EVALUATION OF SCALABLE PPRL SCHEMES WITH A NATIVE LSH DATABASE ENGINE

Dimitrios Karapiperis[1], Chris T. Panagiotakopoulos[2] and
Vassilios S. Verykios[3]

[1]School of Science and Technology, Hellenic Open University, Greece
[2]Department of Primary Education, University of Patras, Greece
[3]School of Science and Technology, Hellenic Open University, Greece

ABSTRACT

In this paper, we present recent work which has been accomplished in the newly introduced research area of privacy preserving record linkage, and then, we present our L-fold redundant blocking scheme, that relies on the Locality-Sensitive Hashing technique for identifying similar records. These records have undergone an anonymization transformation using a Bloom filter-based encoding technique. We perform an experimental evaluation of our state-of-the-art blocking method against four other rival methods and present the results by using LSHDB, a newly introduced parallel and distributed database engine.

KEYWORDS

Locality-Sensitive Hashing, Record Linkage, Privacy-Preserving Record Linkage, Entity Resolution

1. INTRODUCTION

A series of economic collapses of bank and insurance companies recently triggered a financial crisis of unprecedented severity. In order for these institutions to get back on their feet, they had to engage in merger talks inevitably. One of the tricky points for such mergers is to be able to estimate the extent to which the customer bases of the constituent institutions are in common, so that the benefits of the merger can be proactively assessed. The process of comparing the customer bases and finding out records that refer to the same real world entity, is known as the *Record Linkage*, the *Entity Resolution* or the *Data Matching* problem.

Record Linkage consists of two main steps. In the first step potentially matched pairs of records are searched, while in the second step these pairs are matched. The *searching step*, or commonly known as *blocking*, addresses the problem of bringing together for comparison tentative matched pairs of records, while disregarding the unpromising ones. The searching step should be able to identify a minimal superset of the matched pairs so that no computational resources are wasted in comparison operations during the following step. The second step, known as the *matching step*, entails the comparison of record pairs which have been brought together for comparison in the previous step. The matching step is implemented either in an exact or in an approximate manner. An exact matching of two records can be regarded as a binary decision problem with two possible outcomes denoting the agreement or disagreement of these records. Approximate matching

David C. Wyld et al. (Eds) : CCSEIT, AIAP, DMDB, ICBB, CNSA - 2017
pp. 125– 132, 2017. © CS & IT-CSCP 2017 DOI : 10.5121/csit.2017.70612

comprises the calculation of a continuous value similarity metric that usually assumes values in the range of [0,1].

When data to be matched is deemed to be sensitive or private, such as health data or data kept by national security agencies, *Privacy-Preserving Record Linkage* (PPRL) techniques should be employed. PPRL investigates how to make linkage computations secure by respecting the privacy of the data, and imposes certain constraints on the two steps of Record Linkage just described, on top of the necessary anonymization of the input records. In addition, the anonymization of the records must be implemented in such a way that (a) no sensitive information in a record is disclosed to parties other than the owner, (b) the anonymization process to be time and cost efficient, and (c) the final deliberation about the linking status of a pair of records, that relies on the comparison of their anonymized form, should be a close approximation of the distance between their original record counterparts.

In this paper, we elaborate on the details of our proposed flexible L-fold redundant blocking scheme, which is structured around an efficient technique for searching potentially matching record pairs. More specifically, our scheme relies on the idea of blocking one record to multiple groups in order to amplify the probability of inserting similar records into the same block. We use the so-called *Locality-Sensitive Hashing* (LSH) technique [1], where we utilize only the necessary number of hash tables. By doing so, we achieve accurate results without imposing unnecessary and additional computational overhead. This LSH-based searching method, as shown experimentally in Section 4, can reduce the number of record pairs that are brought together for comparison up to 98% of the total comparison space. Experimental results demonstrate the effectiveness and the superiority of our method by comparing it with four state-of-the-art private blocking methods.

The structure of the paper is organized as follows: related work is given in Section 2. In Section 3, we illustrate some basic building components used by our approach, while Section 4 formulates the problem being solved. Section 5 exposes the details of our proposed scheme including a theoretical analysis. Section 6 provides an experimental evaluation of our approach against four other state-of-the-art blocking methods. Conclusions are discussed in Section 7.

2. RELATED WORK

Several solutions have been proposed in the literature in the field of efficiently blocking(or searching) similar records. However, the majority of these solutions exhibit poor performance when applied to large data sets. Next, we provide a categorization of these methods in order to be able to study their unique characteristics, as well as to be able to compare them.

We divide the blocking solutions into seven main categories:

- The tree-based blocking methods[4,12] use space-partitioning data structures (KD-Trees, R-Trees etc) to divide a space into non-overlapping regions.

- The hierarchy-based searching which relies on the categorization of records into generalized hierarchies based on the semantics of values of selected fields [3].

- The reference-based clustering [6,11,13] where global clusters are created based on publicly available sets of values.

- The multi-sampling reference-based transitive closure clustering [7].

- The neighborhood-based searching [6] which uses the sorted neighborhood method [2] that creates windows of possibly similar records.

- The randomized hash-based blocking [8, 9, 10] which relies on the Locality-Sensitive Hashing technique.

3. PROBLEM FORMULATION

Let us assume, two data custodians, Alice and Bob, who wish to link their records. Since, these data are considered as sensitive, they have to independently anonymize them. These anonymized records should *securely* reflect the linking status of the original records, so that the linkage process can be feasible. The anonymized data sets are then submitted to a Trusted Third Party (TTP) that will conduct the linkage process.The PPRL process is summarized in Figure 1.

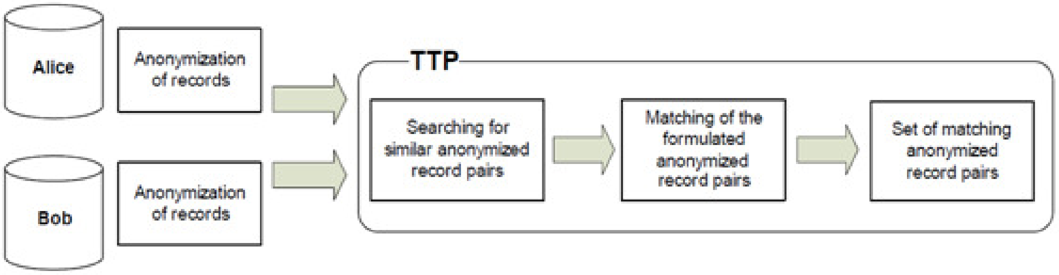

Alice
Anonymization of records
Bob
Anonymization of records
TTP
Searching for similar anonymized record pairs
Matching of the formulated anonymized record pairs
Set of matching anonymized record pairs

Figure 1. Alice and Bob submit their anonymized data sets to a Trusted Third-Party (TTP), which performs the linkage task.

Due to the large number of records in modern databases, searching for matching pairs using the brute-force approach is quite inefficient. Therefore, the TTP should utilize a blocking method that will mostly generate matching pairs and will provide *theoretical guarantees of completeness* of the generated results.

4. BACKGROUND

4.1 Anonymization of Records Using Bloom Filters

Bloom filters [14] are widely used in the literature due to their effectiveness and simplicity. A Bloom filter is implemented as a bitmap array of size ρinitialized with zeros. In order to represent a string as a Bloom filter, we hash each q-gram of that string using a number b of keyed hash message authentication code (*HMAC*) functions, such as *HMAC-MD5* and*HMAC-SHA2* which associate b positions to certain q-grams (the number of possible q-grams is much larger compared to the available positions).Guidelines for enhancing the privacy of Bloom filters can be found in[15]. Figure 2 illustrates the creation of field-level and record-level Bloom filters.

4.2 Hamming Locality Sensitive hashing (HLSH)

This technique guarantees that almost every *similar* record pair will be identified with high probability. HLSH works in the binary Hamming metric space $S = \{0, 1\}^{\rho}$,where ρdenotes its dimensionality. Therefore, records should be embedded in S, for example as Bloom filters, in order to use HLSH. The similarity between a pair of records is defined by a distance threshold θ($d <= \theta$).

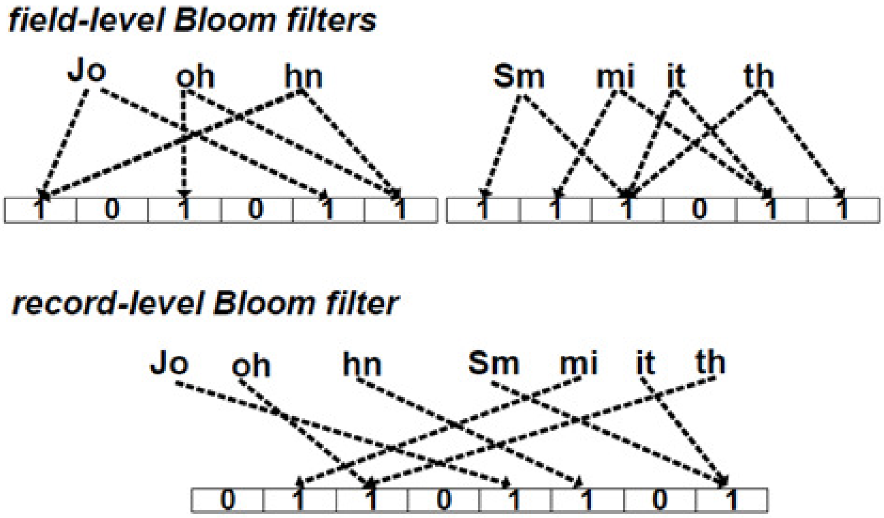


Figure 2. Creating field-level and record-level Bloom filters.

5. Randomized LSH-Based Blocking Using HLSH

In this section, we present the details of our LSH-based proposed method and then introduce LSHDB a freely available similarity database.

5.1 HLSH

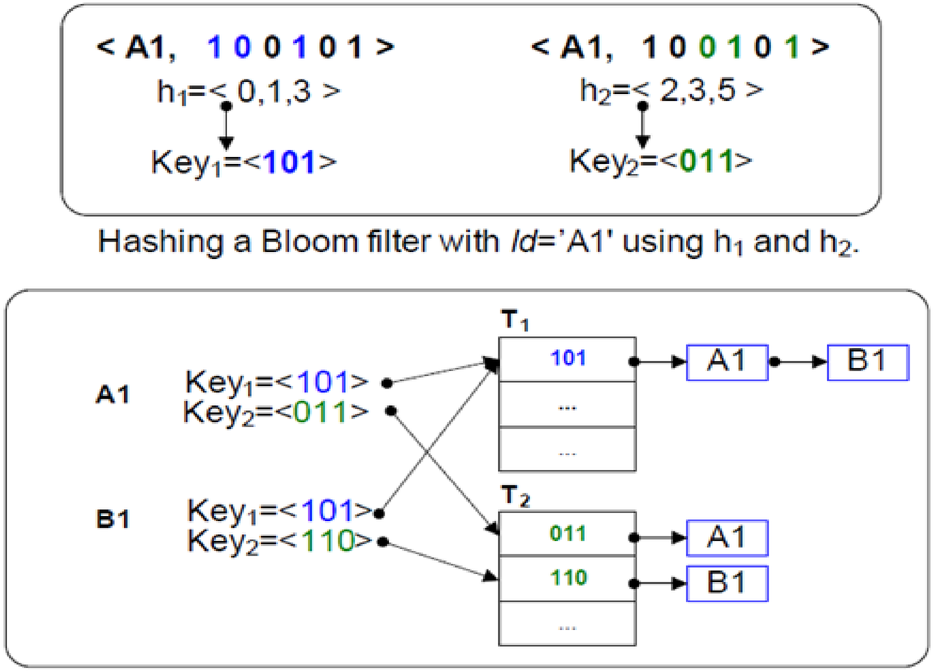
HLSH bases its operation on *L* independent hash tables. Each hash table, denoted by T_l where *l=1,...,L*, consists of key-bucket pairs where a bucket hosts a linked list which is aimed at grouping similar Bloom filter pairs. Each hash table has been assigned a composite hash function h_l which consists of a fixed number k of base hash functions. A base hash function applied to a Bloom filter returns the value of its j-th position where $j \in \{0,\ldots,\rho-1\}$ chosen uniformly at random. The result of ah_l, which essentially constitutes the blocking key, specifies into which bucket of some T_l, a Bloom filter will be stored. This randomized process is illustrated in Figure 3.

We assume a pair of Bloom filters of distance d less than or equal to a predefined threshold θas a *similar pair*. The smaller the Hamming distance of a Bloom filter pair is, the higher the probability for ah_l to produce the same result. During the matching step, we scan the buckets of each T_l and formulate Bloom filter pairs.

The optimal number *L* of the T_l's that should be utilized by HLSH is:

$$L = \lceil \frac{\ln(\delta)}{\ln(1-p^k)} \rceil,$$

where p denotes the probability of a base hash function of producing the same result by hashing two similar Bloom filters. By using this structure, each similar Bloom filter pair will be returned with high probability 1 -δ, as δ is usually set to a small value, say δ=0.1.



Hashing a Bloom filter with *Id*='A1' using h_1 and h_2.

Inserting into the buckets of T_1 and T_2 the *Id*'s of the Bloom filters. Similar Bloom filters with *Id*'s 'A1' and 'B1' are blocked together into a bucket of T_1.

Figure 3. Hashing a pair of similar Bloom filters using HLSH.

5.2 THE LSHDB PARALLEL AND DISTRIBUTED ENGINE

LSHDB[16] is the first parallel and distributed engine for record linkage and similarity search. LSHDB materializes an abstraction layer to hide the mechanics of the Locality-Sensitive Hashing, which is used as the underlying similarity search engine. LSHDB creates the appropriate data structures from the input data and persists these structures on disk using a noSQL engine. It inherently supports the parallel processing of distributed queries, is highly extensible, and is easy to use.

Upon the creation of a database, termed as data store, the developer needs to specify only two parameters: (i) the LSH method that will be employed, e.g.,Hamming, Min-Hash, or Euclidean LSH, and (ii) the underlying noSQL data engine that will be used to host the data. After these decisions have been made, LSHDB builds the necessary hash tables, which are stored on disk by the chosen noSQL system. To the best of our knowledge, LSHDB is the first record linkage and similarity search system in which parallel execution of queries across distributed data stores is inherently crafted to achieve fast response times.

6. EXPERIMENTAL EVALUAION

We evaluate HLSH in terms of (a) the accuracy in finding the truly matching record pairs, (b) the efficiency in reducing the number of candidate pairs, and (c) the execution time. We use two semi-synthetic data sets, denoted by A and B, of size equal to 1,000,000 records each, extracted from the NCVR list (http://dl.ncsbe.gov/index.html?prefix=data/). Insert, edit, delete, and transpose operations, chosen in random order, are used to perturb the values of each field of certain marked records from A, which are placed in set B. For the experiments, we used a simple PC with an Intel i5-2400 and 16 GB RAM. The software components were developed using the Java programming language (JDK 1.8).

The Pairs Completeness(PC) and the Reduction Ratio (RR) metrics are employed to evaluate the accuracy in identifying the matching record pairs and the reduction of the comparison space, respectively. PC denotes the number of the truly matching record pairs identified by each method. Conversely, RR indicates the percentage of the reduction of the total comparison space between the two data sets. Specifically, the fraction of the number of distance computations performed to the total number of all possible distance computations subtracted from 1. We ran each experiment 10 times, and plotted the average values in the figures shown below.

We compared HLSH with four state-of-the-art blocking methods. The first of these methods, denoted by KDT [12], relies on kd-trees to formulate blocks of records which have previously been embedded into the Euclidean space. The second method, symbolized by HG [3], categorizes the records into generalized hierarchies based on the semantics of values of selected attributes. PHN [5] uses phonetic encoding functions to generalize strings, while AHC [11] employs agglomerative hierarchical clustering to create blocks, which are generated by the TTP for a set of public reference values of a chosen field. Then, each data custodian assigns her records into the formulated blocks.

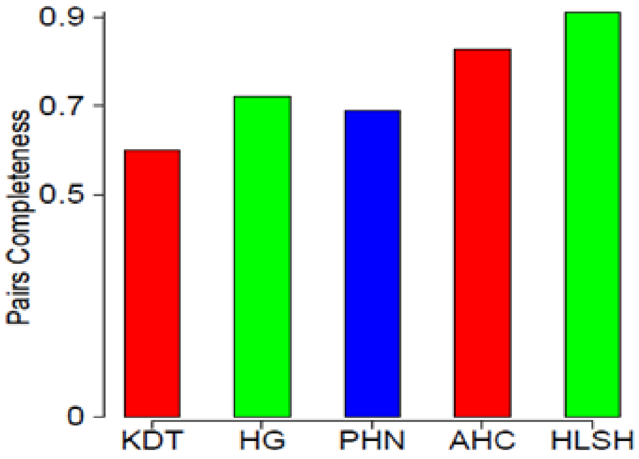


Figure 4.The Pairs Completeness rates.

Figure 4 illustrates the PC rates achieved by each method. We observe that HLSH and AHC achieve the highest scores. However, we have to note that the performance of AHC is highly dependent on the choice of the reference values. We tested several sets of reference values but only achieved high PC rates, when those sets were supersets of the field values. Conversely, if those sets were not supersets of the field values, the PC rates dropped considerably below 70%. HG and PHN achieved stable performance, whose rates, however, were also around 70%. KDT exhibited large deviations from its mean rate, mainly due to the deficiencies of the embedding method used.

The reduction of the comparison space, as measured by the RR, is shown in Figure 5. HLSH and AHC exhibit comparable performance reaching almost 98% reduction. PHN scores rates very close to 90%, while HG and KDT exhibit inferior performances.

7. CONCLUSIONS

Linking large collections of records by simultaneously protecting their privacy has arisen recently as an intriguing problem in the core of the domain known as Privacy-Preserving Record Linkage. In this paper, we expose the details of HLSH blocking method, and experimentally compare it with four state-of the-art private blocking methods in the context of LSHDB, an newly introduced data engine for big data computations. HLSH outperformed these methods in terms of the accuracy of the results as well as the running time.

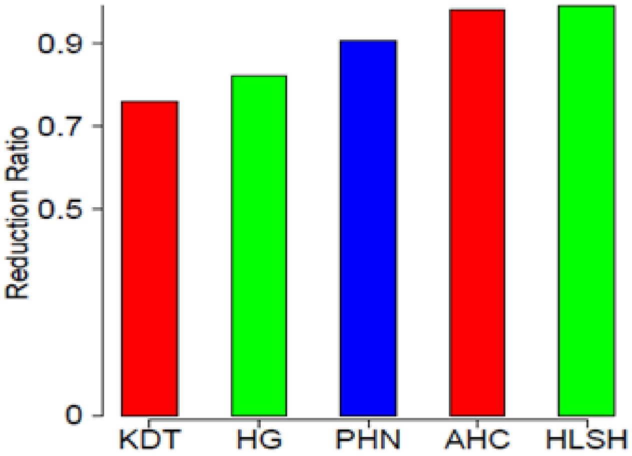


Figure 5.The Reduction Ratio.

REFERENCES

[1] A.Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In VLDB, pages 518–529, 1999.

[2] M.A. Hernandez and S.J. Stolfo. Real world data is dirty: Data cleansing and the merge/purge problem. Data Min. And Knowl.Disc., 2(1):9 – 37, 1988.

[3] A. Inan, M. Kantarcioglou, E. Bertino, and M. Scannapieco. A hybrid approach to private record linkage. In ICDE, pages 496 – 505, 2008.

[4] A. Inan, M. Kantarcioglu, G. Ghinita, and E. Bertino. Private record matching using differential privacy. In EDBT, pages 123 – 134, 2010.

[5] A. Karakasidis and V.S. Verykios. Privacy preserving record link a geusing phonetic codes. In BCI, pages 101 – 106, 2009.

[6] A. Karakasidis and V.S. Verykios. A Sorted Neighborhood Approach to multidimensional Privacy Preserving Blocking. In ICDMWorkshops, pages 937 – 944, 2012.

[7] A. Karakasidis, G. Koloniari, and V. S. Verykios. Scalable Blocking for Privacy Preserving Record Linkage. In KDD, pages 101 – 106, 2015.

[8] D. Karapiperis and V.S. Verykios. An LSH-based Blocking Approach with a Homomorphic Matching Technique for Privacy-PreservingRecord Linkage. TKDE, 27(4):909–921, 2015.

[9] D. Karapiperis and V.S. Verykios. A fast and efficient HammingLSH-based scheme for accurate linkage. KAIS, pages 1–24, 2016.

[10] H. Kim and D. Lee. Fast Iterative Hashed Record Linkage for Large-Scale Data Collections. In EDBT, pages 525 – 536, 2010.

[11] M. Kuzu, M. Kantarcioglu, A. Inan, E. Bertino, E. Durham, and B. Malin. Efficient privacy-aware record integration. In EDBT,pages 167 – 178, 2013.

[12] M. Scannapieco, I. Figotin, E. Bertino, and A.K. Elmagarmid. Privacy preserving schema and data matching. In SIGMOD, pages 653 – 664,2007.

[13] D. Vatsalan, P. Christen, and V. Verykios. Efficient two-party private blocking based on sorted nearest neighborhood clustering. In CIKM,pages 1949 – 1958, 2013.

[14] R. Schnell, T. Bachteler, and J. Reiher. Privacy-preserving record linkage using Bloom filters. Central Medical Inf. and Decision Making, 9, 2009.

[15] F. Niedermeyer, S. Steinmetzer, Martin M. Kroll, and R. Schnell.Cryptanalysis of Basic Bloom Filters Used for Privacy Preserving Record Linkage. JPC, 6(2), 2014.

[16] D. Karapiperis, A. Gkoulalas-Divanis, and V. Verykios. LSHDB: a parallel and distributed engine for record linkage and similarity search. In ICDMW (DEMO), pages 1-4, 2016.

AUTHORS

Dr.DimitriosKarapiperis is a post-doctoral associate with the Hellenic Open University (HOU), Greece. He holds a PhD degree from HOU, an MSc degree from the University of York, UK, and a BSc degree from the Technological Institute of Thessaloniki, Greece. His research interests lie in the areas of privacy-preserving record linkage, entity resolution, and similarity search, where he develops randomized algorithms and data structures.

Chris T. Panagiotakopoulos is α Professor with the Division of General Sciences, Department of Primary Education at the University of Patras, Greece. His research interest is focused on the Educational Technology and especially on the development and use of educational software, educational robotics, web technologies and open and distance learning.

Vassilios S. Verykios received the Diploma degree in Computer Engineering from the University of Patras, Greece in 1992, and the MSc and the PhD degrees from Purdue University, USA in 1997 and 1999, respectively. Since January of 2016, he is a Professor in the School of Science and Technology at the Hellenic Open University in Greece where he has been the Director of the Graduate Program on Information Systems since 2012.