Blocking anonymized data

Jordi Nin
IIIA, Artificial Intelligence Research Institute
CSIC, Spanish National Research Council
Campus UAB s/n
08193 Bellaterra (Catalonia, Spain)
jnin@iiia.csic.es

Vicenç Torra
IIIA, Artificial Intelligence Research Institute
CSIC, Spanish National Research Council
Campus UAB s/n
08193 Bellaterra (Catalonia, Spain)
vtorra@iiia.csic.es

Abstract

Nowadays, privacy is an important issue, for this reason many researchers are working in the development of new data protection methods. The aim of these methods is to minimize the *disclosure risk* (DR) preserving the data utility. Due to this, the development of better methods to evaluate the DR is an increasing demand. A standard measure to evaluate disclosure risk is record linkage (RL). Normally, when data sets are very large, RL has to split the data sets into blocks to reduce its computational cost.

Standard blocking methods need a non protected attribute to build the blocks and, for this reason, they are not a good option when the protected data set is completely masked. In this paper, we propose a new blocking method which does not need a blocking key to build the blocks, and therefore, it is suitable to split fully protected data sets. The method is based on aggregation operators. In particular, in the OWA operator.

Keywords: Blocking methods, OWA Operators, Record Linkage.

1 Introduction

Managing large volumes of confidential data is a common practice in any organization. In many cases, it is necessary to protect this confidential data in order to publicly release it without revealing confidential information that could be linked to an specific individual or entity.

Significant efforts have been made to develop a wide range of protection methods [1, 4]. In order to compare two protection methods, it is necessary to use a *score* [3] which takes into account both *Disclosure Risk* (DR) and *Information Loss* (IL). Normally, disclosure risk is calculated using different *Record Linkage* (RL) methods [11], and IL is calculated computing the difference among several statistics between the original and the protected data set.

Theoretically, RL compares all the records in the data sets under analysis in order to decide which records belongs to the same individual. In practice, since the size of the data sets is usually very large, comparing all the records between them becomes unfeasible. Therefore, RL resorts to blocking methods [6, 7] that try to gather all the records that present a potential resemblance, only applying RL within each block. Typically, blocking methods are based on a common attribute without errors.

Usually, protection methods modify the values of the original data set to difficult the linkage between the protected and the original data set. Since all attributes in the protected data set present some noise, the application of standard blocking methods is unfeasible.

In this paper, we present the *fuzzy blocking* (FB), a new blocking method which substitutes the blocking key by an OWA operator[13] with a fuzzy quantifier. We will show in this paper that FB method outperforms standard methods comparing the number of well classified records inside the blocks.

The structure of the paper is as follows. In Section 2 we explain some basics needed to understand our approach. Then, in Section 2, we present our approach to *fuzzy blocking*. Section 3 describes the experiments. Finally, the paper finishes with some conclusions and a description of future work.

2 Preliminaries

In this section we review a few definitions that are needed latter on. We start with a brief explanation

of the two standard blocking methods, following, we continue with the definition of the OWA operator in terms of a fuzzy quantifier.

2.1 Traditional Blocking Methods

2.1.1 Standard Blocking.

The Standard Blocking method (SB) clusters records that share the same blocking key (BK) [7] into blocks. A blocking key is defined based on information extracted from one or more attributes. Usually, a blocking key can be either a common categorical attribute, *e.g. marital status* {single, married, divorced and widowed}, or a common numerical attribute, *e.g. birth date*.

Selection of the attribute is a critical point in standard blocking. If the final blocks contain a large number of records, then RL algorithms may have a huge computational cost. If the final blocks contain a small number of records, then RL algorithms may not be able to find all the occurrences of the same individual and show a poor accuracy.

2.1.2 Sorted Neighborhood.

The Sorted Neighborhood (SN) method [6] sorts the records based on a sorting key (SK), and then moves a window called Sliding Window (SW) of fixed size l sequentially over the sorted records. RL is applied into the records inside the sliding window.

An important problem with sorted neighborhood arises if a number of records, larger than the window size, have the same value in a SK. For instance, let us suppose that we are using sorted neighborhood with two similar files based on a SK extracted from an attribute *'surname'*. Typically, if the data sources are large enough, there will be thousands of records containing the value *'William'* or *'Smith'* in that attribute and, therefore, not all the records with the same value in a SK will be compared.

2.2 Protection Methods

In this section, we review the three protection methods used in the experiments done in this paper: *Rank Swapping* (RS-p) [9], *Microaggregation* (MIC-vm-k) [5] and *Lossy Compression* (JPEG-k) [8].

The RS-p protection method sorts the values of each attribute. Then, each value is swapped with another sorted value chosen at random within a restricted range of size p. MIC-vm-k builds small clusters from v variables of at least k elements and replaces original values by the centroid of the clusters that the record belongs to. And finally, JPEG-k protects a data set interpreting it as an image and then computing the *compressed image* and replacing the original numerical values by the ones corresponding to such *compressed image*.

2.3 OWA operators

A function $Q : [0, 1] \rightarrow [0, 1]$ is a non-decreasing fuzzy quantifier if it satisfies: (i) $Q(0) = 0$; (ii) $Q(1) = 1$; (iii) $x > y$ implies $Q(x) \geq Q(y)$.

Let Q be a non-decreasing fuzzy quantifier [13], then a mapping $OWA_Q : \mathbb{R}^N \rightarrow \mathbb{R}$ is an *Ordered Weighting Averaging (OWA) operator* of dimension N if

$$OWA_Q(a_1, ..., a_N) = \sum_{i=1}^{N} (Q(i/N) - Q((i-1)/N)) a_{\sigma(i)}$$

where σ is defined as a permutation of $\{1, \ldots, N\}$ such that $a_{\sigma(i)} \geq a_{\sigma(i+1)}$.

3 Fuzzy Blocking (FB)

In the scenario presented in Section 1, the data sets where record linkage is applied only noisy attributes are not shared. Therefore, in this scenario, standard blocking methods are not suitable.

We propose a new type of blocking method which substitutes the blocking key by the results of an OWA operator. We call this method *Fuzzy Blocking* (FB).

The FB method works as follows:

1. The data in the data set is normalized.
2. A fuzzy quantifier is selected
3. OWA operator with the fuzzy quantifier is computed using all the attributes in each record
4. The OWA result is rounded and the record is blocked using this value as a blocking key.

The fuzzy blocking method has two parameters: The OWA quantifier that decides the way to aggregate, and the rounding method that decides the number of blocks to do.

4 Experiments

To analyze the feasibility of our approach we have tested our approach with two different experiments:

Experiment 1. The files of five standard parameterizations of the two best ranked protection method

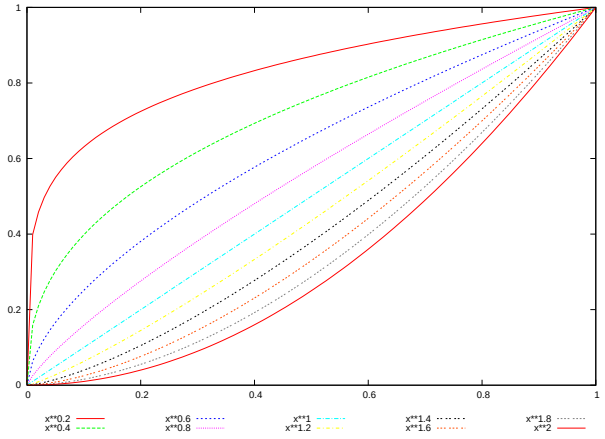

Figure 1. Graphical representation of Q_1^α

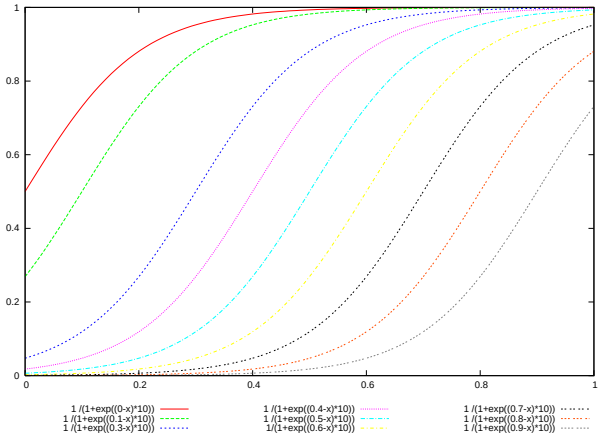

Figure 2. Graphical representation of Q_2^α

in the survey [4] are used. Such two methods are rank swapping [9] and microaggreagation [5]

Experiment 2. The files of the ten protection methods and parameterizations with less disclosure risk of the same survey than Experiment 1 are considered. These parameterizations belong to Rank Swapping and JPEG (Lossy compression) protection methods.

The data set used in [4] was extracted from the US Census Bureau and is described in detail in [2]. The *Census* data set contains 1080 records consisting of 13 attributes.

Before the application of our method we have considered a pre-processing step that consisted on the normalization of the data. This normalization are based on the translation of data values from the [max, min] interval into [0,1] using $x' = (x - \min(v))/(\max(v) - \min(v))$ (where x is the previous value, and $\max(v)$ and $\min(v)$ are the maximum and minimum values for the corresponding variable v).

In our experiments, we have tested three different fuzzy quantifiers. The quantifiers are defined below and their graphical representation is given in Figures 1, 2 and 3:

$$Q_1^\alpha(x) = x^\alpha \ \ for\ \alpha = 1/5, 2/5, \cdots, \ldots, 10/5$$

$$Q_2^\alpha(x) = 1/(1 + e^{(\alpha - x)*10}) \ for\ \alpha = \{0, 0.1, \ldots 0.9\}$$

$$Q_3^\alpha(x) = \begin{cases} 0 & \text{if } x \leq \alpha \\ 1 & \text{if } x > \alpha \end{cases} \quad for\ \alpha = \{0, 0.1, \ldots 0.9\}$$

The comparison between the fuzzy blocking method and the two standard blocking methods explained in

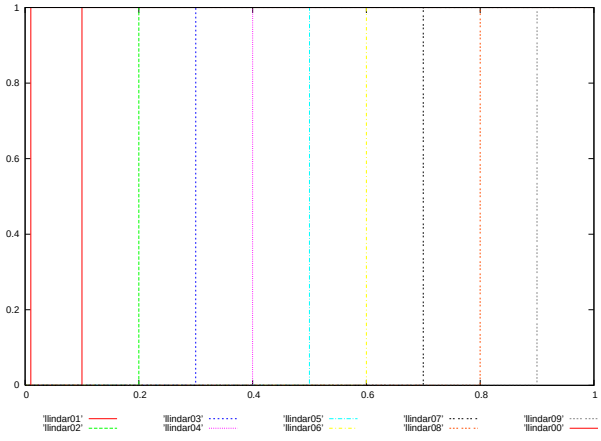

Figure 3. Graphical representation of Q_3^α

Section 2 pressumes that it is not possible to know which is the best attribute for building the blocks in standard methods and, analogously, that it is not possible to know which is the best quantifier for the fuzzy blocking method.

For this reason we show in Tables 1 and 2 the average results of the application of standard blocking and sorted neighborhood methods using one of the thirteen attributes each time. To test the fuzzy blocking method, we have executed one experiment for all possible parameterizations of the three quantifiers showed in Figures 1, 2 and 3. In the case of standard blocking and sorted neighborhood methods we have computed the average using thirteen test and for the fuzzy blocking method we have computed the average using thirty tests.

To compare all the methods equally, we have used a similar block size:

	FB	SB	%SB	SN	%SN
RS-1	1041.0	1025.0	1.56%	1080.0	-3.61%
RS-5	937.6	849.5	10.37%	1047.5	-10.49%
RS-10	840.3	672.0	25.04%	474.5	-22.9%
RS-15	752.8	527.4	42.74%	328.0	129.51%
RS-20	677.0	440.2	53.80%	249.2	171.70%
MIC-3m-7	775.5	737.3	5.18%	848.4	-8.59%
MIC-3m-9	737.4	702.8	4.93%	813.2	-9.32%
MIC-3m-10	716.7	679.1	5.54%	796.7	-10.04%
MIC-4m-4	876.1	825.5	6.12%	841.4	4.13%
MIC-4m-5	819.9	777.8	5.42%	806.3	1.69%

Table 1. Average execution results for the best protection methods. FB stands for Fuzzy Blocking, SB stands for Standard Blocking and SN stands for Sorted Neighborhood. Columns three and five are the % of improvement of FB respect SB and SN respectively.

FB Method. The blocks are built rounding the result of the OWA operator to one decimal. All the records with the same rounded result are clustered in the same block. As the range of the aggregation is [0,1], with this approach we are building ten different blocks.

SB Method. The parameterization of this method is similar to the fuzzy blocking method. The blocks are built rounding the normalized values of the attribute selected as key to one decimal.

SN Method. The size of the sliding window is fixed to the 10% of the data set size.

Table 1 shows the results for Experiment 1. As we can observe, fuzzy blocking always outperforms the results of the standard blocking method (see columns two and three), independently of the protection method and the parameterizations used to protect the data. The improvements of fuzzy blocking compared to standard blocking are in some case around to 50%.

If we compare the results of the fuzzy blocking method with the ones of the sorted neighborhood method (see columns four and five), we observe that in a few experiments sorted neighborhood method obtains better results than the fuzzy blocking method. However, fuzzy blocking outperforms in more than 100% the results obtained by sorted neighborhood method in some other cases. This is possibly due to the fact that for RS-1, RS-5 and RS-10 most of the values are swapped by other values inside the sliding window, and therefore, sorted neighborhood is the most suitable blocking method when Rank Swapping is used with a small swap parameter. We can observe that in RS-15 and RS-20 some values are swapped by values out of the sliding window. In this case, fuzzy blocking is clearly better than sorted neighborhood with improvements greater than 100%. A similar situation happens with microaggregation. Recall that microaggregation computes the protected value as the clustering of the centroid where the record belongs to. For this reason when the cluster has a small number of records, all the protected values are close to the original ones and sorted neighborhood shows a quite better results in some cases.

In Table 2, we show the average results obtained in Experiment 2. In this case, where the protected files have a lower disclosure risk, and, therefore, the noise addition in the protected data set is greater than in the Experiment 1, fuzzy blocking method outperform in all the experiments the results obtained by standard blocking and sorted neighborhood methods, with improvements in some cases up to 150%. This is possible due to the strong effect that noise causes in the key of the standard blocking methods. As the fuzzy blocking method does not need a key to build the blocks is more resistant to the effect of the noise in the data.

5 Conclusions and future work

In this paper, we have presented a new method for blocking data minimizing the effects of the noise in the process of building blocks. We have shown that fuzzy blocking outperforms traditional blocking methods.

Another relevant conclusion of the experiments is that knowledge on the protection method can ease the selection of the most suitable parameterization in the blocking method.

Future directions of this work include doing more experiments in other settings (*e.g.*, data cleaning or data integration) where data sets are not protected. That is, sets where noise is accidental (*e.g.*, misspellings or typos) and not added on purpose.

	FB	SB	%SB	SN	%SN
RS-20	677.0	440.2	53.80%	249.2	171.71%
RS-19	687.0	453.9	51.38%	259.9	164.32%
JPEG-10	558.0	352.3	58.38%	295.9	88.61%
RS-18	697.2	471.4	47.90%	272.1	156.24%
JPEG-15	589.9	383.7	53.74%	321.2	83.64%
RS-15	752.8	527.4	42.74%	328.0	129.51%
JPEG-20	655.0	428.3	52.93%	363.0	80.44%
RS-16	734.2	506.0	45.10%	298.0	146.38%
RS-17	724.6	469.6	54.30%	282.0	156.96%
RS-14	768.2	555.2	38.36%	338.5	126.92%

Table 2. Average execution results for protection methods with less DR. FB stands for Fuzzy Blocking, SB stands for Standard Blocking and SN stands for Sorted Neighborhood. Columns three and five are the % of improvement of FB respect SB and SN respectively.

Acknowledgments

This work was partly funded by MEC (project SEG2004-04352-C04-02). Jordi Nin wants to thank the Spanish Council for Scientific Research (CSIC) for his I3P grant.

References

[1] Adam, N. R., Wortmann, J. C., (1989), Security-Control for statistical databases: a comparative study. ACM Computing Surveys 21, 4, 515-556.

[2] CASC: Computational Aspects of Statistical Confidentiality, EU Project, http://neon.vb.cbs.nl/casc/

[3] Domingo-Ferrer, J., Torra, V., (2001), Disclosure Control Methods and Information Loss for Microdata, Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies, Elsevier Science, Pages 91-110, 2001.

[4] Domingo-Ferrer, J., Torra, V., (2001), A Quantitative Comparison of Disclosure Control Methods for Microdata, Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies, Elsevier Science, Pages 111-133, 2001.

[5] Domingo-Ferrer, J., and Mateo-Sanz, J. M., (2002),Practical data-oriented microaggregation for statistical disclosure control, Transactions on Knowledge and Data Engineering, IEEE, Volume 14, Pages 189–201.

[6] Hernandez, M., Stolfo, S., (1998), Real-world data is dirty: Data cleansing and the merge/purge problem. Data Mining and Knowledge Discovery, 1(2), 1998.

[7] Jaro, M. A., (1989), Advances in Record Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. Journal of the American Statistical Society, 84(406):414-420, 1989.

[8] Joint Photographic Experts Group (2001) Standard IS 10918-1 (ITU-T T.81), http://www.jpeg.org.

[9] Moore, R., (1996), Controlled Data Swapping Techniques for Masking Public Use Microdata Sets, U. S. Bureau of the Census (Unpublished manuscript).

[10] Sweeney, L., (2001), Information explosion, in Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies, eds. P. Doyle, J. I. Lane, J. M. Theeuwes and L. M. Zayatz, Elsevier, 43–74.

[11] Torra, V., Domingo-Ferrer, J., (2003), Record linkage methods for multidatabase data mining, Information Fusion in Data Mining, Springer, 101-132.

[12] Murphy, P., M., Aha, D. W., (1994), UCI Repository machine learning databases, http://www.ics.uci.edu/∼mlearn/MLRepository.html, Irvine, CA: University of California, Department of Information and Computer Science.

[13] Yager, R. R., (1993), Families of OWA operators, Fuzzy Sets and Systems, 59 125-148.

[14] Yager, R. R., (2004), Data Mining Using Granular Linguistic Summaries, in V. Torra, Information Fusion in Data Mining, Springer, 211-229.