Vicenç Torra; Yasunori Endo; Sadaaki Miyamoto

On the comparison of some fuzzy clustering methods for privacy preserving data mining: Towards the development of specific information loss measures

Kybernetika, Vol. 45 (2009), No. 3, 548--560

Persistent URL: http://dml.cz/dmlcz/140011

KYBERNETIKA — VOLUME 45 (2009), NUMBER 3, PAGES 548 – 560

ON THE COMPARISON OF SOME FUZZY CLUSTERING METHODS FOR PRIVACY PRESERVING DATA MINING: TOWARDS THE DEVELOPMENT OF SPECIFIC INFORMATION LOSS MEASURES

Vicenç Torra, Yasunori Endo and Sadaaki Miyamoto

Policy makers and researchers require raw data collected from agencies and companies for their analysis. Nevertheless, any transmission of data to third parties should satisfy some privacy requirements in order to avoid the disclosure of sensitive information.

The areas of privacy preserving data mining and statistical disclosure control develop mechanisms for ensuring data privacy. Masking methods are one of such mechanisms. With them, third parties can do computations with a limited risk of disclosure.

Disclosure risk and information loss measures have been developed in order to evaluate in which extent data is protected and in which extent data is perturbated. Most of the information loss measures currently existing in the literature are general purpose ones (i. e., not oriented to a particular application). In this work we develop cluster specific information loss measures (for fuzzy clustering). For this purpose we study how to compare the results of fuzzy clustering. I. e., how to compare fuzzy clusters.

Keywords: privacy preserving data mining, statistical disclosure control, fuzzy clustering, fuzzy c-means, fuzzy c-means with tolerance, comparison of fuzzy clusters

AMS Subject Classification: 68T05, 68T37,68T99

1. INTRODUCTION

The increase in computational power makes researchers and decision makers interested in the analysis of raw data. Also, companies, that generate a huge burden of data, often need to transfer these data to third parties for their analysis. As data usually contains sensitive information about people and corporations their release to third parties requires the application of mechanisms to ensure data privacy [9].

Two main approaches are being considered in the literature for ensuring data privacy: the cryptographic and the perturbative ones. The former approach [20] consists of, first, encrypting the data and, then, developing algorithms and applying them to the encrypted data. Finally, the results of the computation are decrypted by the data owners. As all the process is done using encrypted data, privacy is ensured. The latter approach [12] consists of perturbating the data (e. g., introducing some kind of noise in them). Then, data is published or transferred to third parties for

their analysis. This approach is simpler for the third parties (e. g., data analyzers) as standard algorithms can be applied to the published data. We only need here the method to perturbate the data. They are called masking methods and have been developed in the area of privacy preserving data mining (PPDM) and statistical disclosure control (SDC).

In this paper we focus on this perturbative approach. The difficulties of this approach are due to the fact that the selection of the *appropriate quantity of noise* is not a trivial problem and, that, for some methods, the finding of the optimal solution is an NP problem (this is the case of microaggregation, see [15]). This is so because not any perturbation is acceptable. Note that a large perturbation makes data useless for analysis and, in contrast, a small perturbation does not ensure data protection. In order to evaluate the appropriateness of a masking method (with respect to the noise that it is added to the data), a few measures have been developed. They are called [4, 18] information loss measures and disclosure risk measures.

Formally, information loss measures are to evaluate in what extent perturbated data is still useful for a posterior analysis and disclosure risk measures are to evaluate in what extent perturbated data compromises privacy. As information loss and disclosure risk are in contradiction, a good masking method is the one that permits to find a good trade-off (a good compromise) between information loss and disclosure risk.

As stated above, the goal of information loss measures is to assess the validity of the perturbated data for posterior analysis. Informally, what is expected is that the results of any analyses using the perturbated data are similar to the results of the same analyses using the original data. This validity naturally depends on the analysis to be performed. Nevertheless, it is often the case that the office in charge of the protection (e. g. a statistical office or a department in a company) is not well aware of the exact analysis that will be performed. Due to this, some general purpose information loss measures have been developed. These measures compare some basic statistics for the original and the protected files. These measures have permitted the construction of some comparisons (e. g. rankings) between different methods for data protection.

However, although such general purpose measures are, in general, of great help, no detailed analysis exists yet on whether they are also suitable for coping with the information loss occurring with specific data analysis (e. g., with usual data mining-like applications).

In this work we study the case of clustering algorithms being applied to the data sets. In particular, we study how to define information loss measures to evaluate the information loss related to clustering, and, more specifically, to fuzzy clustering. This requires the development of measures to compare fuzzy clusters in an arbitrary space. We propose two different measures that apply to fuzzy partitions constructed from cluster centers. Then, we apply our approach to two types of fuzzy clustering methods, evaluating the loss for a data protection mechanism. The paper reports the results obtained. Future work requires an extensive comparison of the information loss proposed here and the general purpose ones.

The structure of the paper is as follows. In Section 2 we briefly describe the information loss and disclosure risk measures as well as the fuzzy clustering methods. Then, in Section 3, we present our approach to develop clustering specific information loss measures, and we propose two measures for comparing fuzzy clusters. In Section 4 we describe the experiments done and the results obtained. The paper finishes with some conclusions and outlines some future work.

2. PRELIMINARIES

In this section we review some previous results that are needed later on in this paper. We start considering the measures to evaluate the performance of masking methods. Then, we will review the two fuzzy clustering algorithms that will be used later on in this paper.

2.1. Evaluating the performance of masking methods

As we have stated in the introduction, masking methods are the tools for perturbating an original file so that when the protected version of the file is released no disclosure of information arises. That is, the methods modify the original data so that intruders cannot obtain sensitive information about data respondents from the modified data (e. g., the salary of a particular person can not be found in the released data set).

Three types of measures have been developed to assess the quality of the released data. We have given in the introduction a short notice about them, we will give now some additional details. Although the paper is focused on information loss, we include the other two for completeness.

Information loss measures. They are designed to establish in which extent published data is still valid for carrying out the experiments planned on the original data. Information loss measures take into account the similarity between the original data set and the protected one, as well as the differences between the results that would be obtained with the original data set and the results that would be obtained from the masked data set.

Two types of information loss measures can be distinguished: specific ones and generic ones. Specific ones presume that we know the analyses that a data user will apply to the data. Then, under this premise, we can evaluate the similarity between the results obtained when applying the same analysis to either the original data file or the protected one (e. g., comparing the results of k-means over the original file and the protected one). In contrast, generic information loss measures do not presume this specific knowledge. I.e., the intended use of the data is not known. Then, some basic statistics are computed for both files, and the information loss measure is defined as their difference. For example, [3] considers differences on the means and covariances.

Disclosure risk measures. They are used to evaluate the extent in which the protected data ensures privacy. Among the different approaches that have been con-

sidered to measure the risk, we underline here a computational one that uses record linkage [16, 19]. This approach is based on the following scenario where an intruder intends to disclose some information from a published file. Formally, an intruder is assumed to have a data file with some pairs of non-confidential identifiers and quasi-identifiers. The identifiers are variables that uniquely identify an individual (e. g. passport number) and the quasi-identifiers are variables that when combined can be used to identify an individual (e. g. zip code, age, work). Then, the intruder using a record linkage method links his file with the published data file. The proportion (or number) of correct links obtained by the intruder is a measure of the risk. E. g., maximum risk is when she correctly links all the records.

Different record linkage methods have been developed some of them trying to take advantage of the masking method (e. g., [14]) or any other information available on the data [17].

Scores. These measures have been defined to summarize both information loss and disclosure risk. When these two measures are commensurate, it is possible just to combine them using the average.

Using these three measures it is possible to evaluate masking methods. In general, a good masking method needs a low information loss and a low disclosure risk. As these two measures are usually in contradiction, it is then expected a good trade-off between the two measures. The scores [4], as well as other tools, as the R-U maps [5, 6, 7] that permit to visualize the trade-off, permit to compare the different existing methods as well as the different parameterizations that exist for each of the methods.

2.2. Fuzzy clustering

Clustering methods, when applied to a set of data, typically build a partition of the data. In the case of fuzzy clustering, a fuzzy partition is built (instead of a crisp one). In this work we have considered two such methods for fuzzy clustering. They are the fuzzy c-means and the fuzzy c-means with tolerance. We will briefly describe these two methods below. See [8, 13] for details on the fuzzy c-means with tolerance. Fuzzy c-means, that was first proposed in [1], is described in most books on fuzzy sets and fuzzy clustering. See, e. g., [11].

In the description below we consider that we have a set of objects $X = \{x_1, \ldots, x_n\}$ and we want to build c clusters from this data. Such parameter c is expected to be given by the user. Then, the method builds the clusters which are represented by membership functions μ_{ik}, where μ_{ik} is the membership of the kth object (x_k) to the ith cluster.

In the case of fuzzy c-means with tolerance, we will have an additional parameter for each x_k. This parameter, κ_k, which is given by the user, is used to represent a maximum boundary for the error on each data. Formally, the fuzzy c-means with tolerance considers that data might contain errors and, due to this, each object x_k is permitted to be displaced from its position (within the tolerance limit κ_k) in the clustering process. $\kappa_k \geq 0$.

Both methods need an additional parameter, it is the value m. This value, that should satisfy $m \geq 1$ should also be given by the user. This parameter plays a central role. With values of m near to 1, solutions tend to be crisp (with the particular case that $m = 1$ corresponds to the crisp c-means, or k-means). Instead, larger values of m yield to clusters with increasing fuzziness in their boundaries.

Fuzzy c-means. Formally, the fuzzy c-means clustering algorithm constructs the fuzzy partition μ from X solving the following minimization problem. In the formulation of the problem, v_i is used to represent the cluster center, or centroid, of the ith cluster.

$$J_{\mathrm{FCM}}(\mu, V) = \left\{ \sum_{i=1}^{c} \sum_{k=1}^{n} (\mu_{ik})^m ||x_k - v_i||^2 \right\} \tag{1}$$

subject to the constraints $\mu_{ik} \in [0, 1]$ and $\sum_{i=1}^{c} \mu_{ik} = 1$ for all k.

We will denote by M the values μ that satisfy these constraints.

A (local) optimal solution of this problem is obtained using an iterative process that interleaves two steps. One that estimates the optimal membership functions of elements to clusters (when centroids are fixed) and another that estimates the centroids for each cluster (when membership functions are fixed). This iterative process is described in Algorithm 1.

Algorithm 1. Fuzzy c-means.

Step 1. Generate initial μ and V

Step 2. Solve $\min_{\mu \in M} J_{\mathrm{FCM}}(\mu, V)$ computing:

$$\mu_{ik} = \Big(\sum_{j=1}^{c} \Big(\frac{||x_k - v_i||^2}{||x_k - v_j||^2} \Big)^{\frac{1}{m-1}} \Big)^{-1}$$

Step 3. Solve $\min_V J_{\mathrm{FCM}}(\mu, V)$ computing:

$$v_i = \frac{\sum_{k=1}^{n} (\mu_{ik})^m x_k}{\sum_{k=1}^{n} (\mu_{ik})^m}$$

Step 4. If the solution does not converge, go to Step 2; otherwise, stop.

Fuzzy c-means with tolerance. This problem is solved in a similar way, in terms of an optimization problem. In this case, the function to optimize is the following one.

$$J_{\mathrm{FCMt}}(\mu, V) = \left\{ \sum_{i=1}^{c} \sum_{k=1}^{n} (\mu_{ik})^m ||x_k + \varepsilon_k - v_i||^2 \right\} \tag{2}$$

subject to the constraints $\mu_{ik} \in [0, 1]$ and $\sum_{i=1}^{c} \mu_{ik} = 1$ for all k, and with $||\varepsilon_k||^2 \leq ||\kappa_k||^2$.

This problem is solved in a way similar to the optimization problem given for the fuzzy c-means. That is, an iterative algorithm that interleaves the calculations of μ and of v_i. In addition, in the case of the fuzzy c-means with tolerance we need to compute the ε_k. As for μ and v_i, an approximation is computed in each step. The iterative process is described in Algorithm 2. As in the case of the fuzzy c-means, a (local) minimum of Equation (2) is found.

Algorithm 2. Fuzzy c-means with tolerance.

Step 1. Generate initial μ and V

Step 2. Solve $\min_{\mu \in M} J_{\mathrm{FCMt}}(\mu, E, V)$ computing:

$$\mu_{ik} = \Big(\sum_{j=1}^{c} \Big(\frac{||x_k + \varepsilon_k - v_i||^2}{||x_k + \varepsilon_k - v_j||^2} \Big)^{\frac{1}{m-1}} \Big)^{-1}$$

Step 3. Solve $\min_{\varepsilon \leq \kappa} J_{\mathrm{FCMt}}(\mu, E, V)$ computing:

$$\varepsilon_k = -\alpha_k \sum_{i=1}^{c} \mu_{ik}^m (x_k - v_i)$$

where

$$\alpha_k = \min \Big(\frac{\kappa_k}{|| \sum_{i=1}^{c} \mu_{ik}^m (x_k - v_i)||}, \frac{1}{\sum_{i=1}^{c} \mu_{ik}^m} \Big)$$

Step 4. Solve $\min_V J_{\mathrm{FCMt}}(\mu, E, V)$ computing:

$$v_i = \frac{\sum_{k=1}^{n} (\mu_{ik})^m x_k}{\sum_{k=1}^{n} (\mu_{ik})^m}$$

Step 5. If the solution does not converge, go to Step 2; otherwise, stop

3. THE COMPARISON OF FUZZY CLUSTERS AND THE DEVELOPMENT OF SPECIFIC INFORMATION LOSS MEASURES

The development of specific information loss measures when the intended use is fuzzy clustering needs the definition of measures for comparing fuzzy clusters. For this purpose, we consider the following informal definition for cluster specific information loss measures.

Definition 1. Let X be the original data set, and let X' be the protected data set. Let cl be a clustering method that given a data set returns a set of clusters. Then, a function $\mathrm{CS} - \mathrm{IL}$ is a specific information loss measure for fuzzy clusters (cluster specific information loss measure) when the more dissimilar is $\mathrm{cl}(X)$ to $\mathrm{cl}(X')$, the larger is $\mathrm{CS} - \mathrm{IL}(\mathrm{cl}(X), \mathrm{cl}(X'))$.

Naturally, it is expected that $\mathrm{CS} - \mathrm{IL}(\mathrm{cl}(X), \mathrm{cl}(X)) = 0$ holds for all data sets X.

Any measure of this form permits us to analyze the results of any clustering algorithm with respect to privacy. Nevertheless, the cornerstone is how to define the function CS − IL.

Formally, this function can be seen as a distance between two clustering results. In our case, we have that the clustering methods are fuzzy clustering ones and, thus, the result is a fuzzy cluster. Therefore, we should consider distances between fuzzy clusters or between fuzzy partitions. That is, we should consider distances that apply to two sets of fuzzy clusters. In the two fuzzy clustering methods considered, the fuzzy clusters are defined in terms of their single cluster center. We will take advantage of this information when defining the distance.

Formally, we have considered two alternative definitions. They are defined below.

Distance between cluster centers. That is, given two sets of fuzzy clusters A and B, we compare each cluster center in A with each cluster center in B. Then, we assign each cluster in A to a cluster in B. Finally, we compute the distance between the assigned clusters and the whole distance is its summation. Formally, let $a \in A$ denote the cluster centers in A and $b \in B$ denote the cluster centers in B, then we compute $d(a_i, b_j)$ for all pairs of a, b in A, B. Let π be an assignment of a_i in b_j, then the distance between the two sets of clusters is

$$d_1(A, B) = \sum_{i=1,\ldots,c} (a_i - b_{\pi(i)})^2.$$

We use an eager method to construct π. Formally, each a_i is assigned to the nearest record in B. So, we define $\pi(i)$ as follows: $\pi(i) = \arg\min_j d(a_i, b_j)$.

Distance based on memberships. The previous distance considers only the clusters but neither any information on the number of objects that have been clustered (the *size* of the cluster), their position or membership. To avoid this drawback, we have defined another distance that takes all this into account and also whether objects are clustered in the same cluster by the clustering methods. The alternative distance considers the differences between membership functions. Formally, let μ_{ik}^A be the membership of the kth object to the ith cluster in the set of clusters A, and let μ_{ik}^B be the corresponding membership in the set of clusters B; then, the distance between the two set of clusters is

$$\sum_{k=1}^{n} \sum_{i=1}^{c} \left(\mu_{ik}^A - \mu_{ik}^B\right)^2$$

Note that the actual computation of this distance needs to find a correct alignment between the clusters in A and B. That is, we need that i denotes the same cluster for both A and B. In our particular application, we use here the π constructed for computing the previous distance. Therefore, the actual distance is as follows:

$$d_2(A, B) = \sum_{k=1}^{n} \sum_{i=1,\ldots,c} \left(\mu_{ik}^A - \mu_{\pi(i)k}^B\right)^2$$

Naturally, these two distances satisfy the property that $d(A, B) = 0$ if and only if $A = B$.

Note that while the second distance takes additional information into account, its computational cost is larger. Given c clusters, and n data in a t dimensional space, the cost of computing d_1 is $O(c \cdot t)$ while the cost of computing d_2 is $O(c \cdot t \cdot n)$.

Using these two definitions, $\mathrm{CS} - \mathrm{IL} = d_1$ or $\mathrm{CS} - \mathrm{IL} = d_2$, we have two cluster specific information loss measures.

4. EXPERIMENTS AND ANALYSIS

To evaluate our proposal, we have considered the evaluation of a set of data files protected using a particular masking method with several parameterizations.

It has to be said that before applying the clustering algorithm, the two files have been normalized. The transformation for variable x was to replace it by a new one x' that corresponds to the previous one x minus the mean and divided by its deviation.

The original data file. The experiments were carried out using the same original file already used by a few researchers (see e. g. [4, 10, 19]) to evaluate information loss (generic measures), and disclosure risk. The file consists of 1080 numerical records described in terms of 13 variables. The file is described in detail in [2].

Two sets of experiments were carried out. One set corresponds to the case of using the whole file and the other set corresponds to the case of using only two of the variables (the two first columns in the original file).

The protection method used is noise addition. This method consists of the perturbation of the original file adding noise. In short, noise is added to each variable of the file using a $N(0, ps)$ where s is the standard deviation of the variable in the original file, and p is the parameter. The following values of p were used: 0.01, 0.02, 0.06, 0.08 and up to 0.2 with 0.02 increments.

Then, the methods proposed are valid for comparing partitions as soon as the distance is monotonic with respect to the error introduced in the data. This is basically the case of the measures here. However, the computation of the distance faced a set of difficulties. In the rest of the section we revise the methodology used to validate the distance, the difficulties encountered as well as the results obtained.

4.1. Dealing with local minima

The fuzzy clustering algorithms under consideration in this work can lead to local minima, and the clusters obtained depend on the initializations. Due to this, a few different approaches have been considered in the initialization of each execution. Formally, we have initialized the algorithms assigning some of the data being clustered to the set of cluster centers. Such assignment has been done at random, and different random seeds have been used in different executions. This results into different clusters for the same data set and the same parameters.

Then, to avoid the computation of the distance using a local minimum instead of a global one, we have executed each method 20 times. Then, among the 20 solutions

found (the 20 *slightly different* fuzzy partitions obtained by the algorithms), we have selected the one with minimal objective function.

Table 1 shows that the problem of local minima was a critical point of our approach. This is the case, even when we are considering 20 executions. Note that with a noise equal to 0.2, we have two cases with $\kappa_1 = 0$ and $\kappa_2 = 0.1$, one that has $d_2 = 0.6$ and the other with $d_2 = 49.5$. This latter case corresponds to an objective function with a value of 3000 while the first one corresponds to an objective function with a value of 2973.0. So, the result with the largest distance corresponds to a local optimum. Note also that such results cause a disruption on the monotonicity of d_2 for the FCM with tolerance with this particular parameterization. Another case of convergence to different local minima has been found when clustering the original file with $\kappa_1 = 0.1$. We obtained two different sets of clusters in two different experiments with $\kappa_2 = 0.1$ and $\kappa_2 = 0.2$. One resulted with an objective function equal to 2997 and the other equal to 2969. Both cases should have lead to the same optimal solution as κ_2 does not influence the result of the original data set (in both cases $\kappa_1 = 0.1$). Note that such divergences correspond to the optimal solutions found after 20 executions. Additional executions could be considered, but after some trials we heuristically found that 20 was an acceptable trade-off between finding a better optimal solution and the high computational cost of considering additional executions.

Table 1. Distances d_1 and d_2 between the clusters originated from the original and the protected file for different values of noise and using FCM with tolerance as the clustering algorithm. κ_1 corresponds to the κ used with the original file and κ_2 corresponds to the κ used with the protected file. The values achieved for the objective function are also included for each protected file. The optimal value found for the original file were 2826 in the three cases with $\kappa_1 = 0$, 2997 in the case of $\kappa_1 = 0.1$ but $\kappa_2 = 0.1$ and 2969 in the case of $\kappa_1 = 0.1$ but $\kappa_2 = 0.2$. Note that in this latter two cases, there is a single optimal solution.

κ_1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.1	0.1	0.1	0.1
κ_2	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.1	0.1	0.1	0.2	0.2	0.2
Noise	d_1	d_2	$O.F.$	d_1	d_2	$O.F.$	d_1	d_2	$O.F.$	d_1	d_2	$O.F.$	d_1	d_2	$O.F.$
0.0	4.1	80.0	2972.0	0.7	0.4	2969.0	1.3	1.5	3117.0	3.4	19.1	2969.0	0.7	0.4	3117.0
0.1	0.7	0.4	2970.0	0.7	0.4	2970.0	1.3	1.6	3118.0	3.2	19.0	2970.0	0.7	0.4	3118.0
0.2	0.7	0.6	2973.0	4.3	49.5	3000.0	1.4	1.9	3121.0	0.1	0.2	3000.0	0.7	0.7	3121.0
0.4	0.7	0.9	2980.0	0.6	1.0	2980.0	1.2	1.3	3128.0	10.0	23.0	3025.0	0.7	0.9	3128.0
0.6	–	–	–	0.7	2.0	2988.0	1.3	3.0	3137.0	3.4	20.8	3031.0	0.7	1.9	3137.0
0.8	–	–	–	0.8	3.9	3008.0	1.3	4.4	3158.0	3.7	20.9	3008.0	0.8	3.7	3158.0
1.0	–	–	–	1.1	8.4	3033.0	1.6	8.1	3185.0	3.9	25.5	3033.0	1.1	8.2	3185.0
1.2	–	–	–	1.2	12.1	3051.0	5.7	51.2	3226.0	3.8	32.3	3051.0	1.5	14.4	3280.0
1.4	–	–	–	1.0	12.7	3066.0	1.4	1.2	3220.0	3.8	28.3	3066.0	1.2	12.2	3220.0
1.6	–	–	–	1.4	12.3	3111.0	1.5	1.8	3268.0	1.7	33.1	3111.0	3.7	91.7	3271.0
1.8	–	–	–	1.5	22.2	3141.0	1.8	20.3	3229.0	4.3	38.6	3141.0	1.4	20.7	3299.0
2.0	–	–	–	7.1	110.3	3185.0	3.1	111.4	3327.0	4.4	97.4	3185.0	7.0	104.8	3348.0

4.2. Results

Table 2 includes the results for the FCM when all 13 variables were used. The table includes the measures for the distance as well as the objective functions. Two executions are presented, both with 10 clusters. Table 3 presents the results for the

same algorithm when only 2 variables are used. In this case the convergence of the algorithm is much better. Two executions are given one with 10 clusters and the other with 20. Table 1, already discussed above, presents the results for the FCM with tolerance. 13 variables were used in the experiments. In this case, several executions with different values of κ_1 and κ_2 are included.

Table 2. Distances d_1 and d_2 between the clusters originated from the original and the protected file for different values of noise and using the fuzzy c-means (FCM) as the clustering algorithm. Two different executions, both computing 10 clusters for each file. The values achieved for the objective function are also included for each protected file. The optimal value found for the original file was 2851 in the first execution (left) and 2829 in the second one (right).

Noise	d_1	d_2	$O.F.$	d_1	d_2	$O.F.$
0.0	3.21	40.73	2826.0	3.93	91.3	2826
0.1	3.21	40.67	2827.0	3.97	91.45	2827
0.2	3.17	40.86	2829.0	3.94	90.89	2829
0.4	0.32	0.92	2859.0	4.07	93.05	2835
0.6	3.28	42.09	2844.0	6.92	113.76	2867
0.8	3.48	43.48	2862.0	4.19	91.53	2862
1.0	3.55	48.87	2886.0	4.37	99.33	2886
1.2	2.24	55.56	2908.0	2.75	68.04	2903
1.4	1.44	18.35	2935.0	4.53	99.53	2918
1.6	2.27	36.83	2978.0	6.98	103.84	2978
1.8	2.71	45.59	3006.0	4.68	99.20	2989
2.0	4.24	96.87	3028.0	2.70	31.17	3013

Table 3. Distances d_1 and d_2 between the clusters originated from the original and the protected file for different values of noise and using the fuzzy c-means (FCM) as the clustering algorithm. Executions with only 2 variables. The results of two different executions, the first one with 10 clusters and the second one with 20 clusters are given. The values achieved for the objective function are also included for each protected file. The optimal value found for the original file was 225.26 in the first execution, 10 clusters, (left) and 107.06 in the second one, 20 clusters, (right).

	10 clusters			20 clusters		
Noise	d_1	d_2	$O.F.$	d_1	d_2	$O.F.$
0.0	5.00E-09	1.00E-15	225.26	2.86	208.90	107.19
0.1	0.10	0.92	225.67	3.03	157.10	107.20
0.2	0.08	1.74	225.02	0.69	13.46	107.21
0.4	0.21	8.45	224.63	1.80	113.00	106.97
0.6	0.49	25.27	225.45	2.15	73.73	106.67
0.8	3.16	217.38	224.85	3.22	214.29	108.47
1.0	1.29	73.13	226.53	2.80	224.25	108.66
1.2	3.80	252.37	225.21	3.96	259.46	109.11
1.4	0.66	80.99	227.00	4.45	318.17	109.61
1.6	3.13	257.35	228.43	2.92	337.55	112.14
1.8	3.20	315.55	230.97	5.11	454.07	111.77
2.0	3.25	313.78	231.82	5.31	510.52	110.00

Analysing the results we can point out the following.

Comparison between clustering methods. The FCM with tolerance shows a better performance with respect to the monotonicity of the distance with respect

to the error than the standard FCM. So, in this sense the FCM with tolerance has more tolerance to error.

Nevertheless, it is important to point out that such results heavily depend on the particular κ selected for the experiments. The selection of this value has required several experiments, as initial (and larger) values were rather inadequate (non monotonic results were obtained).

Therefore, in this setting, the selection of an appropriate κ is a critical aspect for the FCM with tolerance.

Comparison between the two measures d_1 and d_2. Distance d_2 seems to be sounder than d_1. The problems with the convergence of the clustering method causes a disruption of the monotonicity for both measures. Nevertheless, the number of distorted points seems to be larger for d_1. So, d_2 is more resilient to errors. Besides of that, the correlation coefficient between d_2 and the amount of error is larger than the same coefficient for d_1. In addition, the range of d_2 is larger, which helps in the comparison of the results. Besides of that, d_2 seems to exploit better the information available as it uses additional information corresponding to the elements being clustered. So, in fact, the sizes of the clusters are taken into account in d_2 (not only the cluster centers) as in d_1.

FCM with tolerance. The results for this method are adequate when we used $\kappa_1 \leq \kappa_2$. That is, when the error of the second file is defined to be larger than the error of the first file. This condition seems natural in our context, where the second file is the first file with some error.

5. CONCLUSIONS AND FUTURE WORK

In this work we have considered the problem of comparing fuzzy partitions, and proposed two methods for this purpose, with the final goal of developing specific information loss measures. We have studied the two methods empirically in the context of privacy preserving data mining. We have described the experiments performed. Their aim was to validate the distances proposed. We have outlined the main difficulties we have encountered.

The main one was the difficulty of reaching the global optima before comparing the partitions. Due to the fact that clustering methods often tend to stop at local optima, the comparisons included a few bad results (e. g. non zero distance between two fuzzy partitions obtained with the same clustering method). This was due to the comparison of the results of different local optima.

All in all, we have shown that the measures have an appropriate performance as they are monotonic with respect to the error introduced to the data. Of the two proposed distances, the distance d_2 has a slightly better performance than d_1.

A few questions are left open in this work and require some future research. First, in relation to the FCM with tolerance, additional experiments are needed to confirm that the use of $\kappa_1 = \kappa_2$ is not meaningful in our context. For example, considering some additional experiments and doing experiments with other files. Also, the selection of the right κ is a challenging problem. We used here an heuristic

approach for selecting such values. Further research is needed to find the appropriate values for a given problem.

Other future work consists of considering the extension of our approach for comparing partitions to other situations where the fuzzy set of a cluster is not solely based on a single cluster center.

In addition to that, we plan to apply our approach to a set of different masking methods (with different parameterizations) in order to compare their effect with respect to fuzzy cluster methods. At this point, we will compare the results of generic information loss measures with the specific ones defined here. In this analysis, fuzzy clustering methods other than the ones included here can also considered.

ACKNOWLEDGEMENTS

Partial support by the Spanish MEC (projects ARES – CONSOLIDER INGENIO 2010 CSD2007-00004 – and eAEGIS – TSI2007-65406-C03-02) and grant "Salvador de Madariaga" PR2007-0122 is acknowledged.

(Received September 25, 2007.)

REFERENCES

[1] J. C. Bezdek: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York 1981.

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

[3] J. Domingo-Ferrer and V. Torra: Disclosure control methods and information loss for microdata. In: Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies (P. Doyle, J. I. Lane, J. J. M. Theeuwes, and L. M. Zayatz, eds.), Elsevier 2001, pp. 91–110,

[4] J. Domingo-Ferrer and V. Torra: A quantitative comparison of disclosure control methods for microdata. In: Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies (P. Doyle, J. I. Lane, J. J. M. Theeuwes, and L. M. Zayatz, eds.), Elsevier 2001, pp. 111–133.

[5] G. Duncan, S. Fienberg, R. Krishnam, R. Padman, and S. Roehrig: Disclosure limitation methods and information loss for tabular data. In: Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies (P. Doyle, J. I. Lane, J. J. M. Theeuwes, and L. M. Zayatz, eds.), Elsevier 2001, pp. 135–166.

[6] G. Duncan, S. Keller-McNulty, and S. Stokes: Disclosure Risk vs. Data Utility: The R-U Confidentiality Map. Technical Report No. 121 of National Institute of Statistical Sciences 2001, www.niss.org.

[7] G. Duncan, S. Keller-McNulty, and S. Stokes: Database Security and Confidentiality: Examining Disclosure Risk vs. Data Utility Through the R-U Confidentiality Map. Technical Report No. 142 of National Institute of Statistical Sciences 2004, www.niss.org.

[8] Y. Hasegawa, Y. Endo, Y. Hamasuna, and S. Miyamoto: Fuzzy c-means for data with tolerance defined as hyper-rectangle. In: Proc. MDAI 2007 (Lecture Notes in Artificial Intelligence 4617), pp. 237–248.

[9] J. Lane, P. Heus, and T. Mulcahy: Data access in a cyber world: Making use of cyberinfrastructure. Trans. Data Privacy *1* (2008), 2–16.

[10] P. Medrano-Gracia, J. Pont-Tuset, J. Nin, and V. Muntés-Mulero: Ordered data set vectorization for linear regression on data privacy. In: Proc. MDAI 2007 (Lecture Notes in Artificial Intelligence 4617), Springer, Berlin 2007, pp. 361–372.

[11] S. Miyamoto and K. Umayahara: Methods in gard and fuzzy clustering. In: Soft Computing and Human-Centered Machines (Z.-Q. Liu and S. Miyamoto, eds.), Springer, Tokyo 2000, 85–129.

[12] S. Mukherjee, Z. Chen, and A. Gangopadhyay: A privacy-preserving technique for Euclidean distance-based mining algorithms using Fourier-related transforms. The VLDB Journal *15* (2006), 293–315.

[13] R. Murata, Y. Endo, H. Haruyama, and S. Miyamoto: On fuzzy c-means for data with tolerance. J. Advanced Computational Intelligence and Intelligent Informatics *10* (2006), 5, 673–681.

[14] J. Nin, J. Herranz, and V. Torra: Rethinking rank swapping to decrease disclosure risk. Data and Knowledge Engrg. *64* (2008), 1, 346–364.

[15] A. Oganian and J. Domingo-Ferrer: On the complexity of optimal microaggregation for statistical disclosure control. Statistical J. United Nations Economic Commission for Europe *18* (2000), 4, 345–354.

[16] V. Torra and J. Domingo-Ferrer: Record linkage methods for multidatabase data mining. In: Information Fusion in Data Mining (V. Torra, ed.), Springer 2003, pp. 101–132.

[17] V. Torra and J. Nin: (2008) Record linkage for database integration using fuzzy integrals. Internat. J. Intel. Systems *23* (2008), 715–734.

[18] M. Trottini: Decision Models for Data Disclosure Limitation. Ph.D. Dissertation, Carnegie Mellon University 2003, http://www.niss.org/dgii/TR/Thesis-Trottini-final.pdf.

[19] W. E. Yancey, W. E. Winkler, and R. H. Creecy: Disclosure risk assessment in perturbative microdata protection. In: Inference Control in Statistical Databases 2002 (Lecture Notes in Computer Science 2316), Springer, Berlin 2003, pp. 135–152.

[20] A. C. Yao: Protocols for secure computations. In: Proc. 23rd IEEE Symposium on Foundations of Computer Science, Chicago 1982, pp. 160–164.

Vicenç Torra, IIIA, Artificial Intelligence Research Institute, CSIC, Spanish Council for Scientific Research, Campus U.A.B. s/n, 08193 Bellaterra, Catalonia. Spain.
e-mail: vtorra@iiia.csic.es

Yasunori Endo and Sadaaki Miyamoto, Department of Risk Engineering, School of Systems and Information Engineering, University of Tsukuba, Ibaraki 305-8573. Japan.
e-mails: endo@risk.tsukuba.ac.jp, miyamoto@risk.tsukuba.ac.jp