Parameter Determination of ONN (Ordered Neural Networks) Technical Report IIIA-TR-2007-05

Jordi Pont-Tuset[1] Pau Medrano-Gracia[1] Jordi Nin[2]

Josep-L. Larriba-Pey[1] Victor Muntés-Mulero[1]

[1]DAMA-UPC, Computer Architecture Dept.
Universitat Politècnica de Catalunya
Campus Nord UPC, C/Jordi Girona 1-3
08034 Barcelona, (Catalonia, Spain)
{jpont,pmedrano,larri,vmuntes}@ac.upc.edu

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

November 23, 2007

Abstract

The need for data privacy motivates the development of new methods that allow to protect data minimizing the disclosure risk without losing information. In this paper, we propose a new protection method for numerical data called *Ordered Neural Networks* (ONN) method. ONN presents a new way to protect data based on the use of Artificial Neural Networks (ANN). ONN combines the use of ANN with a new strategy for preprocessing data consisting in the vectorization, sorting and partitioning of all the values in the attributes to be protected in the data set. We also present an statistical analysis that allows to understand the most important parameters affecting the quality of our method, and we show that it is possible to find a good configuration for these parameters. Finally, we compare our method to the best methods presented in the literature, using data provided by the US Census Bureau. Our experiments show that ONN outperforms the previous methods proposed in the literature, proving that the use of ANNs in these situations is convenient to protect the data efficiently without losing the statistical properties of the set.

1 Introduction

Managing confidential data is a common practice in any organization. In many cases, these data contain valuable statistical information required by third parties and, thus, privacy becomes essential, making it necessary to release data sets preserving the statistics without revealing confidential information. This is a typical problem, for instance, in statistics institutes.

In this scenario, an intruder might try to re-identify a percentage of the protected individuals by applying *Record Linkage* (RL) techniques [9, 18] between some attributes in the protected data set and some attributes obtained from other data sources which includes at least one identifier[1]. Depending on the non-protected attributes obtained by the intruder from other sources, the probability of re-identifying individuals increases, in other words, the larger the number of attributes known, the higher the probability to reveal the identity of the individuals in the protected data set.

Special efforts have been made to develop a wide range of protection methods. These methods aim at guaranteeing an acceptable level of protection of the confidential data. The number of techniques applied to protect data is very large, ranging from simply swapping values of the data set [13] to using complex data models [5].

We present a new type of perturbative[2] protection method called *Ordered Neural Network* (ONN). ONN is based on the use of an array of *artificial neural networks* (ANN) [11] in order to protect the numerical values of a data set. ONN consists of a set of steps that include data preprocessing, the learning process using the ANNs, and the final protection step. The combination of these steps improves the capacity of our method to protect data without losing information.

We also present an statistical model that allows to predict the quality of the protected data set, depending on a set of the most influent parameters in the protection process. This study allows us to find out which are these parameters and to understand their impact on the quality of the results.

This paper is organized as follows. Section 2 presents some ANN basics. In Section 3, we present a detailed description of the ONN method. Section 4 describes the measure used to evaluate a protection method. Section 5 presents an statistical analysis of ONN. Section 6 presents some results. Section 7 includes a brief overview of the related work. Finally, Section 8 draws some conclusions and presents some future work.

2 Preliminaries

In this section, we introduce the minimum basic knowledge necessary to follow the details of our method. First, we give a brief description of a general ANN.

[1] The identifier attributes are used to identify the individual unambiguously. A typical example is the passport number.

[2] Following the definition used in [8].

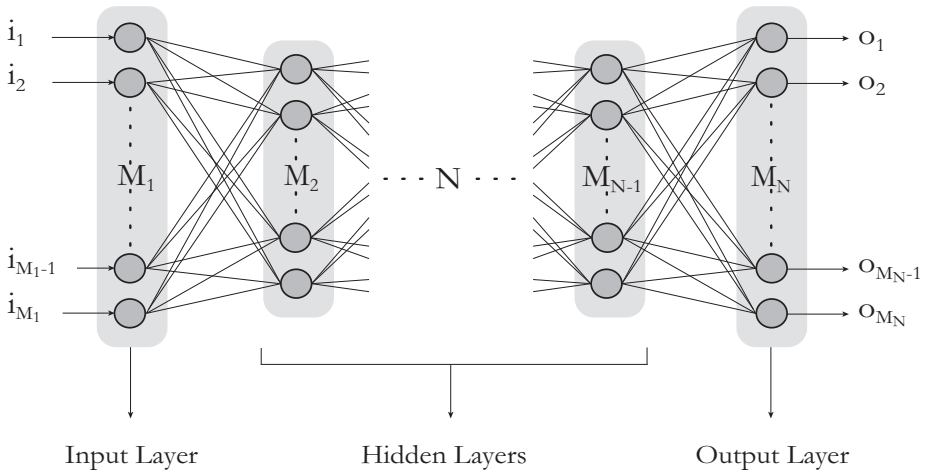

Figure 2: Representation of the activation function and the range of values after

normalization.

$\overline{v}_{m,n}$ the nth element of $\overline{P}_m$.
The normalized input values are defined as:

$$\begin{cases} \overline{v}_{m,n} := & 2\,B_{in}\,\frac{v_{m,n}-\min_m}{\max_m-\min_m} - B_{in} & \text{if } \max_m \neq \min_m \\ \overline{v}_{m,n} := & 0 & \text{if } \max_m = \min_m \end{cases}$$

where $0 \leq m < P$ and $0 \leq n < k$.
Note that $\max_m = \min_m$ means that all the values in the partition are the same. In this case, the normalized value is set to 0, in other words, it is centered in the normalization range.

Analogously, the desired output, which it is denoted by $y_{m,n}$, where $0 \leq m < P,\ 0 \leq n < k$, is normalized between $B1_{out}$ and $B2_{out}$:

$$\begin{cases} \overline{y}_{m,n} := & (B2_{out} - B1_{out})\,\frac{y_{m,n}-\min_m}{\max_m-\min_m} + B1_{out} \\ \overline{y}_{m,n} := & 0.5 \end{cases}$$

where the first component of this expression is used when $\max_m \neq \min_m$ and the second component is used otherwise.
The range of values of the desired outputs is then ($B1_{out}$, $B2_{out}$). Notice that, in the input layer, the outputs fall in the same range, making the training process easier.

Learning

Finally, ONN creates an array of ANNs in order to learn the whole data set, where each ANN is associated to a partition. Therefore, the array contains P ANNs. The objective for each ANN is to learn the values in its corresponding partition. However, an specific ANN is not only fed with the values in that partition, but uses the whole data set to learn. In some sense, using the whole data set, we are adding non-linear noise to the learning process by using input data that is not correlated with the data to be learned.

Specifically, the ANN that learns the values of a partition $\overline{P}_m$ receives the nth value of that partition, $\overline{v}_{m,n}$, together with the $P-1$ nth values of the remaining partitions. Since the network is intended to learn all values $\overline{v}_{m,n}$ in $\overline{P}_m$, the desired output is set to $\overline{y}_{m,n} = \overline{v}_{m,n}$. This process is repeated iteratively until the learning process finishes.

In our proposal, each ANN contains three layers: the input layer, a single hidden layer and the output layer, which can be described as follows:

Input Layer. The input layer consists of $M_1 = P$ neurons. Each of them takes the data from a different partition as input. That is, input of the ith neuron comes from partition $\overline{P}_i$.

Hidden Layer. The hidden layer has $M_2 = n_h$ neurons. As explained in [11], n_h has an effect on the speed of the learning process and the ability of the network to learn complex patterns.

Output Layer. The output layer consists of one single neuron ($M_3 = 1$).

The structure of the array of neural networks is shown in Figure 4.

All networks are forced to learn the example pairs updating their weights by using the iterative backpropagation algorithm explained in Section 2.2. The quality of the data protection obtained depends on the level of accuracy reached when the algorithm is stopped. This level depends basically on the parameters and structure that define the array of ANNs and the way the training set is preprocessed.

Protecting Data

Once the ANNs have been trained, and therefore the weights updated, the last step obtains the protected values for the data set.

Let $\overline{p}_{m,n}$ be the protected value for $\overline{v}_{m,n}$. As mentioned before, the mth ANN of the array has been trained to reproduce $\overline{v}_{m,n}$ when the values in the P input neurons is $\overline{v}_{0,n}, ..., \overline{v}_{P-1,n}$. This way, $\overline{p}_{m,n}$ is defined as the output obtained when having $\overline{v}_{0,n}, ..., \overline{v}_{P-1,n}$ as input of the mth already trained ANN.

Finally, the protected value $p_{m,n}$ for $v_{m,n}$ is obtained by de-normalizing $\overline{p}_{m,n}$ as follows:

$$p_{m,n} = \min_m + \frac{(\overline{p}_{m,n} - B1_{out})(\max_m - \min_m)}{B2_{out} - B1_{out}}$$

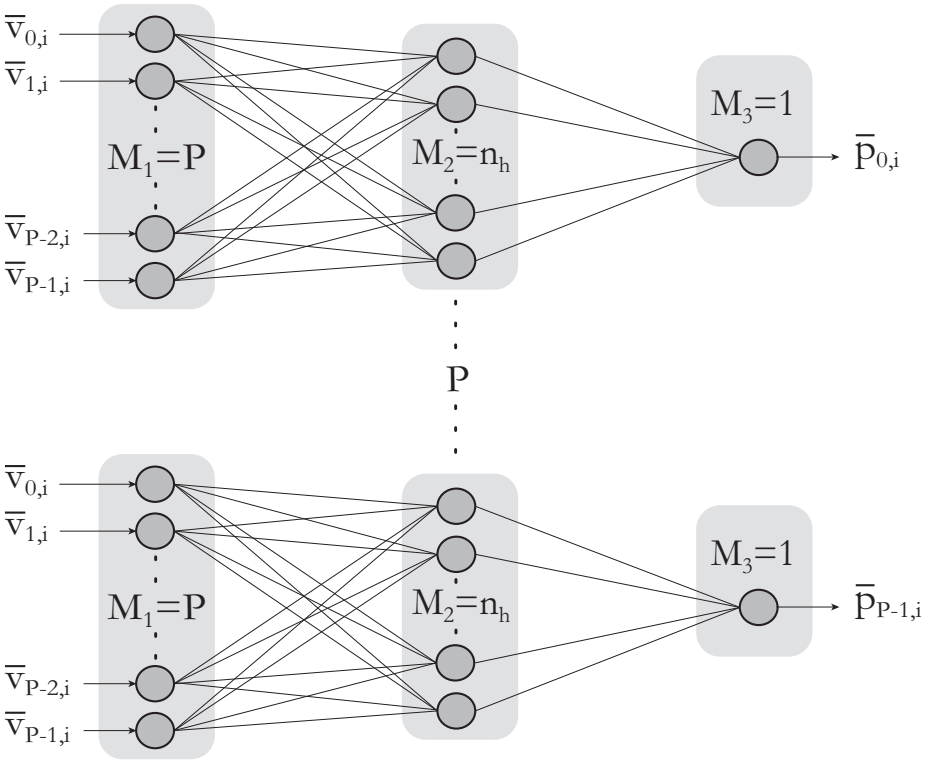

Figure 3: Array of ANNs used by ONN.

If $\max_m = \min_m$ the expression used is the following:

$$p_{m,n} = (\overline{p}_{m,n} + 0.5)v_{m,n}$$

The protected values $p_{m,n}$ are placed in the protected data set in the same place occupied by the corresponding $v_{m,n}$ in the original data set. This way, we are undoing the sorting and vectorization steps.

4 Scoring protection methods

In order to measure the quality of a protection method, we need a protection quality measurement that assigns a score to a method depending on its capacity to (i) make it difficult for an intruder to reveal the original data and (ii) to avoid the information loss in the protected data set. In this paper, we use the score defined in [8] which has been used in several other works.

In order to calculate the *score*, we must first calculate some statistics:

- **Information Loss (IL):** Let X and X' be matrices representing the original and the protected data set, respectively. Let V and R be the covariance matrix and the correlation matrix of X, respectively; let $\overline{X}$ be the vector of variable averages for X and let S be the diagonal of V. Define V', R', $\overline{X}'$, and S' analogously from X'. The information loss is computed by averaging the mean variations of $X - X'$,$V - V'$,$S - S'$, and the mean absolute error of $R - R'$ and multiplying the resulting average by 100.

- **Disclosure Risk (DR):** We use the three different methods presented in [18] in order to evaluate DR: (i) *Distance Linkage Disclosure risk* (DLD), which is the average percentage of linked records using distance based RL, (ii) *Probabilistic Linkage Disclosure risk* (PLD), which is the average percentage of linked records using probabilistic based RL and (iii) *Interval Disclosure risk* (ID) which is the average percentage of original values falling into the intervals around their corresponding masked values. The three values are computed over the number of attributes that the intruder is assumed to know. The Disclosure Risk is computed as $DR = 0.25 \cdot DLD + 0.25 \cdot PLD + 0.5 \cdot ID$.

- **Score:** The final score measure is computed by weighting the presented measures and it was also proposed in [8]:

$$score = 0.5\,IL + 0.5\,DR$$

A simplified version of the score can be used involving only DLD. In this case, the score is

$$score_{simp} = 0.5\,IL + 0.25\,DLD + 0.25\,ID$$

Due to the large execution cost of PLD, we use the $score_{simp}$ to save the execution time of the experiments run in the next section. This simplified score was used in [16] for similar reasons. Note that the better a protection method, the lower its score.

5 ONN Parametrization

In previous sections, we have seen that there are several parameters that must be taken into consideration in order to adjust ONN. The suitability of neural networks has shown to be deeply dependant on the specific problem to be solved. Since the initial weights are randomly assigned to ANNs, whether adjusting these parameters has a direct effect on the quality of the results is still an intriguing question.

Here, we present an statistical analysis that allows us to accurately adjust the parameters involved in ONN, showing that it is possible to find a good set of values that consistently improve the scores obtained by our method. Specifically, we show that it is possible to find a model that predicts the average best score obtained by ONN.

5.1 Analysis of Variance

The variables studied in this chapter are usually called *factors* in statistics. A *factor* is considered to be a categorical variable, i.e., a variable that gives the appropriate label of an observation after allocation to one of several possible categories or values [10], called *levels* in statistics.

The technique used to find this model is the *Analysis of Variance* [12, 15] which is the equivalent to regression when we have categorical variables instead of continuous variables. The Analysis of Variance (from here on ANOVA) is the statistical technique that allows us to distinguish between the variability in the data due to an specific cause, controlled by the experimenter, from the variability prompted by other circumstances. As a broad outline, ANOVA aims at decomposing the total variability of a sample among different parts corresponding to the factors that could potentially be the cause. Once the contribution of each level of each factor in the result is estimated, by comparing variances using a Fisher test, we can decide if the differences caused by the different levels of a factor are statistically significant or not. In our case, the ANOVA allows us to estimate the contribution of each parameter to the score obtained from a set of data protected using the ONN method. Therefore, we will be able to see whether the contribution of each variable, alone or in combination with others, is significant or not.

If the levels of a factor are fixed a priori, which is our case, they are considered constants and the factor is known as a *fixed effects factor*.

In order to define an appropriate model to fit a set of data, we must also take into account the interaction between the factors. Two factors A and B are said to be *crossed factors* when each level of A is observed for each level of B and

vice versa. Given two crossed factors, it makes sense to ask for their interaction, i.e. the situation where two explanatory variables do not act independently on the response variable. More exactly, A and B interact when the effect of a given level of A depends on the level of B with which it is combined. The ANOVA technique allows to determine which interactions are significant.

In order to accept an specific model as a good model to fit a set of data, two conditions must be met: (i) the proportion of the variability in the data explained by the model ($\mathcal{R}^2$) must be close to one ($\mathcal{R}^2$ ranges from 0 to 1) and (ii) the residuals, a measure of the discrepancy between the real values and the values predicted by the model, must be independent and follow a normal distribution with zero mean and constant variance (σ^2). The *mean square error* (MSE) is the estimation of σ^2. The smaller the value of MSE, the better the model fits our data. Another important coefficient to take into account is the coefficient of variation (CV), which is defined as:

$$CV = \frac{\sqrt{MSE}}{\overline{y}} \cdot 100 \tag{3}$$

where $\overline{y}$ is the arithmetic mean of all the observations of the response variable. The CV is a measure of dispersion (or spread) relative to the size and distribution of the values in the data set. Since the CV is a dispersion measure, it is desired that the value of this coefficient is small.

5.2 Statistical Model

In this section, we propose an statistical model that allows to estimate the effect of the different factors that have an impact on the score obtained from a data set protected with our method. In order to create and analyze the model and the data, we have used the Statistical Analysis System (SAS) Release 8.00 [1].

5.3 Variables in the model

The dependant variable in our scenario is the score. As we have seen in previous sections, there are several factors that might have an impact on the score. Among them, we have chosen the factors with a higher impact. These parameters, summarized in Table 1, are (i) the number of attributes used by an intruder to reveal data using RL techniques, denoted by V, (ii) the number of partitions into which the data are split, denoted by P, (iii) the normalization range size ($B2_{out} - B1_{out}$) denoted by B, (iv) the learning-rate parameter η, denoted by E in the model, (v) the activation function parameter c, denoted by C in the model and, finally, (vi) the number of neurons in the hidden layer, denoted by H.

There might be more parameters to take into consideration. However, as we will see during this section, the statistical model is accepted only considering the variance produced by these parameters. This implies that the effect of other parameters may be neglected.

Factors	Description
V	# of attributes used in the DR process
P	number of partitions
B	normalization range size
E	learning-rate parameter (η)
C	activation function parameter
H	# of neurons in the hidden layer

Table 1: Independent factors in the model.

5.4 Description of the experiments

The first approach to the values selected for each factor is based on empirical results in order to use reasonable and realistic values. In the case of V, we have chosen two levels corresponding to half of the attributes ($V = 7$) and all the attributes ($V = 13$). Table 2 summarizes the levels used for the different factors in the experiment.

Given these levels for each factor we run experiments for every possible combination of the values of these six factors. The number of combinations can be calculated as $2 \times 7 \times 7 \times 4 \times 5 \times 2 = 3920$. For every possible combination we run 10 executions and calculate the average score obtained. Therefore, we have run 39200 independent executions and, after obtaining the averages, the number of observations for the dependant variable is equal to 3920.

After studying the initial data set, we have not detected any significant outlier out of the 3920 observations. Therefore, it has not been necessary to remove any value from the data set.

5.5 Model definition and Goodness of fit

The process of building a model that properly predicts the behavior of a response variable in front of some factors is iterative. This means that we depart from a maximal or quasi-maximal model and, consecutively, we remove the terms that

Factor	Studied Levels
V	7 and 13
P	8, 9, 10, 12, 13, 15 and 30
B	0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9
E	0.1, 0.2, 0.3 and 0.4
C	2, 2.5, 3, 3.5 and 4
H	2 and 8

Table 2: Factor levels studied in the experiment.

are not statistically significant. In our case, we have removed the interactions between V and C, and between E and C, for not being statistically significant. The final model is:

$$\begin{aligned} y_{ijklmn} &= \mu + \tau_i + \beta_j + \delta_k + \gamma_l + \alpha_m + \lambda_n + (\tau\beta)_{ij} + \\ &\quad +(\tau\delta)_{ik} + (\tau\gamma)_{il} + (\tau\lambda)_{in} + (\beta\delta)_{jk} + (\beta\gamma)_{jl} + \\ &\quad (\beta\alpha)_{jm} + (\beta\lambda)_{jn} + (\delta\gamma)_{kl} + (\delta\gamma)_{kl} + (\delta\alpha)_{km} + \\ &\quad (\delta\lambda)_{kn} + (\gamma\lambda)_{ln} + (\alpha\lambda)_{mn} + e_{ijklmn} \end{aligned} \tag{4}$$

where,

- y_{ijklmn} is the response or dependent variable corresponding to the average score obtained from a data set protected using our method. As already said, the average is calculated from 10 executions under the same conditions.
- μ represents the mean value of the average score under the baseline situation. In general, the baseline situation corresponds to the combination of the last levels of all the factors.
- τ_i, β_j, δ_k, γ_l, α_m and λ_n correspond to the main effects of the six factors presented before. Specifically:
  - τ_i corresponds to the effect of the ith level of V.
  - β_j corresponds to the effect of the jth level of P.
  - δ_k corresponds to the effect of the kth level of B.
  - γ_l corresponds to the effect of the lth level of E.
  - α_m corresponds to the effect of the mth level of C.
  - λ_n corresponds to the effect of the nth level of H.
- $(\tau\beta)_{ij}$ corresponds to the interaction of the ith level of V with the jth level of P. Analogously $(\tau\delta)_{ik}$, $(\tau\gamma)_{il}$, $(\tau\gamma)_{il}$, $(\tau\lambda)_{in}$, $(\beta\delta)_{jk}$, $(\beta\gamma)_{jl}$, $(\beta\alpha)_{jm}$, $(\beta\lambda)_{jn}$, $(\delta\gamma)_{kl}$, $(\delta\alpha)_{km}$, $(\delta\lambda)_{kn}$, $(\gamma\lambda)_{ln}$ and $(\alpha\lambda)_{mn}$ correspond to the different interactions of the corresponding levels between the factors considered in the model, except for the two interactions removed in the model.
- e_{ijklmn} corresponds to the experimental error.

Appendix B details the SAS program used to fit our data following Model (4).

For this model, the value of $\mathcal{R}^2$ is equal to 0.9914 indicating that the model explains 99.14 % of the total variability in the data. The MSE is equal to 0.464 and the CV is equal to 1.27. Both are really small values, indicating once again that the model fits appropriately the data.

The studentized residuals are obtained from the raw residuals by dividing by its estimated standard deviation. This way, we obtain values from a normal distribution with zero mean and variance equal to 1. Since 95 % of the

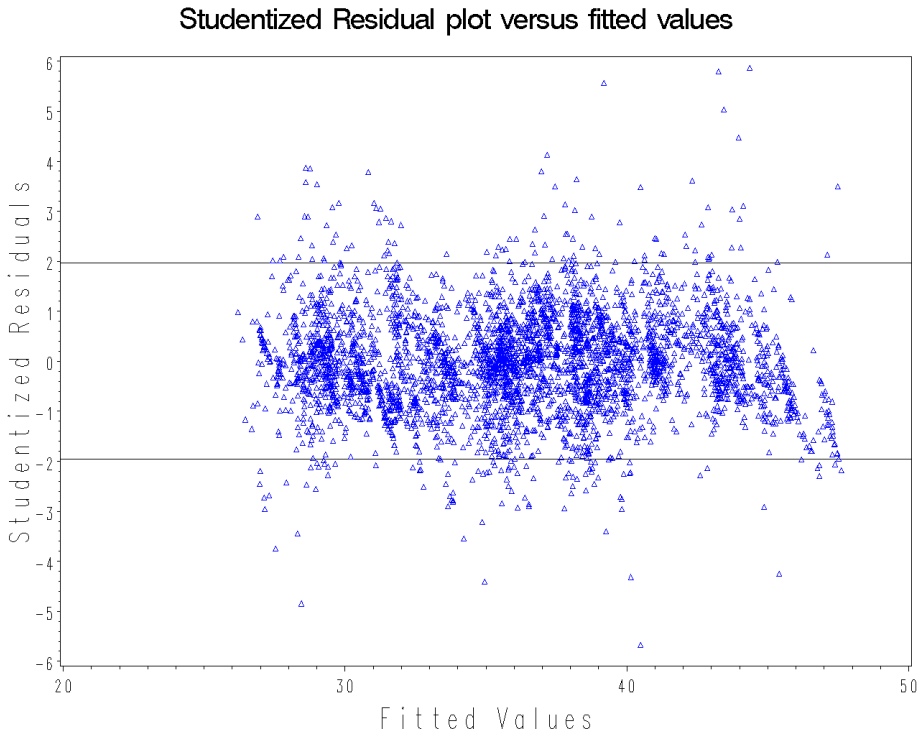

Figure 4: Residuals versus fitted values.

observations from a normal distribution with zero mean and variance equal to 1 range from -1.96 to 1.96, we expect at least 95 % of our studentized residuals to belong to this interval. Figure 5 shows the studentized residuals versus the values fitted by the model. Out of the 3920 residual values, only 205 present an absolute value larger than 1.96, which represents the 5.22 % of the total residual values.

As shown in Figure 6, the studentized residuals follow a normal distribution with zero mean.

Model Analysis

In this subsection, we first analyze the importance and effect of the different factors studied separately. Second, we study the most significant interactions.

Since our model satisfactorily predicts the real score values, we can assure that the factors used in the model are the most significant ones and, also, that they are sufficient to explain the variability of the quality of the results.

The first conclusion extracted from the results is that the number of variables used in the RL process (V) is very influent on the score. Intuitively, the larger the number of variables in the model, the higher the DR and, subsequently, the score. The second most important factor in terms of its effect on the score is

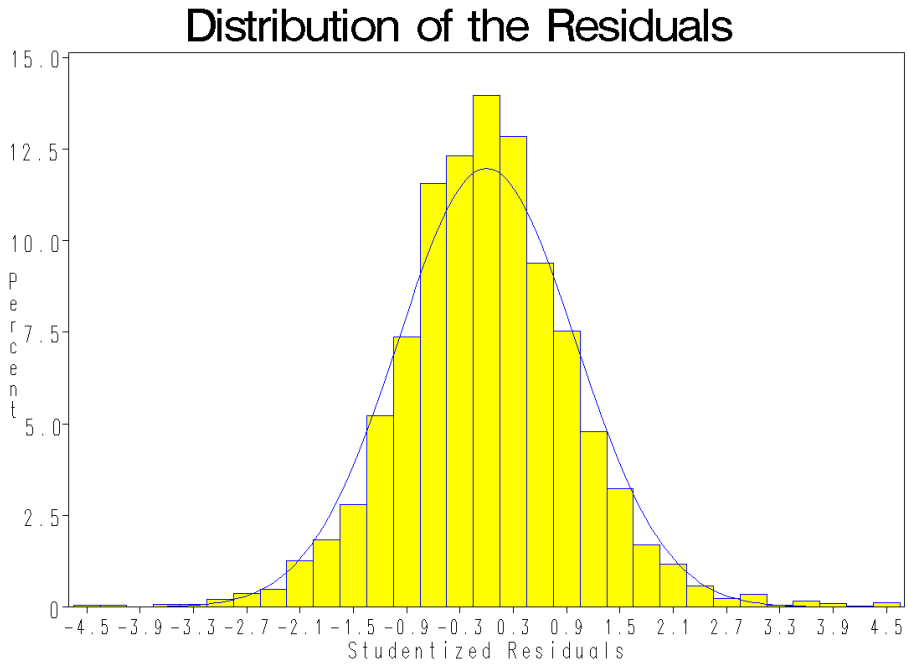

Figure 5: Distribution of Studentized Residuals.

the size of the normalization range (B). Figure 7 shows the average score for different values of the size fo the normalization range, both when the number of variables used in the process is 7 and 13. As expected, the larger the size of the normalization range, the easier the recognition of patterns using ANNs. Nevertheless, later in this subsection, we will see that the best parametrization might not imply having a size close to 0.95, like this first analysis would suggest. Also, note that a normalization range size tending to 1 would also be unsuitable since:

$$\lim_{B \to 1} B_{in} = \infty$$

In this situation, the ANN would not be able to learn.

Our results show that the number of partitions P is also statistically significant. This makes sense since P allows to control the IL of the ONN method. In general, the larger the number of values in a partition, the larger the difference between the original and the protected data because the learning process becomes more difficult. Therefore, as this difference grows, the IL increases affecting negatively the score.

Although factors E and H are statistically significant, variations in their values have a lower impact on the score, compared to V, B and P. In general, the larger η or the number of hidden neurons, the lower the scores. Finally, C is the less significant factor, meaning that there is not a clear relation between choosing one of the levels of this factor and the score.

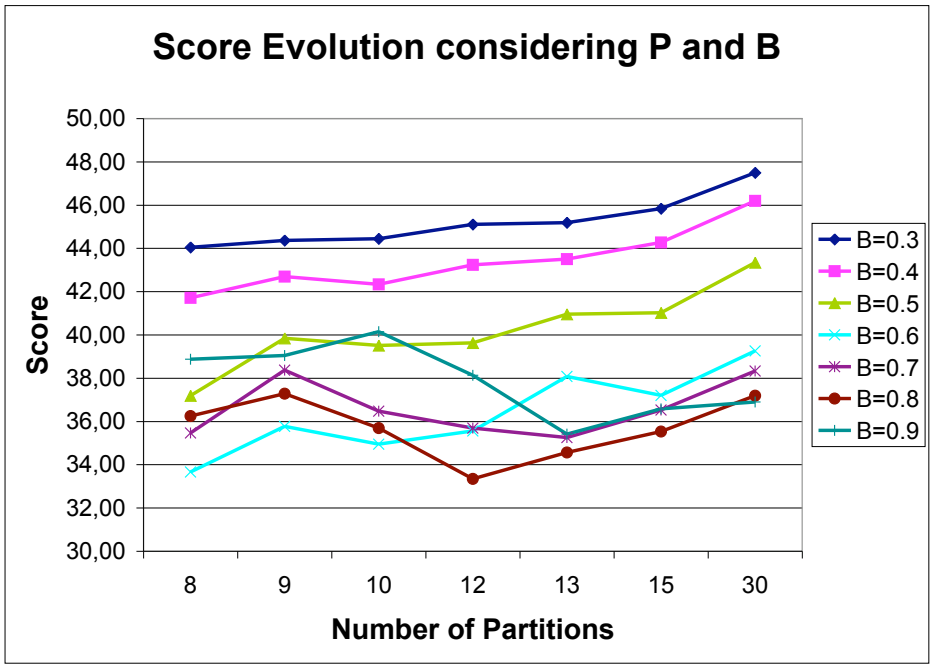

Figure 6: Graphical score evolution considering factors B and T.

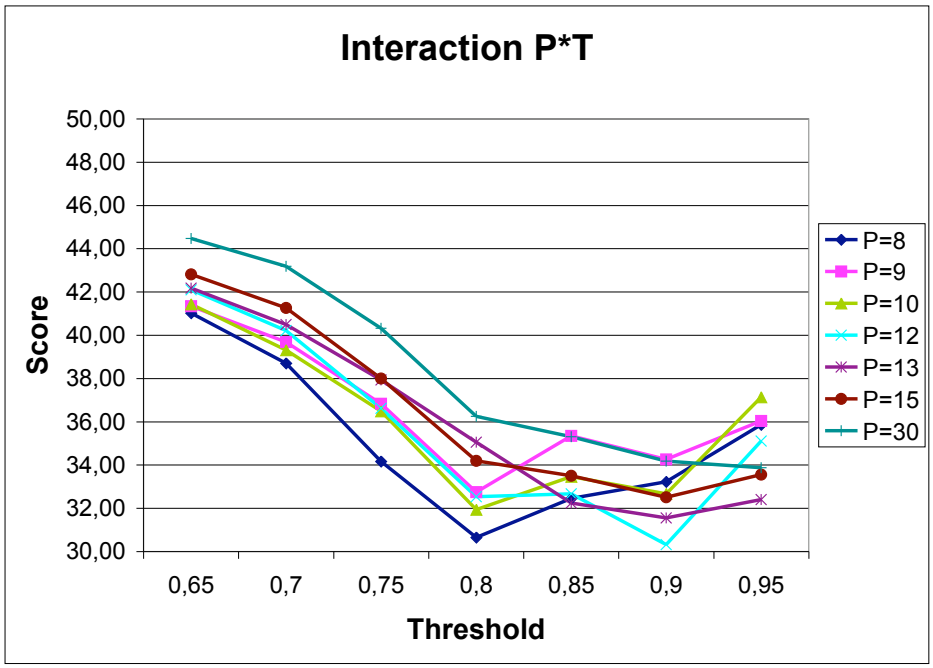

Figure 7: Graphical score evolution considering factors P and T.

Regarding the interactions, we have seen that all of them are statistically significant except for those between V and C and between E and C. Among the significant interactions, we can see that those between factors P and B, V and B, and V and P, in this order, present a larger effect on the score. For example, studying the most significant interaction (between P and B), as observed in Figure 8, the values in our data set show that there is not a unique optimum number of partitions, but it depends on the normalization range size. Specifically, this figure shows that there are two specific configurations that achieve small values for the scores. These configurations correspond to the cases where $B = 0.6$ and $P = 8$, and $B = 0.8$ and $P = 12$, respectively. Although it cannot be seen in the plots, these results stand when we analyze the values for the two levels of V separately.

A detailed analysis of all the interactions is beyond the scope of this paper due to a lack of space. However, studying these interactions we conclude that

	P	B	E	C	H
A	8	0.8	0.4	4.0	8
B	10	0.8	0.1	3.5	8
C	10	0.8	0.1	3.0	8
D	10	0.8	0.1	4.0	8
E	8	0.8	0.1	4.0	2
F	8	0.8	0.3	4.0	8
G	8	0.8	0.1	3.5	2

(a)

	P	B	E	C	H
a	8	0.795	0.4	4.0	8
b	8	0.785	0.3	4.0	8
c	8	0.800	0.4	4.0	8
d	10	0.805	0.2	4.0	8
e	8	0.795	0.3	4.0	8
f	8	0.820	0.4	4.0	8
g	8	0.815	0.4	3.5	8

(b)

Table 3: ONN parameters used in the experiments.

there are two combinations of the values of the parameters that reduce the score. Specifically, the configurations near to $(P, B, E, C, H) = (8, 0.6, 0.3, 4, 8)$ and $(P, B, E, C, H) = (10, 0.6, 0.3, 4, 8)$ obtain the best scores both for $V = 7$ and $V = 13$.

6 Experiments

In order to test ONN, we use the data set provided by the US Census Bureau, described in detail in [8, 14]. The *Census* data set is used in other works like [7, 8, 19]. This data set contains 1080 records consisting of 13 attributes (which is equal to 14040 values to be protected).

In this section, we compare ONN with the best ranked protection methods presented in the literature. The survey in [8] reviews the most common protection methods concluding that *Rank Swapping* (RS-p) [13] and *Microaggregation* (MIC-vm-k) [6] are the two methods that obtain lower scores for numerical data protection. For this reason, we compare ONN with these two methods. Specifically, we have chosen the best five parameterizations for RS-p and MIC-vm-k.

RS-p 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.

Analogously to RS-p and MIC-vm-k, we have computed the score of the best five parameterizations of ONN obtained from the ANOVA (see Table 3.a), and the scores of the best five parameterizations extracted from a more accurate ad-hoc search of the parameter values, using configurations close to the best ones found by the ANOVA (see Table 3.b), executing another 4000 extra runs.

We have divided our experiments into two different scenarios. First, we assume that the intruder only has half of the original protected attributes ($V = 7$), this scenario was used in [8]. Second, we assume that the intruder is able to obtain all the original attributes ($V = 13$). This scenario could be considered the most favorable scenario for the intruder.

Table 4.a shows the scores in the first scenario. As we can observe, the IL

Method	IL	DR	SCR
RS-14	23.83	24.21	24.02
RS-17	27.40	21.87	24.64
RS-12	21.08	27.83	24.45
RS-15	27.44	23.62	25.53
RS-13	25.39	26.35	25.87
MIC4m17	23.98	31.67	27.82
MIC4m19	26.10	31.09	28.59
MIC4m11	21.27	36.22	28.74
MIC3m20	21.95	35.85	28.90
MIC3m15	18.98	39.33	29.15
ONN-A	20.42	26.25	23.33
ONN-B	20.59	26.95	23.77
ONN-C	20.31	27.26	23.78
ONN-D	20.66	26.96	23.81
ONN-E	22.38	25.65	24.01
ONN-a	19.70	26.37	23.04
ONN-b	19.24	26.91	23.17
ONN-c	20.42	26.25	23.33
ONN-d	20.00	26.67	23.33
ONN-e	21.01	25.72	23.37

(a)

Method	IL	DR	SCR
RS-14	23.83	31.32	27.58
RS-17	27.40	28.43	27.92
RS-12	21.08	35.83	28.46
RS-15	27.44	30.51	28.98
RS-13	25.39	33.79	29.59
MIC4m17	23.98	40.80	32.39
MIC4m19	26.10	40.10	33.10
MIC4m11	21.27	46.45	33.86
MIC3m20	21.95	46.70	34.33
MIC3m15	18.98	50.94	34.96
ONN-A	21.45	33.52	27.49
ONN-F	22.23	32.99	27.61
ONN-E	22.38	33.04	27.71
ONN-B	20.59	34.88	27.73
ONN-G	22.56	33.20	27.88
ONN-a	19.70	33.73	26.72
ONN-b	19.24	34.33	26.78
ONN-e	21.01	33.02	27.01
ONN-f	23.33	31.01	27.17
ONN-g	22.86	31.60	27.23

(b)

Table 4: Average results of IL, DLD, PLD, ID using (a) 7 variables and (b) 13 variables.

when protecting data using ONN is lower than that obtained using RS-p or MIC-vm-k. This means that ONN is able to fit the data set better than the other two approaches. These results are coherent with the methodology used by ONN to protect data. Since ONN is trained using the data set after being preprocessed, the patterns learned depend on values that come from different individuals in the original data set. Because of this, it would be necessary for an intruder to know the values in each partition to be able to understand the learned patterns. Since this information is no longer available after protecting the data, with ONN we can get lower ILs while preserving relatively good rates of DR.

Regarding DR, the best disclosure risk corresponds to RS-p. This results make sense because when the intruder has a reduced set of variables, it is very difficult to re-identify individuals because, by swapping, RS-p mixes values from different individuals. ONN presents a good DR, better than that obtained by MIC-vm-k.

Observing the scores, ONN shows to be the best protection method among those presented in this paper and, therefore, all the methods studied in [8]. The scores obtained by ONN are better than those obtained by RS-p and MIC-vm-k, both using the configurations of ONN extracted from the ANOVA and the

configurations of ONN determined by the ad-hoc accurate search. Naturally, the scores in the latter case are even lower, being the best score equal to 23.04. Note that, although the DR is lower for RS-p, the scores show that ONN is better ranked, meaning that the benefits obtained by avoiding the IL compensate for the increase in the DR.

Table 4.b shows similar results for the second scenario, where the intruder has all the variables. As we can observe, the results are very similar. It is important to notice that, the larger the number of variables known by the intruder, the more similar the DR presented by RS-p and ONN.

7 Related Work

Privacy in statistical databases (PSD) is about finding trade-offs to the tension between the increasing societal and economical demand for accurate information and the legal and ethical obligation to protect the privacy of individuals and enterprisers which are the respondents of the statistical data. Statistical agencies cannot expect to collect accurate information from an individual or entity unless they feel that the privacy of their responses is guaranteed; also, recent surveys of web users show that a majority of these are unwilling to provide sensible data to a web site unless they know that privacy protection measures are in place [2].

For these reasons, special efforts have been made to develop a wide range of protection methods. These methods aim at guaranteeing an acceptable level of protection of the confidential data. Good surveys about protection methods can be found in the literature [3, 8].

Another research area where privacy is involved, is Preserving Privacy Data Mining (PPDM) [4]. PPDM studies the case when a data mining technique allows an intruder to obtain confidential information about specific individuals.

Recently, some authors are working in RL improvements to increase the DR of a specific protection methods, for example, in [17] the authors show that the IPSO protection method [5], which uses multiple regression models to protect numerical data, increases its DR when a Mahalanobis distance is applied to Distance Based RL. For this reason, it is clear that new protection methods have to be designed to make the re-identification process more difficult, avoiding ah-hoc attacks.

8 Conclusions & Future Work

In this paper, we have presented ONN, a new method for protecting data minimizing the information loss. To our knowledge, no previous attempts had been made to use artificial neural networks for this purpose. The use of neural networks, combined with other preprocessing techniques, to create a protected data set from the original data has shown to be very useful and efficient. Specifically, we have proven that ONN reduces the combined disclosure risk and the information loss metric beyond the best approaches presented in the literature.

Through an statistical analysis, we have shown that it is possible to find a configuration of the different parameters in the method that allows us to tune our algorithm to obtain a good protection score. Among the main conclusions, we have detected the most important parameters to explain the variation in the score, we have deduced that the number of partitions used in our method controls the information loss, and we have seen that the normalization method used by ONN is necessary.

Future directions of this work include the establishment of a set of criteria that allow to automatically tune the parameters of our method. It would also be interesting to extend the use of ANN for data quality by providing solutions in order to prepare data containing blanks or outliers, for data mining.

9 Acknowledgments

The authors want to thank Generalitat de Catalunya for its support through grant number GRE-00352 and Ministerio de Educación y Ciencia of Spain for its support through grant TIN2006-15536-C02-02. Jordi Nin wants to thank the Spanish Council for Scientific Research (CSIC) for his I3P grant.

References

[1] S. O. D. V. 8. http://v8doc.sas.com/sashtml.

[2] M. S. Ackerman, L. F. Cranor, and J. Reagle. Privacy in e-commerce: examining user scenarios and privacy preferences. In *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, pages 1–8, New York, NY, USA, 1999. ACM Press.

[3] N. R. Adam and J. C. Wortmann. Security-control for statistical databases: a comparative study. *ACM Computing Surveys*, 21:515–556, 1989.

[4] R. Agrawal and R. Srikant. Privacy preserving data mining. In *Proc. of the ACM SIGMOD Conference on Management of Data*, pages 439–450. ACM, 2000.

[5] J. Burridge. Information preserving statistical obfuscation. *Statistics and Computing*, 13:321–327, 2003.

[6] J. Domingo-Ferrer and J. M. Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *Knowledge and Data Engineering, IEEE Transactions on*, 14:189–201, 2002.

[7] J. Domingo-Ferrer, J. M. Mateo-Sanz, and V. Torra. Comparing sdc methods for microdata on the basis of information loss and disclosure risk. In *Pre-proceedings of ETK-NTTS.2001 (vol. 2)*, pages 807–826. Eurostat, 2001.

[8] 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*, pages 111–133. Elsevier Science, 2001.

[9] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):1–16, 2007.

[10] B. S. Everitt. *The Cambridge Dictionary of Statistics*. Cambridge Univ. Press, Cambridge, UK, 1998.

[11] J. A. Freeman and D. M. Skapura. *Neural Networks: Algorithms, Applications and Programming Techniques*, pages 1–106. Addison-Wesley Publishing Company, 1991.

[12] D. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, New York, 1991.

[13] R. Moore. Controlled data swapping techniques for masking public use microdata sets, u. s. bureau of the census (unpublished manuscript). year 1996.

[14] C. C. A. of Statistical Confidentiality Project. http://neon.vb.cbs.nl/casc/.

[15] H. Scheff. *The Analysis of Variance*. John Wiley & Sons, Inc., New York, NY, USA, 1959.

[16] F. Sebé, J. Domingo-Ferrer, J. M. Mateo-Sanz, and V. Torra. Post-masking optimization of the tradeoff between information loss and disclosure risk in masked microdata sets. In *Inference Control in Statistical Databases, Lecture Notes in Computer Science 2316*, pages 187–196. J. Domingo-Ferrer (Ed.), 2002.

[17] V. Torra, J. M. Abowd, and J. Domingo-Ferrer. Using mahalanobis distance-based record linkage for disclosure risk assessment. *Lecture Notes in Computer Science, Springer-Verlag*, 4302:233–242, 2006.

[18] V. Torra and J. Domingo-Ferrer. Record linkage methods for multidatabase data mining. In *Information Fusion in Data Mining*, pages 101–132. Springer-Verlag, 2003.

[19] W. Yancey, W. Winkler, and R. Creecy. Disclosure risk assessment in perturbative microdata protection. In *Inference Control in Statistical Databases, volume 2316 of Lecture Notes in Computer Science*, pages 135–152. Springer-Verlag, 2002.

A Model definition

A.1 Simple Example

The process of building a model that properly predicts the behavior of a response variable in front of some factors is iterative. This means that we depart from a maximal or quasi-maximal model and, consecutively, we remove the terms that are not statistically significant.

Following, we initially present a simple example that allows us to understand how to build an statistical model using the ANOVA technique. Let us suppose that we need to study the impact of the number of processors and the memory available on the execution time of a process. To that end, we study the case for 2, 4 and 8 processors (factor A) and 512 MB and 1 GB memory cards (factor B). For any combination of an specific number of processors and an specific amount of memory, we run the process ten times obtaining ten observations. Thus, the number of observation in our data set is equal to $3 \times 2 \times 10 = 60$. The maximal initial model to analyze the set of data obtained would be the following:

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + e_{ijk}$$

where, y_{ijk} is the kth observation of the response variable under the ith level for the number of processors (i has three levels, i.e. it can take three possible values: $i = 1$ corresponds to 2 processors, $i = 2$ corresponds to 4 processors and $i = 3$ corresponds to 8 processors) and the jth level for the memory available (which has two levels: $j = 1$ corresponds to 512 MB and $j = 2$ corresponds to 1 GB). Parameter μ indicates the expected time for a baseline situation. In general, the baseline situation corresponds to the combination of the last levels of all the factors. In our case, it would correspond to the configuration where we have 8 processors and 1 GB of memory. Parameter α_i is a real number that indicates the variation from μ produced by the knowledge of the number of processors corresponding to the ith level of A. Analogously, β_j is a real number that indicates the variation from μ produced by the knowledge of the available memory corresponding to the jth level of factor B. The term $(\alpha\beta)_{ij}$ corresponds to the interaction of the two factors. Its value represents the contribution in the execution time due to the fact that the observation has been obtained under the ith level for the number of processors and the jth level for the available memory. This interaction will not be significant if the effect of the number of processors is the same for any amount of available memory. The term e_{ijk} corresponds to the experimental error and contains the information in the data which is not explained by any of the different factors. This term is equal to the observed value minus the predicted value under conditions i, j and k and it is also known as the *raw residual*.

In order to estimate the parameters of the model, some conditions must be assumed. By default, the statistical software used (SAS v8.0 [1]) assumes the constraints known as *corner-point constraints*, which are: $\alpha_3 = 0, \beta_2 = 0, (\alpha\beta)_{12} = (\alpha\beta)_{22} = (\alpha\beta)_{32} = 0$. This conditions assume the situation where

we use 8 processors and 1 GB of memory as the baseline, as we mentioned before.

Once the parameters are estimated, ANOVA allows us to accept or refuse the following three hypothesis tests:

$$\begin{aligned} H_0 &: \forall i \in \{1,2,3\}\ \alpha_i = 0 \\ H_1 &: \exists i \in \{1,2,3\} \text{ such that } \alpha_i \neq 0 \end{aligned}$$

$$\begin{aligned} H_0 &: \forall j \in \{1,2\}\ \beta_j = 0 \\ H_1 &: \exists j \in \{1,2\} \text{ such that } \beta_j \neq 0 \end{aligned}$$

$$\begin{aligned} H_0 &: \forall (i,j) \in \{1,2,3\} \times \{1,2\}\ (\alpha\beta)_{ij} = 0 \\ H_1 &: \exists (i,j) \in \{1,2,3\} \times \{1,2\} \text{ such that } (\alpha\beta)_{ij} \neq 0 \end{aligned}$$

If any of these H_0 hypothesis is accepted, it means that the corresponding factor or interaction is statistically not significant and can be removed from the model in order to simplify it. For further details we refer the reader to [15].

Sometimes, it is not possible to find a good statistical model for a data set because the response variable is not a linear function of the factors. When this happens, the first thing to do is to apply a transformation on the response variable in order to achieve this linearity. The most common transformations are the squared root and the logarithm. When we apply a transformation, instead of studying the variability produced by the factors on the response variable, we study the variability produced in the new scale measurement. Nevertheless, applying the inverse function, it is also possible to extract conclusion on the initial response variable.

A.2 SAS Program

The program used in SAS is as follows:

```
PROC GLM;
CLASS V P T E C H;
MODEL y = V P T E C H V*P V*T V*E V*H P*T P*E P*C P*H
            T*E T*C T*H E*H C*H / SOLUTION;
OUTPUT out=fittedAndResidual P=Fitted R=residuals
            student=stud stdr=standard;
```

B SAS Program

The program used in SAS is as follows:

```
PROC GLM;
CLASS V P B E C H;
```

```
MODEL y = V P B E C H V*P V*B V*E V*H P*B P*E P*C P*H
            B*E B*C B*H E*H C*H / SOLUTION;
OUTPUT out=fittedAndResidual P=Fitted R=residuals
            student=stud stdr=standard;
```