The Journal of Systems and Software 137 (2018) 335–354

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Heuristic-based approaches for speeding up incremental record linkage

Dimas Cassimiro do Nascimento[a,b,*], Carlos Eduardo Santos Pires[a], Demetrio Gomes Mestre[a]

[a] Federal University of Campina Grande, Brazil
[b] Federal Rural University of Pernambuco, Brazil

ARTICLE INFO

Article history:
Received 15 December 2016
Revised 24 November 2017
Accepted 30 November 2017
Available online 12 December 2017

Keywords:
Record linkage
Deduplication
Incremental clustering
Heuristics

ABSTRACT

Record Linkage is the task of processing a dataset in order to identify which records refer to the same real world entity. The intrinsic complexity of this task brings many challenges to traditional or naive approaches, especially in contexts such as Big Data, unstructured data and frequent data increments over the dataset. To deal with these contexts, especially the latter, an incremental record linkage approach may be employed in order to avoid (re)processing the entire dataset to update the deduplication results. For doing so, different classification techniques can be employed to identify duplicate entities. Recently, many algorithms have been proposed to combine collective classification, which employs clustering algorithms, together with the incremental principle. In this article, we propose new metrics for incremental record linkage using collective classification and new heuristics (which combine clustering, coverage component filters and a greedy approach) to speed up even more a solution to incremental record linkage. These heuristics have been evaluated using three different scale datasets and the results were analyzed and discussed based on both classical and the newly proposed metrics. The experiments present different trade-offs, regarding efficacy and efficiency results, which are generated by the considered heuristics. Also, the results indicate that, for large and frequent data increments, it is possible to slightly reduce efficacy results by employing a coverage filter-based heuristic that is reasonably faster than the current state-of-the-art approach. In turn, it is also possible to employ single-pass clustering algorithms, which are able to execute significantly faster than the state-of-the-art approach at the cost of sacrificing precision results.

1. Introduction

It has been largely recognized that real world data is often dirty and this problem may generate significant losses to enterprises (Sadiq, 2013). One kind of data quality problem that may affect datasets is the presence of duplicate entities. The task of identifying duplicate entities in a dataset is commonly referred as deduplication, record linkage or record reconciliation (Christen, 2012a). Deduplication has huge practical implications in a wide variety of applications like retrieval of image information, Internet monitoring and scientific data management (Dharavath and Kumar, 2015).

This task is particularly challenging for mainly four reasons: i) it presents quadratic complexity, with respect to the size of the dataset(s), to be solved when employing a naive approach; ii) it is often necessary to tune the values of quite a few parameters, such as similarity functions and thresholds; iii) after the completion of the task, many possible matches may need to be re-evaluated by means of a costly and slow human-based clerical review (Christen, 2012a) process; and iv) it is necessary to choose between different existing classification techniques to categorize entities as duplicated (matches), non-matches or possible matches. In turn, each one of these problems may be even more augmented by factors such as the unstructured nature of the data, changes in the business logic rules, the Big Data scale as well as the size and frequency of the data increments that affect a dataset. Thereby, many efforts (Sadiq, 2013; Christen, 2012a, 2012b; Loshin, 2010; Batini et al., 2009) have been made by researchers and practitioners to propose solutions that intend to deal with these challenges and aim to produce effective and efficient record linkage results.

To tackle the first problem, indexing techniques (Christen, 2012a, 2012b) were proposed aiming to reduce the number of

* Corresponding author.
E-mail addresses: dimascnf@copin.ufcg.edu.br (D.C. do Nascimento), cesp@dsc.ufcg.edu.br (C.E. Santos Pires), demetriogm@gmail.com (D. Gomes Mestre).

https://doi.org/10.1016/j.jss.2017.11.074
0164-1212/

comparisons that are performed by record linkage algorithms. This technique aims to group or order similar entities according to a specific criterion and limit the comparisons only between the entities that present common characteristics. This is usually done by evaluating the so-called blocking (or sorting) keys and comparing entities with identical or similar keys. To this end, it is employed blocking (Christen, 2012a, 2012b), windowing (Yan et al., 2007; Kolb et al., 2010) or canopy clustering (Christen, 2012a) techniques. Note that the fewer comparisons between the entities are determined by the indexing phase, the more is the risk of missing true matches and prejudicing the efficacy of the solution. Clearly, there is a trade-off between efficacy and efficiency that is generated by the indexing phase.

To handle the second problem, empirical investigations have been carried out (Draisbach and Naumann, 2013; Köpcke et al., 2010) to determine proper values for parameters regarding different types and contexts of datasets. In turn, the most effective way to minimize the third problem, which means to minimize the clerical review effort, is by trying to maximize the efficacy results generated by the employed record linkage algorithm.

Lastly, the classification phase may be performed based on a pairwise (Christen, 2012a), machine learning (Elmagarmid et al., 2007), rule (Whang and Garcia-Molina, 2014), reference table (Wang et al., 2013) or a collective (Gruenheid et al., 2014; Hassanzadeh et al., 2009) classification approach. The latter approach aims to classify entity pairs not only based on their pair-wise similarities but also using information on how records are related or linked to other entities (Christen, 2012a). In practice, the challenge of applying collective classification is to scale this approach when dealing with Big Data datasets or dynamic datasets (particularly, large and frequent data increments). To tackle these scenarios and still present acceptable performance results, Incremental Record Linkage (IRL) (Whang and Garcia-Molina, 2014; Gruenheid et al., 2014; Costa et al., 2010) approaches can be employed. IRL consists in re-processing only the portion of the matching results that were affected by the data increments. Recent breakthroughs have been made (Whang and Garcia-Molina, 2014; Gruenheid et al., 2014) regarding the combination of collective classification with incremental record linkage and this attempt has shown to be very promising according to the obtained state-of-the-art results.

Given the promising results obtained by collective classification-based IRL approaches, in this article, we aim to investigate how to employ heuristics to speed up even more the IRL process and still maintain reasonable efficacy results. For doing so, taking into account the goals and algorithms for evaluating and solving IRL, which are available in the state of the art, we raise the following research questions:

- *How to employ heuristics to speed up even more the IRL process, compared to the state-of-the-art approach, and still maintain acceptable efficacy results*?
- *How to measure the degree of stability, i.e., the performance and efficacy over time, related to an IRL method*?
- *Does the appliance of single-pass clustering algorithms (Hassanzadeh et al., 2009) produce good IRL results*?
- *Does the combination of single-pass clustering algorithms with a greedy approach produce good IRL results*?
- *How to efficiently select a few clusters to be processed by an IRL method and still maintain acceptable efficacy results*?

These questions are investigated either theoretically or empirically throughout this article. In particular, we offer four contributions in this article. First, we propose new metrics for evaluating both the efficacy and efficiency of IRL results. Second, we propose coverage component filters, which are heuristics that aim to limit the number of clusters processed by IRL methods. Third, we propose a metaheuristic that allows the generation of different heuristic-based IRL methods. Fourth, we propose heuristics that are able to speed up even more collective classification-based IRL by combining: i) well-known single-pass clustering algorithms; ii) coverage component filters; and iii) a state-of-the-art Greedy approach. Finally, we evaluate the proposed heuristics, using different scale datasets and the results regarding both classical and the proposed metrics, when compared to the state-of-the-art approach (Greedy Algorithm Gruenheid et al., 2014).

The rest of this article is organized as follows. Section 2 discusses a motivating example, lists the existing goals for collective classification-based IRL and presents the adopted notation regarding graph representation and clustering. Section 3 presents new metrics for IRL. Section 4 presents single-pass clustering algorithms and the proposed metaheuristic and coverage component filters that are employed in this article. Section 5 describes the experimental goals, results and related discussions. In Section 6, we discuss related work. Finally, in Section 7 we conclude the article and present perspectives for further works.

2. Incremental record linkage

In this section, we explain the overall processes of collective classification and collective classification-based IRL (henceforth, IRL), introduce the adopted notation and present a motivating example.

2.1. Collective classification-based IRL

Clustering is the process whereby a set of entities (or records) is divided into several clusters, the members of each cluster being in some way similar to each other and different from the members of the other clusters (Zat and Messatfa, 1997). Clustering is a well-known data mining technique and is useful in a variety of applications, including: pattern-analysis, decision making, image segmentation, document retrieval and duplicate detection. In this article, clustering is employed in order to aid the classification step of the incremental record linkage problem.

The processes of collective classification of duplicated entities and IRL are summarized in Fig. 1 and Fig. 2, respectively. The former process works as follows. First, the dataset is initially submitted to a similarity join (Das Sarma et al., 2014) operation. Ideally, efficient state-of-the-art approaches (Das Sarma et al., 2014; Vernica et al., 2010; Jacox and Samet, 2008; Okcan and Riedewald, 2011; Xiao et al., 2011) should be used to execute the similarity join in order to speed up this step. The output of this step is then used to create a similarity graph, which maps each record in the dataset to a vertex and generates an edge between each pair of vertices whose similarity is greater or equal to a predefined input threshold. After that, the similarity graph is submitted to a clustering algorithm, which usually tries to maximize or minimize an objective function, in order to produce disjoint clusters (also known as non-overlapping clustering Gutierrez-Rodríguez et al., 2015). As a result, each cluster represents the duplicated entities, and thus, the collective clustering results represent the record linkage results.

In turn, an IRL workflow (Fig. 2) is performed each time a data increment affects the dataset. First, the data increments (i.e., insert, update or delete operations) are applied to the dataset. Then, these modifications are reflected on the similarity graph. Afterwards, a specific strategy is employed to select a subset of clusters to be updated (i.e., improved). The selected clusters are then updated in order to consider the newly processed data increments. Once this phase is completed, the record linkage results are consequently updated as well. Alternatively, an indexing technique, such as blocking (Christen, 2012b), can be applied to the dataset prior to the