the end, the most fitter chromosome is selected as the optimal solution:

The quality of the solution calculated by genetic algorithm is conditioned by two major steps that are population initialization and record assignment steps.

- **Population initialization :**

The first population is initialized randomly with genes representing a cluster that has initially one record. In order to have more concise chromosomes genes must represent elements from distinct entities. Otherwise, multiple genes may represent the same entity. Thus, individuals are less fitter for reproduction or as final solution. Moreover, initializing n chromosome each one with a random length k may generate same or approximate combinations of genes across multiple chromosomes. Therefore, diversification is not guaranteed and the solution may not converge quickly to an optimum solution.

- **Record assignment :**

In this step, unassigned records are assigned to clusters. The process consists of comparing each records with other records from each cluster and to cluster to which the distance
is minimum will contain the record. Applying complex similarity function on multiple attributes will result in a huge execution time and consequently low efficiency of the algorithm.

For the sake of enhancing the genetic algorithm, we have added two features to the underlying steps.

- **The concept of diversification**: A relevant population must have different combination of record in order to cover multiple arrangement possibilities. We propose the following strategy for generating chromosome. The first chromosome takes k unique elements from the set of records. The second chromosome takes m unique elements from the remaining points and so on. . . Hence, the unicity of the genes of each chromosome permits guaranteeing the diversity of individuals and thus improve the quality of subsequent populations.

- **The concept of filtering**: A comparison between two record is the application of a complex similarity function on multiple attributes. The adaptive filtering technique [13] aims to reduce the number of record pair comparisons by ignoring the comparison between pairs that are considered to be unlikely matches.

The filtering can be done using two modes:

- Length filtering: where two records are declared as unlikable matches if the difference between their filtering variables is greater than a predefined value k.

- Count filtering: where each of the two filtering variables is divided into a set of bigrams, then the number of common bigrams is compared to a predefined value Cm. If the number of common bigrams is lower than \( C_m \) then the records are declared as unlikable matches.

Experiments showed that this technique have reduced the number of comparison down to 80% [13]

In order to take advantage from that technique, its integration is done as the following:

- In the population initialization step, the first genes of each chromosome must be distinct. We ensure that condition by using filtering between the gene to be added and the existing ones. If they represent unlikable matches then the gene is distinct and will be assigned. That way we ensure distinct clusters, with minimum cost for similarity comparison.

- In the Record assignment step, before a record is added to a cluster, the calculation of the distance to all clusters is done. The incorporation of filtering permits to keep only likable matches of the current record. Therefore, the search space is reduced, the cost of unnecessary comparisons is minimized, and the execution time as well.

### 3.1. Objective function used

Entity resolution consists of mapping records to its corresponding distinct entity. Let \( R = \{r_1, r_2,..., r_n\} \) be the set of n records to be deduplicated. The goal is to split records to k clusters \( C = \{C_1, C_2,...C_K\} \). The cardinality of the cluster \( C_k \) is \( n_k \). The center of a cluster \( C_k \) is denoted by \( c_k \). The center of all records is denoted by \( c \).
The clustering of these records satisfies the following equations:

\[ \bigcup C_k = R \]

\[ C_k \cap C_k = \emptyset \quad 1 \leq K \neq k \leq K \]

The center calculation is described in the equation below:

\[ \text{center} = \arg\min_{r \in C_k} \frac{\sum_{r_j \in C_k, r_j \neq r} \text{dist}(r, r_j)}{n_k - 1} \]

Cluster validity indices [12] are utilized for the purpose of measuring the correctness of the obtained partitioning. In that paper we have considered CH index [14] as our objective function. It is described in details bellow

3.1.1. CH index

The Calinski-Harabasz index [14] is computed by the following equation.

\[ CH = \frac{\text{trace}(S_B)}{\text{trace}(S_W)} \times \frac{n-k}{k-1} \]

\[ \text{trace}(S_B) = \sum_{k=1}^{K} n_k \times \text{dist}(c_k, c) \]

\[ \text{trace}(S_W) = \sum_{k=1}^{K} \sum_{r_q \in C_k} n_k \times \text{dist}(r_q, C_k) \]

\[ \text{trace}(S_B) \] is related to the distance between clusters and should be maximized. \[ \text{trace}(S_W) \] is the internal distance which should be minimized. Thus better partitioning is achieved by maximizing the value of the index.

4. Experimental Evaluation

For evaluating the performance of our algorithm, it has been implemented in python using a Macbook Pro machine with 16 GB of ram and i7 processor. The dataset used are Restaurant and DBLP-ACM. The first dataset contains records about restaurants in USA. The second dataset presents record from bibliographic citations databases. In order to measure the performances of the proposed approach, the used metrics are F-score which is a harmonic mean between precision and recall. The percentage of saved comparison due to filtering which reflects the number of optimized comparisons related to unlikable matches.

For the restaurant dataset, the blocking key value used is the restaurant’s name concatenated with the phone number. For DBLP dataset, the blocking key used is the combination of the publication title and the year of publication.

| Datasets      | Number of instances | True matches |
|---------------|---------------------|--------------|
| Restaurant    | 862                 | 112          |
| DBLP-ACM      | 4910                | 2224         |

Table 1. Description of datasets used for evaluation purpose
For the experimental purpose, we use CH index [14] as objective function. Here we have executed the following experiments:

- We have utilized different distance measures.
- For a particular distance measure and for particular datasets we have executed the proposed algorithm 10 times. After execution, a set of solution is obtained. We select the best solution according to the F-measure value.
- Results are compared with the traditional genetic algorithm.

Table 2 and Table 3 show the obtained values for F-Score applied on both datasets using the traditional and improved version of genetic algorithm.

For restaurant dataset, considering Jaccard distance measure, improved genetic algorithm outperforms the genetic based implementation in terms of F-Score. For the other similarity functions as well, the same behavior persists. The gap of performance is due to the utilization of diversification principle in solution’s generation. The covering of more distinct solutions permits to enlarge the search space and increase the chance to find a global optimum.

DBLP-ACM dataset has different schema and presents complex relation between attributes. The performance of the two algorithms differ and has the same gap pattern. For all used distance metrics, the improved genetic algorithm perform better than traditional genetic algorithm. For the same number of iterations, the convergence of improved genetic algorithm to a better solution is due to the elimination of redundant combinations. In fact, the generation of more concise individuals, followed by their recombination through mutation and crossover generate fitter individuals after each generation, which leads to quick convergence to an optimum compared to the traditional genetic algorithm.

The second experiment aims to measure the filtering reduction ratio related the elimination of unlikable matches. The measured value represent the percentage of saved comparisons.

In figure one, the percentage of saved comparison due to the utilization of filtering is measured with regard to the variation of the filtering length S. A filtering length of zero is equivalent to not using filtering. And therefore the percentage of reduced comparisons will be equal to Zero. Meanwhile, as the length of filtering is increased, the number of unlikable matches grows as

| Algorithm       | Jaccard | Match-max | Levenshtein | Jaro-winkler |
|-----------------|---------|-----------|-------------|-------------|
| Restaurant      | 72      | 72        | 73          | 72          |
| DBLP-ACM        | 88      | 85        | 87          | 88          |

**Table 2. Results for genetic algorithm**

| Algorithm       | Jaccard | Match-max | Levenshtein | Jaro-winkler |
|-----------------|---------|-----------|-------------|-------------|
| Restaurant      | 84      | 84        | 83          | 83          |
| DBLP-ACM        | 92      | 90        | 89          | 89          |

**Table 3. Results for improved genetic algorithm**
Figure 1. Percentage of reduced comparisons for improved genetic algorithm

well and consequently the number of saved comparisons increases as well. Using this technique permits reducing the execution time of our algorithm and hence improving its efficiency

5. Conclusion and Future work

In this study, we address the problem of entity resolution from different sources. As data grows day by day, it is quite hard to manage huge number of redundant records. Consequently, Entity matching is an inevitable task in the field of data cleaning and integration. Most of proposed approaches are supervised in nature and require some labeled data. Meanwhile, most of real world datasets don’t provide labeled data. Being inspired by this observation we presented in this paper an unsupervised approach for entity matching. Considering the problem of entity matching as a partitioning problem where an objective function is to be optimized. A generic framework is proposed which can deal with any type of dataset. The proposed algorithm is based on genetic algorithm with some added features. Real encoding is used to represent clusters in the form of a chromosome. A myriad of mutation and crossover operators were utilized for recombination. The similarity measure is handled by some previously proposed distance measures. Our proposed approach is able to determine the number of partitions and also obtain the appropriate clustering. The performance of our algorithm is evaluate on two well known datasets. The results are compared with the traditional genetic algorithm and we have obtained good improvements.

This study raises some interesting ideas for future studies. The genetic algorithm based solution framework is very time consuming. In the future we would like to work on that path. A parallel implementation is required in order to reduce the time complexity. Moreover, it can be beneficial to apply same improvement on an multi-objective evolutionary algorithm. We believe that it will improve the performance of the framework for entity matching. Also, it would be interesting to work on automatic selection of attributes used for comparison.

References
[1] CHRISTEN P AND GOISER K, Quality and Complexity Measures for Data Linkage and Deduplication, Quality
Measures in Data Mining. Studies in Computational Intelligence 2007.

[2] Christen P, Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer Publishing Company. Incorporated 2012.

[3] Draisbach U, Naumann F, Szott S and Wonneberg O, Adaptive Windows for Duplicate Detection, Proceedings of the 2012 IEEE 28th International Conference on Data Engineering (ICDE '12). IEEE Computer Society 2012

[4] Aassem Y, Hafidi I and Aboutabit N. 2020, Enhanced Duplicate Count Strategy: Towards New Algorithms to Improve Duplicate Detection, In Proceedings of the 3rd International Conference on Networking, Information Systems Security (NISS2020) 2020.

[5] Benkhaled H and Berrabah D and Boufares F, A novel approach to improve the Record Linkage process, International Conference on Control, Decision and Information Technologies (CoDIT), 2019: p1504-09.

[6] Carvalho D M, Laender A, Alves M, Silva D A, A genetic programming approach to record deduplication, IEEE Transactions on Knowledge and Data Engineering 2012: p399-412.

[7] Isele R and Bizer C, Learning expressive linkage rules using genetic programming, Proceedings of the VLDB Endowment 2012 : p1638-49.

[8] Yin X, Han J and Yu P, Object distinction: Distinguishing objects with identical names, EEE 23rd International Conference on Data Engineering 2007

[9] Tang J, Zhang J, Yao L, Li J, Zhang L and Su Z, Arnetminer: extraction and mining of academic social networks, Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008: p990-98.

[10] Tang J, Fong ACM, Wang B and Zhang J A unified probabilistic framework for name disambiguation in digital library, IEEE Transactions on Knowledge and Data Engineering, 2012: p975-p987.

[11] Mishra S, Saha S and Mondal S, GAEMTBD: Genetic algorithm based entity matching techniques for bibliographic databases, Appl Intell, 2017: p197-p230.

[12] Mishra S, Saha S and Mondal S, A multiobjective optimization based entity matching technique for bibliographic databases, Expert Systems with Applications, 2016: p100-15.

[13] Gu L and Baxter R, Adaptive filtering for efficient record linkage, Proceedings of the 2004 SIAM International Conference on Data Mining, 2004: p477-81.

[14] Calinski T and Harabsz J , A dendrite method for cluster analysis, Communications in Statistics-Theory and Methods, 1974 : p1-27.