PRAM Optimization Using an Evolutionary Algorithm

Jordi Marés and Vicenç Torra

IIIA - Institut d'Investigació en Intelligència Artificial
CSIC - Consejo Superior de Investigaciones Científicas
Campus de Bellaterra, 08193 Bellaterra, Catalonia, Spain
{jmares,vtorra}@iiia.csic.es

Abstract. PRAM (Post Randomization Method) was introduced in 1997 but it is still one of the least used methods in statistical categorical data protection. This fact is because of the difficulty to obtain a good transition matrix in order to obtain a good protection. In this paper, we describe how to obtain a better protection using an evolutionary algorithm with integrated information loss and disclosure risk measures to find the best matrix. We also provide experiments using a real dataset of 1000 records in order to empirically evaluate the application of this technique.

Keywords: Information Privacy and Security, Evolutionary Algorithms, Post Randomization Method, Information Loss, Disclosure Risk.

1 Introduction

As there are continuously more and more public dataset availables for analyses, more reliable protection methods are needed to ensure the privacy of the data. An obvious measure to maintain the privacy of the individuals is to replace or suppress any explicit identifier. However the application of this measure alone may be insufficient. Linking groups of records between different data sets might reveal the identity of individuals and involve an unauthorized disclosure of sensitive information [7].

When applying Statistical Disclosure Control (SDC) methods, one has to deal with two competing goals: the microdata file has to be safe enough to guarantee the protection of individual respondents but at the same time the loss of information should not be too large. The discussion for this can be found in [3].

In our case, we focus on categorical data which has more limited actions to perform when protecting because arithmetic operations are not allowed here. Then, the only actions that can be performed with categorical data are exchange of categories by others that already exist, and generalization of some categories into a newer ones, so having only two different actions for protection makes it a difficult task.

J. Domingo-Ferrer and E. Magkos (Eds.): PSD 2010, LNCS 6344, pp. 97–106, 2010.

The Post Randomization Method (PRAM) was introduced in [6] as a method for categorical variables disclosure control in microdata files. In [4] and [5], the method and some of its implications were discussed in more detail. However, the PRAM method is still one of the least used statistical categorical data protection methods because of the difficulty to obtain a good transition matrix in order to obtain a good protection. This was demonstrated in the experiments done in [8] where PRAM protections were the ones with the worse scores.

In this paper, we present a new way to find a good transition matrix which provides us with a good categorical microdata protection, using an evolutionary algorithm applied to an initial mask. The method is bootstrapped with the PRAM matrices decribed in [4,5,6,8].

The reminder of the paper is organized as follows. A brief explanation of the Post Randomization Method and a description of the types of matrices we have used is provided in Section 2, followed by an outline of evolutionary algorithms and a description of our proposed algorithm in Section 3. Experimental results are given in Section 4.

2 The Post Randomization Method (PRAM)

PRAM is a probabilistic, perturbative method for disclosure protection of categorical variables.

This method is based on changing the scores on some categorical variables for certain records to a different score according to a prescribed Markov matrix. This matrix contains a row for each possible value of each variable to be protected, and each row contains the probabilities of changing the original data value to any other value. These probability matrices are very important in order to obtain a good protection.

2.1 PRAM Matrices

There are different ways to define the Markov matrices in the literature. We discuss here two of the approaches, which are the most commonly used. In the discussion we understand p_{kl} as the probability of changing a value k to a value l. Then, $\sum_{l=1}^{n} p_{kl} = 1$, where n is the number of categories. We choosed two types of matrices design to work. The first type is a fully-filled matrix with the off-diagonal elements depending on the corresponding frequencies in the original microdata file. This approach has been used in [2]. Formally, the probability p_{kl} for $k \neq l$ is defined by

$$p_{kl} = \frac{(1 - p_{kk})(\sum_{i=1}^{n} T_\xi(i) - T_\xi(k) - T_\xi(l)}{(n-2)(\sum_{i=1}^{n} T_\xi(i) - T_\xi(k))} \tag{1}$$

where $T_\xi(i)$ is the frequency of the category i inside the original dataset for the actual variable. In the approach p_{kk} is left as constant, that is, $p_{kk} = p$ for all k. The key point of this equation is that it assigns the higher exchange probabilities

to the categories with less frequency. In this way, the resultant dataset has more confusion.

The second type is a fully-filled matrix with the diagonal elements depending on the corresponding frequencies in the original microdata file. This approach has been used in [8]. In this case the row values are determined by the following expressions:

$$p_{kk} = 1 - (\theta T_\xi(K)/T_\xi(k)) \tag{2}$$

for $k = 1, \ldots, n$ and, then,

$$p_{kl} = \frac{1 - p_{kk}}{n - 1} \tag{3}$$

for $k \neq l$, where $T_\xi(K)$ is the lower value frequency higher than zero, and θ is a parameter in $[0, 1]$. In our experiments we have used $\theta = 0.7$.

2.2 Analytical Measures

There exist two measures to evaluate the performance of a protection method: the information loss and the disclosure risk.

Information loss is known as the quantity of harm that is inflicted to the data by a given masking method. This measure is small when the analytic structure of the masked dataset is very similar to the structure of the original dataset, so, the motivation for preserving the structure of the dataset is to ensure that the masked dataset will be analytically valid and interesting.

Assessment of the quality of a protection method cannot be limited to information loss because disclosure risk has also to be measured. Disclosure risk is known as the quantity of original data that can be obtained by an intruder from the masked dataset. This measure is small when the masked dataset values are very different to the original values.

The problem here is that both measures are inversely related so the higher information loss the lower disclosure risk, and the inverse. In order to perform a good protection there must be a minimised combination of both measures.

3 Outline of Evolutionary Algorithms

Evolutionary algorithms are stochastic optimization and search methods that mimic the metaphor of natural biological evolution. Those algorithms operate on a population of potential solutions P so, formally, the sequence $P(0), P(1), \ldots, P(t)$ is called an *evolution* of $P(0)$.

The population is mantained over all the t generations, where every individual $X_i' \in P(t)$ is related to a potential solution for the given problem. In order to guide the individuals through the generations any "fitness" measure for *evaluation* is needed. The search for new potential solutions is performed selecting some of the individuals and *altering* them using operators such as mutation and

crossover. These operators generate an offspring of new individuals from previous generations. Surviving individuals are going evaluated again, and the process is repeated until some stopping criterion is reached.

Using this basic scheme there are some settings that can be adapted to the problem like how to represent and evaluate the individuals, the stopping criteria, how are the individuals selected and altered from generation to generation.

Alg. 1 shows a pseudo-code summarizing our algorithm which is a generic evolutionary algorithm with some particularities that are described below.

Algorithm 1. Evolutionary Algorithm to Enhance PRAM Matrices

```
Input: P(0) = X initial population
Output: P(t) = X' final population
t ⇐ 0
evaluate(P(0))
while stopping(P(t)) ≠ true; do
  alter ⇐randomly choose between mutation and cross
  if alter by mutation then
    X' ⇐ mutation(X)
  else
    X' ⇐ cross(X)
  end if
  evaluate(X, X')
  t ⇐ t + 1
end while
return P(t)
```

Next subsections describe the key points of our evolutionary algorithm such as individual representation, genetic operators and evaluation function. We will also discuss how information loss and disclosure risk are integrated within an evolutionary algorithm.

3.1 Genotype Encoding

Usually the initial matrix contain values with a lot of decimals so, in order to simplify the values, all the values are multiplied by 100 and only the integer part of the value will be kept for the encoding.

Encoding of the individual X is done value by value transforming them into its Gray code representation. The decision of working with Gray-coded values was taken to avoid abrupt value changes when any bit is altered. Discussion of gray coding can be found in [1].

A complete file encoding example is shown in Fig. 1. The example includes all the steps required during the whole encoding process.

3.2 Genetic Operators

Our proposed algorithm uses two basic operators: crossover and mutation.

0.185	0.283	0.532
0.609	0.089	0.302
0.002	0.277	0.721

PRAM matrix.

185	283	532
609	89	302
2	277	721

PRAM matrix in integers mode.

0010111001	0100011011	1000010100
1001100001	0001011001	0100101110
0000000010	0100010101	1011010001

Binary matrix.

0011100101	0110010110	1100011110
1101010001	0001110101	0110111001
0000000011	0110011111	1110111001

Gray-coded genome matrix.

Fig. 1. Example of genotype encoding

Crossover of the individual X is performed by swapping two ranges of values inside the individual as follows. Take two value positions $\{s, r\}$ at random, and consider that the two values at this position are $x_s \in X$ and $x_r \in X$. Generate a random number m to indicate the length of the ranges. This number must be in the range $[0, min(length(X) - s, length(X) - r, |s - r|)]$, where length($X$) is the total number of values inside the individual X, and —— is the absolute value operator. Then the ranges $[x_s, x_{s+m}]$ and $[x_r, x_{r+m}]$ are swapped obtaining a new individual. For example, having $s < r$ and $X = \{x_1, \ldots, x_n\}$ the new individual will be $X' = \{x_1, \ldots, x_r, \ldots, x_{r+m}, \ldots, x_s, \ldots, x_{s+m}, \ldots, x_n\}$.

Mutation is performed by a simple value mutation as follows. Take a random value of the individual X and consider that the value at this position is x_i with $genome(x_i) = b_j b_{j-1} \ldots b_1$. Choose a bit position k at random, such that $1 \leq k \leq j$. Then a new individual is obtained just by replacing the bit b_k by its negation counterpart, $b'_k = not(b_k)$.

We decided to use the value 0.5 for both crossover rate and mutation rate in order to have mostly the same number of operations performed by each one. A random value (alter) between 0 and 1 decides the operation to perform, using 0.5 as a delimiter.

3.3 Fitness Function

During the evaluation of the PRAM matrix two steps are needed. First of all the matrix has to be used for an implementation of the PRAM method to protect the original file, and then the protected file needs to be used into the evaluator software in order to obtain the results of information loss and disclosure risk related to the matrix.

As we have two measures to minimize, this is a multi-objective optimization problem. To solve this we chose a multi-objective optimization method called Objective Weighting which allows us to combine both measures applying an individual weight to each one. We wanted to give the same importance to both Disclosure Risk (DR) and Information Loss (IL) measures so both have $\frac{1}{2}$ as a weigth value. Then, the individual score can be obtained as follows:

$$Y = PRAM(X') \tag{4}$$

$$Score(X') = \frac{DR(Y) + IL(Y)}{2} \tag{5}$$

As the PRAM protection method takes some random decisions, the method generates different protected files with the same Markov matrix, and these different files will have different scores. In order to have a more robust score we compute 5 protected files for each individual (i.e., each Markov matrix) and we take the average of their scores as the final score. Formally:

$$FinalScore(X') = \frac{\sum_{i=1}^{5} Score(X')}{5} \tag{6}$$

In both mutation and crossover cases, during the evaluation, an elitism replacement strategy is followed which means that the new individual and the old one are compared and only the one with best score will be selected as the individual of the population for the next generation.

4 Experimental Results

To test and empirically evaluate our proposed method we have done several experiments protecting some attributes of a dataset and analising the evolution of the score in each one. The dataset we used is a U.S. Housing Survey of 1993 with 1000 records and 11 categorical attributes.

Here we are going to present, the results for the protection of the DEGREE attribute, which has 8 ordinal categories available, using the two types of PRAM matrices that have been described in section 2.1.

In these experiments, we are going to denote the matrix computed by equation (1) as $nF(p)$ where n is the size of the square matrix and p is the value for the elements in its diagonal, and the matrix computed by equations (2) and (3) as $nD(p)$ where n is the size of the square matrix and p is the value of the parameter θ. In our case, we have used matrices 8F(0.5) and 8D(0.7).

Figure 2 shows the evolution of the information loss, disclosure risk and score of the matrix $8D(0.7)$ over more than 1600 generations. It is easy to see that not all the measures have been decreased (indeed the information loss has increased!) but the adjust of the two measures has performed a very high decrease of the score. In this experiment, the score has decreased from 24.18 to 8.34 what represents almost the third part of the initial score. It can be also seen the effect of the evolutionary algorithm looking for the best adjustment of the measures increasing and decreasing them irregularly like the results around generation 200.

For the second experiment we have used matrix 8F(0.5) to protect the same attribute and we obtained the results shown in Figure 3. In this case, unlike the first experiment, all the measures have decreased forcing to decrease the final score too. This fact demonstrate again that the evolutionary algorithm does not

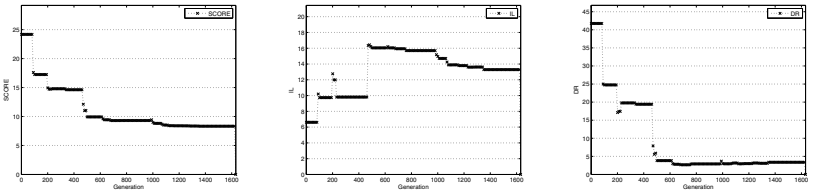

Fig. 2. Measures evolution for DEGREE attribute protection with $8D(0.7)$ matrix

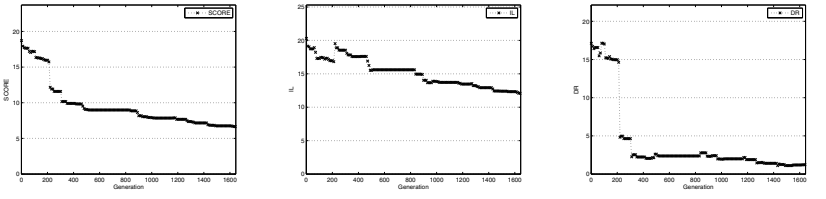

Fig. 3. Measures evolution for DEGREE attribute protection with $8F(0.5)$ matrix

follow any pattern in order to obtain the best result, it is just looking for the best option at the moment. In this experiment the fact of irregular variations of the measures also occurs, this can be observed in the figure within the first 200 generations. Using the $8F(0.5)$ matrix we obtained a decrement of the score from 18.71 to 6.65 what represents,like in the first experiment, alsmost the third part of the initial score.

A more detailed view of the results is given in Figure 4 where disaggregated measures are shown. The first one is the Interval Disclosure (ID) which decreases at the beginning and also from the 1600th generation, being mainly constant between those two decrements. The second measure is the Entropy-Based Information Loss (EBIL) which has a very similar behaviour. Finally the third measure is the Distance-Based Record Linkage (DBRL) which has a decrement at the beginning and after that is quite irregular but mantaining the range of values during all the generations. The rest of the measures that are not shown just mantain more or less their value.

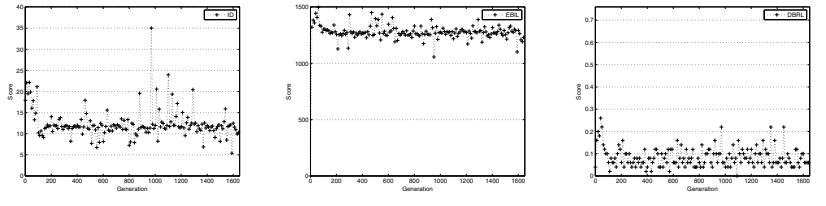

Fig. 4. Some disaggregated measures evolution

Table 1. Initial PRAM matrix 8D(0.7)

0.764	0.039	0.039	0.039	0.039	0.039	0.039	0.000
0.022	0.866	0.022	0.022	0.022	0.022	0.022	0.000
0.015	0.015	0.908	0.015	0.015	0.015	0.015	0.000
0.020	0.020	0.020	0.882	0.020	0.020	0.020	0.000
0.023	0.023	0.023	0.023	0.864	0.022	0.022	0.000
0.048	0.048	0.048	0.048	0.048	0.711	0.048	0.000
0.117	0.117	0.117	0.117	0.117	0.117	0.300	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000

We also want to comment which changes have been performed between the original matrices and the final ones. Table 1 shows the initial 8D(0.7) matrix. It can be seen that the diagonal has the higher values of each row, but all values are different because they are computed depending on the frequency of each category. The rest of the elements in each row have the same value except the last one of each row which we decided to left it as 0.000 because it corresponds to the change to a reserved category.

The final matrix is shown in Table 2. Here we can see that the initial matrix structure has totally changed, so there is only one row with the highest value placed in the diagonal and not all the rows have the same off-diagonal values for all changes. An important point is that after the evolutionary algorithm is applied, there are three rows with their higher value at column 3 what means that almost all appearences of the categories corresponding to those rows will be changed to a single one (i.e., to the third category).

Table 2. Final PRAM matrix 8D(0.7)

0.352	0.026	0.005	0.005	0.173	0.429	0.010	0.000
0.026	0.026	0.026	0.026	0.846	0.026	0.026	0.000
0.057	0.057	0.057	0.739	0.011	0.023	0.057	0.000
0.094	0.774	0.019	0.019	0.019	0.057	0.019	0.000
0.039	0.016	0.921	0.003	0.003	0.014	0.005	0.000
0.006	0.006	0.814	0.006	0.047	0.019	0.102	0.000
0.469	0.000	0.055	0.055	0.055	0.227	0.139	0.000
0.062	0.062	0.430	0.000	0.000	0.000	0.033	0.413

The case of matrix $8F(0.5)$ is more or less the same. Table 3 shows the initial matrix. In this case it can be seen that all diagonal values have the same value, but the off-diagonal ones are all different in each row. This is because of the dependecy on the categories frequency when the matrix is computed. Note that, in this case, the higher values are still in the diagonal. For this matrix we wanted to use the reserved category in order to obtain a more different matrix, so the last column values are different than zero.

Finally, in Table 4 the final matrix is shown. Like in the 8D(0.7) matrix, this one also has the highest values outside of the diagonal. In addition, there are also two groups of two rows (i.e., categories) that are changed to different single categories for each group, obtaining a behaviour like in the case of 8D(0.7) matrix.

Table 3. Initial PRAM matrix 8F(0.5)

0.500	0.067	0.060	0.065	0.067	0.076	0.080	0.083
0.073	0.500	0.058	0.064	0.066	0.075	0.080	0.083
0.072	0.064	0.500	0.062	0.064	0.074	0.080	0.083
0.073	0.065	0.057	0.500	0.066	0.075	0.080	0.083
0.073	0.066	0.058	0.064	0.500	0.075	0.080	0.083
0.074	0.068	0.061	0.066	0.068	0.500	0.080	0.083
0.075	0.068	0.062	0.067	0.069	0.076	0.500	0.083
0.075	0.069	0.062	0.067	0.069	0.077	0.081	0.500

Table 4. Final PRAM matrix 8F(0.5)

0.011	0.009	0.011	0.009	0.926	0.009	0.011	0.014
0.000	0.004	0.971	0.005	0.005	0.005	0.004	0.004
0.027	0.625	0.009	0.027	0.205	0.036	0.036	0.036
0.049	0.037	0.432	0.074	0.062	0.259	0.049	0.037
0.032	0.005	0.006	0.928	0.006	0.008	0.006	0.008
0.892	0.008	0.033	0.008	0.011	0.017	0.017	0.014
0.003	0.030	0.004	0.004	0.466	0.487	0.003	0.003
0.430	0.416	0.005	0.006	0.019	0.006	0.006	0.111

More experiments have been done protecting other attributes and similar results have been obtained.

5 Conclusions

In this paper we have proposed an evolutionary algorithm to seek new and enhanced PRAM matrices in order to obtain better protections for categorical data. The experiments done in this paper have been presented using real survey data.

The $8D(0.7)$ PRAM matrix score got a 65.51% reduction -from 24.18 to 8.34-, and the $8F(0.5)$ PRAM matrix score got a 64.46% reduction -from 18.71 to 6.65- . These results demonstrate the effectiveness of our approach, and show that for some information loss measures the type of PRAM matrices found in this paper might be effective.

Our method has the advantage that can be extended to other measures of information loss and disclosure risk just by changing the fitness function. This property of decoupling of the algorithm from the measures is an important point because it may deserve future research.

On the contrary, the disadvantage is the cost in time for the evaluation of the information loss and disclosure risk. Aproximately, it takes 240 CPU seconds to compute both measures but, if we take in account that we need five computations per generation, a new individual complete evaluation takes 960 CPU seconds. As future work, this is a possible optimization to be explored.

Other lines of future work include the use of PRAM for protecting several variables at the same time and its comparison with other masking methods for categorical data.

In addition, experiments using other datasets with different sizes and structures will be also considered.

References

1. Caruana, R.A., Schaffer, J.D.: Representation and hidden bias: Gray vs binary coding for genetic algorithms. In: Proc. of the 5th Int. Conf. on Machine Learning, pp. 153–161. Morgan Kaufmann, Los Altos (1988)
2. De Wolf, P.P., Van Gelder, I.: An empirical evaluation of PRAM. Discussion paper 04012. Statistics Netherlands, Voorburg/Heerlen (2004)
3. Fienberg, S.E.: Conflict between the needs for access to statistical information and demands for confidentiality. Journal of Official Statistics 10(2), 115–132 (1994)
4. Gouweleeuw, J., Kooiman, P., Willenborg, L., de Wolf, P.P.: Post randomization for statistical disclosure control: Theory and implementation. Journal of Official Statistics 14(4), 463–478 (1998)
5. De Wolf, P.P., Gouweleeuw, J., Kooiman, P., Willenborg, L.: Reflections on pram. In: Statistical Data Protection, pp. 337–349. Office for Official Publications of the European Communities, Luxembourg (1998)
6. Kooiman, P., Willenborg, L., Gouweleeuw, J.: A method for disclosure limitation of microdata. Research paper 9705, Statistics Netherlands, Voorburg (1997)
7. Samarati, P.: Protecting respondents' identities in microdata release. IEEE Transactions on Knowledge and Data Engineering 13(6), 1010–1027 (2001)
8. Domingo-Ferrer, J., Torra, V.: A quantitative comparison of disclosure control methods for microdata. In: Doyle, P., Lane, J.I., Theeuwes, J.J.M., Zayatz, L.V. (eds.) Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies, ch. 6, pp. 111–133. Elsevier, Amsterdam (2001)